



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

Aplicación móvil de control académico utilizando la arquitectura de microservicios bajo la metodología de desarrollo Scrum. Caso: Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos

TESIS

Para optar el Título Profesional de Ingeniero de Sistemas

AUTORES

Omar CUBA ESTRELLA

Maycol Henrry ESPINOZA RAMÍREZ

ASESOR

Percy Edwin DE LA CRUZ VÉLEZ DE VILLA

Lima, Perú

2020



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Cuba, O. & Espinoza, M. (2020). *Aplicación móvil de control académico utilizando la arquitectura de microservicios bajo la metodología de desarrollo Scrum. Caso: Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos.* Tesis para optar el título de Ingeniero de Sistemas. Escuela Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

HOJA DE METADATOS COMPLEMENTARIOS

Código ORCID del autor	Omar Cuba Estrella https://orcid.org/0000-0003-1810-904X Maycol Henrry Espinoza Ramírez https://orcid.org/0000-0001-5747-8089
DNI o pasaporte del autor	Omar Cuba Estrella 72028817 Maycol Henrry Espinoza Ramírez 47113327
Código ORCID del asesor	https://orcid.org/0000-0002-4943-7620
DNI o pasaporte del asesor	08583141
Grupo de investigación	NO
Agencia financiadora	NO
Ubicación geográfica donde se desarrolló la investigación	ESCUELA PROFESIONAL DE OBSTETRICIA, FACULTAD DE MEDICINA – UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS, LA VICTORIA, LIMA, PERÚ (-12.05796288855564, -77.01914274553879)
Disciplinas OCDE	Ingeniería de sistemas y comunicaciones http://purl.org/pe-repo/ocde/ford#2.02.04



**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Profesional de Ingeniería de Sistemas**

Acta Virtual de Sustentación de Tesis

Siendo las 5pm horas del día 28 de diciembre del año 2020 se reunieron virtualmente los docentes designados como miembros de Jurado de Tesis, presidido por el Mg. Santiago Domingo Moquillaza Henríquez (Presidente), el Mg. Luis Ángel Guerra Grados (Miembro) y el Mg. Percy Edwin De la Cruz Vélez de Villa (Miembro Asesor), usando la plataforma Meet para la sustentación Virtual de la tesis Intitulada: “**APLICACIÓN MÓVIL DE CONTROL ACADÉMICO UTILIZANDO LA ARQUITECTURA DE MICROSERVICIOS BAJO LA METODOLOGÍA DE DESARROLLO SCRUM Caso: Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos**”, del Bachiller: **Omar Cuba Estrella**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los Miembros del Jurado.

El Bachiller, en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de 17 (Diecisiete).

A continuación el Presidente del Jurado Mg. Santiago Domingo Moquillaza Henríquez, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 6.05 pm horas, se levantó la sesión.

Presidente

Mg. Santiago D. Moquillaza Henríquez

Miembro

Mg. Luis A. Guerra Grados

Miembro Asesor

Mg. Percy E. De la Cruz Vélez de Villa



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación de Tesis

Siendo las 5pm horas del día 28 de diciembre del año 2020 se reunieron virtualmente los docentes designados como miembros de Jurado de Tesis, presidido por el Mg. Santiago Domingo Moquillaza Henríquez (Presidente), el Mg. Luis Ángel Guerra Grados (Miembro) y el Mg. Percy Edwin De la Cruz Vélez de Villa (Miembro Asesor), usando la plataforma Meet para la sustentación Virtual de la tesis Intitulada: **“APLICACIÓN MÓVIL DE CONTROL ACADÉMICO UTILIZANDO LA ARQUITECTURA DE MICROSERVICIOS BAJO LA METODOLOGÍA DE DESARROLLO SCRUM Caso: Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos”**, del Bachiller: **Maycol Henrry Espinoza Ramírez**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los Miembros del Jurado.

El Bachiller, en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el bachiller obtuvo la nota de 17 (Diecisiete).

A continuación el Presidente del Jurado Mg. Santiago Domingo Moquillaza Henríquez, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 6.05 pm horas, se levantó la sesión.

Presidente

Mg. Santiago D. Moquillaza Henríquez

Miembro

Mg. Luis A. Guerra Grados

Miembro Asesor

Mg. Percy E. De la Cruz Vélez de Villa

Resumen

En el Perú, la calidad educativa en las universidades presenta un bajo nivel a comparación de las demás universidades latinoamericanas según(COMEXPERU, 2011), SINEACE¹(Yanada, Rivera, & Castro, 2012),(Quacquarelli Symonds, 2013),(INEI, 2015), (Quacquarelli Symonds, 2015).

Los estándares del SINEACE que resaltamos son la evaluación del desempeño docente, el adecuado seguimiento a estudiantes y el contar con un sistema informático que apoye a la gestión académica.(SINEACE, 2016)

Los factores académicos como el control de la asistencia a clases, seguimiento académico de alumnos y profesores están relacionados al desempeño académico de alumnos y profesores según (Moromi, 2002), (Rodriguez & Herrera, 2009) y (Tolentino, 2014)

En la Escuela Profesional de Obstetricia – Facultad de Medicina de la Universidad Nacional Mayor de San Marcos se desea contar con un sistema informático que facilite el cumplimiento de los estándares mencionados del SINEACE, actualmente la información de las clases como las asistencias, registro de notas, avisos y consultas no se encuentran centralizadas o disponibles al momento, debido a que se registran en papel físico, archivos excel u otros medios haciendo dificultoso realizar un buen control académico de alumnos y profesores a tiempo.

La propuesta de la presente tesis es desarrollar un sistema informático que solucione el problema descrito y mejore la calidad académica mediante el control académico de la Escuela Profesional de Obstetricia - Facultad de Medicina de la Universidad Nacional Mayor de San Marcos. El sistema propuesto está basado en la solución móvil debido a que era más conveniente su aplicación y el uso de la arquitectura de microservicios en la cual los servicios son aislados, fáciles de implementar, tienen alta cohesión y bajo acoplamiento, proveen de variedad tecnológica, entre otras ventajas.

Además, se optó por SCRUM, la cual es una metodología ágil que permite estar en contacto con los clientes, adaptarse evolutivamente a los cambios, permitiendo acortar tiempos de desarrollo y cuenta con solo documentación necesaria. También asegura una entrega eficiente y organizada del software.

Al finalizar se realizó la validación e implementación del sistema en la cual se revisó si éste provee un adecuado control académico de alumnos y profesores, así como brindar información útil tanto a alumnos, profesores y la dirección de escuela.

PALABRAS CLAVES

Calidad educativa, control académico, rendimiento académico, metodología ágil, microservicios.

¹SINEACE es creada para garantizar que las instituciones educativas en el Perú ofrecen un servicio de calidad

Abstract

In Peru, the educational quality in the universities shows a low level compared to the other Latin American universities according to (COMEXPERU, 2011), SINEACE¹ (Yanada, Rivera, & Castro, 2012),(Quacquarelli Symonds, 2013),(INEI, 2015), (Quacquarelli Symonds, 2015).

The SINEACE standards that we highlight are the evaluation of the teaching performance, the adequate follow-up to students and having a computer system that supports academic management. (SINEACE, 2016)

Academic factors such as the control of class attendance, academic monitoring of students and teachers are related to the academic performance of students and teachers according to (Moromi, 2002), (Rodriguez & Herrera, 2009) and (Tolentino, 2014)

In the Professional School of Obstetrics - Faculty of Medicine of the National University of San Marcos, it is desired to have a computer system that facilitates compliance with the aforementioned standards of SINEACE, currently the information of the classes such as attendance, record of notes, notices and queries are not centralized or available at the moment because they are recorded on physical paper, excel files or other means making it difficult to perform a good academic control of students and teachers on time.

The proposal of this thesis is to develop a computer system that solves the problem described and improves academic quality through academic control of the Professional School of Obstetrics - Faculty of Medicine of the National University of San Marcos. The proposed system is based on the mobile solution because its application was more convenient and the use of the microservice architecture in which the services are isolated, easy to implement, have high cohesion and low coupling, provide technological variety, among other advantages

In addition, SCRUM was chosen, which is an agile methodology that allows us to be in contact with customers, adapt evolutionarily to changes, allowing shorten development times and has only necessary documentation. It also ensures efficient and organized delivery of the software.

At the end, the validation and implementation of the system was carried out in which it was reviewed whether it provides adequate academic control of students and teachers, as well as providing useful information to both students, teachers and the school management.

KEY WORDS

Academic quality, academic performance, academic control, agile methodology, microservices.

¹SINEACE is created to ensure that educational institutions in Peru offer a quality service.

ÍNDICE

ÍNDICE DE TABLAS	11
ÍNDICE DE FIGURAS.....	13
1 INTRODUCCIÓN	18
1.1 Antecedentes	18
1.2 Definición del problema.....	32
1.3 Objetivos	32
1.4 Justificación.....	33
1.5 Alcances y limitaciones	34
1.5.1 Alcances	34
1.5.2 Limitaciones.....	34
1.6 Organización de la tesis	34
2 MARCO TEÓRICO	36
2.1 Definición de términos	36
2.1.1 Control	36
2.1.2 Sistemas de control de gestión.....	36
2.1.3 Calidad educativa.....	38
2.1.4 Acreditación y la mejora continua.....	38
2.1.5 Matriz de estándares de acreditación de Perú	38
2.2 Metodología de desarrollo.....	39
2.2.1 Metodología RUP.....	39
2.2.2 Metodología SCRUM.....	39
2.3 Desarrollo de aplicación móvil.....	40
2.3.1 Web.....	41
2.3.2 Híbrido	41
2.3.3 Javascript nativo	42
2.3.4 Compilación cruzada	43
2.3.5 Nativo	43
2.4 Arquitectura	44
2.4.1 Arquitectura orientada a servicios (SOA)	44
2.4.2 Arquitectura basada en Microservicios	45
3 ESTADO DE ARTE	47
3.1 Taxonomía	47
3.2 Método de revisión de la literatura.....	48
3.2.1 Artículos.....	49

3.2.1.1	Una aproximación holística a las metodologías agiles desde la programación extrema	49
3.2.1.2	JavaScript in mobile applications: React Native vs Ionic vs NativeScript vs Native development	51
3.2.1.3	Disambiguation and Comparison of SOA, Microservices and Self-Contained Systems	53
3.2.1.4	Propuesta de arquitectura de microservicios, metodología Scrum para una aplicación móvil de control académico: Caso Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos.	55
3.2.1.5	Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI	57
3.2.2	Tesis	58
3.2.2.1	Sistema web de gestión académica en el centro de idiomas de la ESPAM MFL	58
3.2.2.2	Sistema de seguimiento de syllabus	60
3.2.2.3	Sistema de seguimiento estudiantil	62
3.2.2.4	Diseño e implementación de una app sobre desarrollo sostenible con backend de arquitectura basada en microservices y de una react native front-end app	64
3.2.2.5	Arquitectura basada en microservicios	66
3.3	Metodología de desarrollo	70
3.3.1	Metodología scrum	70
3.3.1.1	Definición	70
3.3.1.2	Características de scrum	71
3.3.1.3	Flujo scrum	71
3.3.1.4	Elementos de scrum	72
3.3.1.5	Roles	72
3.3.1.6	Lista de requerimientos	73
3.3.1.7	Product backlog	73
3.3.1.8	Sprint scrum	73
3.3.1.9	Planificación	74
3.3.1.10	Sprint backlog	75
3.3.1.11	Scrum diario (Daily)	75
3.3.1.12	Rol del scrum master durante el scrum	75
3.3.1.13	Estimaciones	75
3.3.1.14	Builds continuos y pruebas básicas	76
3.3.1.15	Revisión del sprint	76
3.3.1.16	Reunión retrospectiva scrum	76
3.3.2	Metodología programación extrema	77

3.3.2.1	Definición	77
3.3.2.2	Objetivos	77
3.3.2.3	Fases.....	77
3.3.2.4	Procesos de desarrollo.....	78
3.3.2.5	Valores.....	78
3.3.2.6	Principios.....	79
3.3.3	Metodología Iconix	79
3.3.3.1	Definición	79
3.3.3.2	Fase: Análisis de requisitos.....	80
3.3.3.3	Fase: Diseño preliminar	80
3.3.3.4	Fase: Diseño detallado.....	81
3.3.3.5	Fase: Implementación	81
3.3.3.6	Fase: Pruebas y revisiones	81
3.4	Aplicativo móvil.....	82
3.4.1	Ionic	82
3.4.1.1	Definición	82
3.4.1.2	Componentes UI	82
3.4.1.3	Continuidad de plataforma	82
3.4.1.4	Navegación	82
3.4.1.5	Acceso nativo	82
3.4.1.6	Temática.....	83
3.4.1.7	Arquitectura.....	83
3.4.1.8	Ventajas.....	83
3.4.1.9	Desventajas.....	84
3.4.2	NativeScript	84
3.4.2.1	Definición	84
3.4.2.2	Ventajas.....	84
3.4.2.3	Desventajas.....	85
3.4.2.4	Arquitectura.....	85
3.4.3	React Native	86
3.4.3.1	Definición	86
3.4.3.2	React	86
3.4.3.3	Estructura de componentes	86
3.4.3.4	Flujo de datos unidireccional	86

3.4.3.5	Reconciliación	87
3.4.3.6	Flux	87
3.4.3.7	Partes internas	88
3.4.3.8	De código a renderizado nativo.....	89
3.4.3.9	Puentear la comunicación entre javascript y código nativo.....	90
3.4.3.10	Ambiente de desarrollo	92
3.4.4	Flutter.....	94
3.4.4.1	Definición	94
3.4.4.2	Plataforma específica de código	95
3.4.4.3	Ambiente de desarrollo	95
3.4.4.4	Interfaz de usuario	95
3.4.4.5	Desarrollo de aplicativo	95
3.4.4.6	Depuración y pruebas	96
3.4.5	Xamarin.....	96
3.4.5.1	Definición	96
3.4.5.2	Ambiente de desarrollo	99
3.4.5.3	Desarrollo de aplicación.....	99
3.4.5.4	Pruebas.....	99
3.5	Arquitectura	100
3.5.1	Arquitectura de 3 capas	100
3.5.1.1	Definición	100
3.5.1.2	Capas.....	100
3.5.1.3	Es una arquitectura monolítica	101
3.5.1.4	Ventajas.....	101
3.5.1.5	Desventajas.....	101
3.5.2	Arquitectura orientada a servicios (SOA).....	101
3.5.2.1	Definición	101
3.5.2.2	Elementos de SOA	101
3.5.2.2.1	Aplicación Frontend.....	102
3.5.2.2.2	Servicios.....	102
3.5.2.2.3	Repositorio de servicios.....	103
3.5.2.2.4	Bus de servicios	104
3.5.2.3	Principios de diseño de SOA.....	105
3.5.2.3.1	Estandarización de contratos de servicio.....	105

3.5.2.3.2	Acoplamiento de servicios.....	106
3.5.2.3.3	Abstracción de servicios.....	107
3.5.2.3.4	Reutilización de servicios	108
3.5.2.3.5	Autonomía de servicios	108
3.5.2.3.6	Servicios sin estado.....	109
3.5.2.3.7	Descubrimiento de servicios	110
3.5.2.3.8	Composición de servicios	110
3.5.3	Microservicios.....	111
3.5.3.1	Definición	111
3.5.3.2	Principios.....	111
3.5.3.3	Ventajas.....	113
3.5.3.4	Desventajas.....	115
3.6	Casos de éxito.....	116
3.6.1	Sistema web de gestión académica en el centro de idiomas de la ESPAM MFL	116
3.6.1.1	Objetivos	116
3.6.1.2	Solución	117
3.6.1.3	Resultados	117
3.6.2	Implementación de un sistema web para la mejora del proceso académico de la institución educativa wari-vilca-huayucachi.....	117
3.6.2.1	Objetivos	117
3.6.2.2	Solución	117
3.6.2.3	Resultados	118
3.6.3	Implementación de un sistema web para optimizar la gestión académica en la I.E. “Villa corazón de Jesús” del distrito de San Juan de Lurigancho.	118
3.6.3.1	Objetivos	118
3.6.3.2	Solución	119
3.6.3.3	Resultados	119
3.7	Benchmarking.....	120
4	RESOLUCIÓN DEL PROBLEMA	122
4.1	Cuadros comparativos	122
4.1.1	Comparación metodología de desarrollo.....	122
4.1.2	Comparación frontend (lado cliente)	124
4.1.3	Comparación backend (lado servidor)	127
4.2	Descripción de la solución tecnológica.....	130
4.2.1	Inicio	130

4.2.1.1	Visión del proyecto	131
4.2.1.2	Identificación de scrum master y stakeholders.....	131
4.2.1.3	Equipo SCRUM.....	131
4.2.1.4	Creación del backlog priorizado del producto.....	131
4.2.1.5	Realizar la planificación del lanzamiento.....	131
4.2.2	Análisis y planificación	132
4.2.2.1	Módulos	132
4.2.2.2	Product backlog del sistema	132
4.2.2.3	Sprint 0	143
4.2.2.4	Sprint 1	144
4.2.2.5	Sprint 2	148
4.2.2.6	Sprint 3	153
4.2.2.7	Sprint 4	159
4.2.2.8	Sprint 5	163
4.2.3	Diseño de la solución	167
4.2.3.1	Diseño de la arquitectura.....	167
4.2.3.1.1	Lado cliente	168
4.2.3.1.1.1	React Native	168
4.2.3.1.2	Lado servidor.....	169
4.2.3.1.2.1	Componentes no funcionales	169
4.2.3.1.2.1.1	Eureka-Server.....	169
4.2.3.1.2.1.2	Config-Server.....	170
4.2.3.1.2.1.3	API Gateway	171
4.2.3.1.2.1.4	Composición.....	172
4.2.3.1.2.2	Componentes funcionales	175
4.2.3.1.2.2.1	Autenticación	175
4.2.3.1.2.2.2	Asistencia.....	176
4.2.3.1.2.2.3	Encuesta	178
4.2.3.1.2.2.4	Coparticipación	179
4.2.3.1.2.2.5	Notas	181
4.2.3.1.2.2.6	Periodo académico.....	183
4.2.3.1.2.2.7	Notificación	185
4.2.3.1.2.2.8	Reportes.....	186
4.2.3.2	Herramientas útiles	187

4.2.3.3	Manual de usuario.....	190
4.2.3.3.1	Inicio	190
4.2.3.3.2	Módulo del profesor	191
4.2.3.3.3	Módulo del alumno.....	211
4.2.3.3.4	Módulo director(a) de escuela	219
5	CONCLUSIONES.....	223
6	BIBLIOGRAFÍA.....	224
	ANEXOS	230
	Anexo 1: Encuestas a alumnos en Google Forms	230
	Anexo 2: Constancia de la validación del producto.....	237

ÍNDICE DE TABLAS

Tabla 1 Tópicos centrales y preguntas de sistema de control de gestión.....	37
Tabla 2 <i>Mejora de procesos de gestión académica ESPAM-MFL</i>	117
Tabla 3 <i>Beneficios tangibles del sistema web para I.E. Villa Corazón de Jesús</i>	119
Tabla 4 <i>Beneficios intangibles del sistema web para I.E. Villa Corazón de Jesús</i>	119
Tabla 5 <i>Benchmarking de las soluciones</i>	120
Tabla 6 Comparativa entre metodologías tradicionales y ágiles.....	122
Tabla 7 Resumen porcentual por área de proceso de metodologías ágiles (Scrum, XP, Iconix)	123
Tabla 8 <i>Comparación de frameworks multiplataformas Xamarin, ReactNative, Flutter</i>	124
Tabla 9 Resultado de análisis comparativo de criterios con escala de Likert: desarrollo nativo, React Native, Native Script e Ionic	126
Tabla 10 Comparativa entre arquitectura monolítica y arquitectura de microservicios.....	127
Tabla 11 Comparativa entre SOA y arquitectura de microservicios.....	129
Tabla 12 Creación del backlog priorizado del producto.....	131
Tabla 13 Product backlog del sistema: pesos y estimaciones.....	133
Tabla 14 Product backlog: Descripción de historias de usuario (HU) y lista de criterios de aceptación (CA).....	134
Tabla 15 Sprint 0 del sistema	143
Tabla 16 Sprint 1 del sistema	145
Tabla 17 Sprint 2 del sistema	148
Tabla 18 Sprint 3 del sistema	153
Tabla 19 Sprint 4 del sistema	159
Tabla 20 Sprint 5 del sistema	164
Tabla 21 Distribución de encuestas realizadas por los alumnos	232
Tabla 22 Resultados de la pregunta: ¿Qué tan lejos vive de su centro de estudios (campus universitario) y lugares relacionados como hospitales?.....	233
Tabla 23 Resultados de la pregunta: ¿(Cumplen/Cumplieron) los profesores con todas lasclases del syllabus?	233
Tabla 24 Resultados de la pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las clases avanzadas por el profesor?	234
Tabla 25 Resultados de la pregunta: ¿Qué tan difícil (es/fue) llevar un control de asistencias que has realizado en tus cursos?.....	234
Tabla 26 Resultados de la pregunta: ¿Considera que (hay/hubo) retrasos en el tiempo en el que te enterabas de tus notas?.....	235
Tabla 27 Resultados de la pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las notas que has obtenido en tus cursos?	235
Tabla 28 Resultados de la pregunta: ¿Cree usted que es mejor evaluar al profesor por cada clase en vez de hacerlo una vez cada semestre, de tal forma que se pueda obtener una retroalimentación más rápida y realista?	235
Tabla 29 Resultados de la pregunta: ¿Demora/Demoraba mucho resolver las dudas sobre el curso?	236
Tabla 30 Resultados de la pregunta: ¿Cuánto demora/demoraba en enterarse de los anuncios del profesor tales como postergación de clases, reprogramación o cualquier otro relacionado al curso?	236

Tabla 31 Resultados de la pregunta: ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento (al instante)?	236
Tabla 32 Resultados de la pregunta ¿Considera que la información del curso se encuentra descentralizada y desorganizada?	237
Tabla 33 Resultados de la pregunta ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento (al instante)?	237

ÍNDICE DE FIGURAS

Figura 1. Relación de dimensiones y factores de modelo de acreditación de programas de estudios universitarios (SINEACE, 2016).....	20
Figura 2. Gráfica lineal de pregunta: ¿En que año que se encuentran cursando actualmente o indicar si es egresado? (Elaboración propia)	22
Figura 3. Gráfica lineal de pregunta: ¿Qué tan lejos viven los alumnos? (Elaboración propia).23	23
Figura 4. Gráfica lineal de pregunta: ¿(Cumplen/Cumplieron) los profesores con todas las clases del syllabus? (Elaboración propia).....	23
Figura 5. Gráfica lineal de pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las clases avanzadas por el profesor? (Elaboración propia)	23
Figura 6. Gráfica lineal de pregunta: ¿Qué tan difícil (es/fue) llevar un control de asistencias que has realizado en tus cursos? (Elaboración propia)	24
Figura 7. Gráfica lineal de pregunta: ¿Considera que (hay/hubo) retrasos en el tiempo en el que te enterabas de tus notas? (Elaboración propia).....	24
Figura 8. Gráfica lineal de pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las notas que has obtenido en tus cursos? (Elaboración propia)	25
Figura 9. Gráfica lineal de pregunta: ¿Cree usted que es mejor evaluar al profesor por cada clase en vez de hacerlo una vez cada semestre, de tal forma que se pueda obtener una retroalimentación más rápida y realista? (Elaboración propia)	25
Figura 10. Gráfica lineal de pregunta: ¿Demora/Demoraba mucho en resolver las dudas sobre el curso? (Elaboración propia)	26
Figura 11. Gráfica lineal de pregunta: ¿Cuánto demora/demoraba en enterarse de los anuncios del profesor tales como postergación de clases, reprogramación o cualquier otro relacionado al curso? (Elaboración propia)	26
Figura 12. Gráfica lineal de pregunta: ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizadas disponible para poder consultar en cualquier momento (al instante)? (Elaboración propia)	26
Figura 13. Gráfica lineal de pregunta: ¿Considera que la información del curso se encuentra descentralizada y desorganizada? Por ejemplo: el uso del papel, archivos excel, sistemas web, redes sociales. (Elaboración propia)	27
Figura 14. Gráfica circular de pregunta: ¿Consideraría que un aplicativo móvil(celular) que permita el manejo de información de los cursos sería más útil que el uso del papel o de sistemas que solo funcionen en web? (Elaboración propia).....	27
Figura 15. Árbol de problemas del inadecuado control académico en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia)	29
Figura 16. Diagrama de procesos AS-IS del registro y consulta de asistencias y notas en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia)	30
Figura 17. Diagrama de procesos en base a proceso AS-IS de los anuncios del profesor para los alumnos de su curso en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia).....	31
Figura 18. Diagrama de procesos en base a proceso AS-IS de las preguntas y respuestas de los alumnos y profesores de un curso en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia)	32
Figura 19. Diagrama del árbol de objetivos (Elaboración propia).....	33
Figura 20. Clasificación de las tecnologías de desarrollo móvil(Rinaldi, 2016)	41
Figura 21. Arquitectura de aplicativos híbridos (Alférez, 2018).....	41

Figura 22. Arquitectura de aplicativos móviles basados en javascript (Leler, 2017)	42
Figura 23. Arquitectura de aplicaciones móviles nativas (Leler, 2017)	43
Figura 24. Elementos de una Arquitectura orientada a servicios (Krafzig, Banke, & Slama, 2004)	45
Figura25. Arquitecturas monolíticas vs arquitectura de microservicios (Perez-Herrera, 2015).46	
Figura 26. Clasificación ACM 2012(Association for Computing Machinery, 2019)	47
Figura 27. Taxonomía IEEE 2019 (IEEE, 2019)	47
Figura 28. Líneas de investigación de la facultad de ingeniería de sistemas e informática(Facultad de Ingeniería de Sistemas e Informática UNMSM, 2019)	48
Figura 29. Casos de uso del sistema de seguimiento al estudiante(Freire, 2015).....	63
Figura 30. Diseño interno de capas del sistema (Chacón, 2016).....	65
Figura 31. Arquitectura propuesta: Microservicios de venta online de productos (Perez-Herrera, 2015)	68
Figura 32. Diferencias entre máquina virtual y contenedor docker(Perez-Herrera, 2015).....	69
Figura 33. Flujo de trabajo de scrum (Fernandez & Cadelli, 2014)	72
Figura 34. Fases de XP (Pardo, Triana, & Forero, 2014).....	78
Figura 35. Esquema de metodología Iconix (Porras, 2019)	80
Figura 36. Construcción del diagrama de secuencia (Porras, 2019)	81
Figura 37. Arquitectura de Ionic para plataforma móvil (Christoph, Rösch, & Schuster, 2018)	83
Figura 38. Uso de javascript, css y xml (Branstein & Branstein, 2017).....	84
Figura 39. Como corre compilación JIT en los dispositivos (Branstein & Branstein, 2017)	85
Figura 40. Arquitectura de NativeScript (Branstein & Branstein, 2017).....	85
Figura 41. Flujo de datos en react (Alférez, 2018).....	87
Figura 42. React y React Native Virtual DOM (Alférez, 2018).....	87
Figura 43. Flujo de datos en Flux (Alférez, 2018)	88
Figura 44. Esquema de proceso de ejecución en react native (Alférez, 2018)	89
Figura 45. Puentear en React Native (Alférez, 2018).....	90
Figura 46. Componente de interfaz nativa de usuario en React Native(Alférez, 2018)	91
Figura 47. Componente de interfaz de módulo nativo en React Native(Alférez, 2018).....	91
Figura 48. Arquitectura del sistema Flutter (Alférez, 2018).....	94
Figura 49. Arquitectura de aplicativos móviles usando Flutter (Alférez, 2018).....	95
Figura 50. Ecosistema .NET (Alférez, 2018).....	96
Figura 51. Enlace IOS (Alférez, 2018).....	97
Figura 52. Enlace Android (Alférez, 2018).....	97
Figura 53. Pilas de Xamarin y Xamarin.Forms (Alférez, 2018).....	98
Figura 54. Patrón de Modelo-Vista-ModeloVista (Alférez, 2018).....	98
Figura 55. Dependencia de servicios Xamarin (Alférez, 2018)	99
Figura 56. Arquitectura de 3 capas (Sarasty, 2015)	100
Figura 57. Elementos de SOA (Krafzig, Banke, & Slama, 2004)	102
Figura 58. Componentes de un servicio en SOA (Krafzig, Banke, & Slama, 2004).....	103
Figura 59. Estándar de contratación de servicios en SOA(Erl, 2019)	106
Figura 60. Abstracción de servicios en SOA. (Erl, 2019)	107
Figura 61. Autonomía de servicios en SOA (Erl, 2019).....	109
Figura 62. Tipos de estados de datos en SOA (Erl, 2019).....	109
Figura 63. Ejemplo práctico de microservicios: red social (Newman, 2015, pág. 4).....	114

Figura 64. Escalabilidad de los microservicios (Newman, 2015, pág. 6).....	114
Figura 65. Arquitectura de la solución para el Instituto Educativo Wari-Vilca (Acevedo, 2018)	118
Figura 66. Módulos del sistema (Elaboración propia).....	132
Figura 67. Diagrama del diseño de la arquitectura del sistema. (De la Cruz, Espinoza, & Cuba, 2019)	167
Figura 68. Diagrama del funcionamiento del servidor spring cloud eureka(Rajesh, 2016, págs. 232-235)	170
Figura 69. Diagrama del funcionamiento del servicio spring cloud config (Rajesh, 2016, pág. 213)	171
Figura 70. Tipos de API Gateway (Rajesh, 2016, pág. 153)	171
Figura 71. Diagrama de funcionamiento del API Composer (Richardson, 2018, pág. 239).....	172
Figura 72. Diagrama de API Composer en el cliente. (Richardson, 2018, pág. 241).....	173
Figura 73. Diagrama de API Composer en el API Gateway (Richardson, 2018, pág. 242).....	174
Figura 74. Diagrama de API Composer como servicio aislado (Richardson, 2018, pág. 243) ..	174
Figura 75. Diagrama de base de datos del servicio autenticación(Elaboración propia).....	176
Figura 76. Diagrama de base de datos del servicio asistencia(Elaboración propia)	177
Figura 77. Diagrama de base de datos del servicio encuesta(Elaboración propia)	179
Figura 78. Diagrama de base de datos del servicio coparticipación(Elaboración propia) ..	180
Figura 79. Diagrama de base de datos del servicio notas(Elaboración propia)	182
Figura 80. Diagrama de base de datos del servicio periodo-académico(Elaboración propia) ..	184
Figura 81. Diagrama de base de datos del servicio notificación(Elaboración propia).....	186
Figura 82. Uso de los servicios en la herramienta Postman (Elaboración propia)	188
Figura 83. Uso de Bitbucket con repositorio GIT (Elaboración propia).....	188
Figura 84. Código fuente del repositorio GIT en Bitbucket(Elaboración propia)	189
Figura 85. Uso de nube OneDrive para compartir archivos (Elaboración propia)	189
Figura 86. Interfaz: Acceso al sistema (Elaboración propia).....	191
Figura 87. Interfaz: Elegir accesos (Elaboración propia)	191
Figura 88. Interfaz del profesor: Visualizar las clases de hoy(Elaboración propia)	192
Figura 89. Interfaz del profesor: Visualizar menú lateral (Elaboración propia)	192
Figura 90. Interfaz del profesor: Visualizar clases de hoy(Elaboración propia).....	193
Figura 91. Interfaz del profesor: Visualizar clases de hoy – Iniciar clase (Elaboración propia)	194
Figura 92. Interfaz del profesor: Visualizar clases de hoy – Modificar asistencias de alumnos (Elaboración propia).....	194
Figura 93. Interfaz del profesor: Visualizar clases de hoy – Registrar modificación de asistencias de alumnos (Elaboración propia).....	195
Figura 94. Interfaz del profesor: Visualizar clases de hoy – Terminar clase (Elaboración propia)	196
Figura 95. Interfaz del profesor: Visualizar asistencias – Listar cursos (Elaboración propia) .	196
Figura 96. Interfaz del profesor: Visualizar asistencias – Listar agrupaciones curriculares (Elaboración propia).....	197
Figura 97. Interfaz del profesor: Visualizar asistencias – Listar clases de una agrupación curricular (Elaboración propia)	198
Figura 98. Interfaz del profesor: Visualizar asistencias – Detalle de la clase (Elaboración propia)	199

Figura 99. Interfaz del profesor: Visualizar calificaciones - Listado de calificaciones según variable específica seleccionada (Elaboración propia)	199
Figura 100. Interfaz del profesor: Visualizar calificaciones – Modificar listado de calificaciones (Elaboración propia).....	200
Figura 101. Interfaz del profesor: Visualizar calificaciones – Registro de calificaciones (Elaboración propia).....	200
Figura 102. Interfaz del profesor: Visualizar promedio general (Elaboración propia).....	201
Figura 103. Interfaz del profesor: Visualizar promedio general por alumno (Elaboración propia)	
.....	201
Figura 104. Interfaz del profesor: Visualizar preguntas (Elaboración propia)	202
Figura 105. Interfaz del profesor: Publicar pregunta (Elaboración propia).....	203
Figura 106. Interfaz del profesor: Pregunta publicada (Elaboración propia).....	203
Figura 107. Interfaz del profesor: Realizar respuesta (Elaboración propia)	205
Figura 108. Interfaz del profesor: Respuesta realizada (Elaboración propia).....	205
Figura 109. Interfaz del profesor: Visualizar anuncios. (Elaboración propia).....	206
Figura 110. Interfaz del profesor: Publicar anuncio (Elaboración propia)	207
Figura 111. Interfaz del profesor: Anuncio publicado (Elaboración propia).....	207
Figura 112. Interfaz del profesor: Ver fórmula de grupo (Elaboración propia).....	208
Figura 113. Interfaz del profesor: Personalizar fórmula (Elaboración propia)	209
Figura 114. Interfaz del profesor: Personalizar subfórmulas (Elaboración propia).....	209
Figura 115. Interfaz del profesor: Fórmula confirmar personalización(Elaboración propia) ..	210
Figura 116. Interfaz del profesor: Fórmula correctamente personalizada (Elaboración propia)	
.....	211
Figura 117. Interfaz del alumno: Visualizar las clases de hoy (Elaboración propia).....	212
Figura 118. Interfaz del alumno: Visualizar menú lateral (Elaboración propia)	212
Figura 119. Interfaz del alumno: Visualizar encuesta (Elaboración propia)	213
Figura 120. Interfaz del alumno: Encuesta realizada (Elaboración propia)	214
Figura 121. Interfaz del alumno: Visualizar asistencias – Listar cursos (Elaboración propia)	214
Figura 122. Interfaz del alumno: Visualizar asistencias – Listar agrupaciones curriculares (Elaboración propia).....	215
Figura 123. Interfaz del alumno: Visualizar asistencias – Listado de clases de una agrupación curricular (Elaboración propia)	215
Figura 124. Interfaz del alumno: Visualizar asistencias – Detalle de la clase (Elaboración propia)	216
Figura 125. Interfaz del alumno: Visualizar calificaciones – Listado de calificaciones (Elaboración propia).....	217
Figura 126. Interfaz del alumno: - Visualizar anuncios (Elaboración propia).....	218
Figura 127. Interfaz del alumno: - Ver notificaciones (Elaboración propia).....	218
Figura 128. Interfaz de director(a) de escuela – Visualizar menú lateral (Elaboración propia)	
.....	219
Figura 129. Interfaz de director(a) de escuela: Visualizar estadísticas de asistencias (Elaboración propia).....	220
Figura 130. Interfaz de director(a) de escuela - Visualizar estadísticas de calificaciones (Elaboración propia).....	221
Figura 131. Interfaz de director(a) de escuela: Visualizar estadísticas de encuestas (Elaboración propia)	222
Figura 132. Primera parte del formato de la encuesta (Elaboración propia)	230

Figura 133. Segunda parte del formato de la encuesta (Elaboración propia)	231
Figura 134. Fórmula del cálculo del tamaño de la muestra de población fija (Torres & Paz, 2006)	232
Figura 135. Constancia de validación de la especialista (Elaboración propia).....	238

1 INTRODUCCIÓN

1.1 Antecedentes

De los últimos años, existe información de encuestas, rankings e indicadores que muestran que la educación universitaria en Perú ha tenido muchos inconvenientes a continuación se muestran las comparaciones del ranking de las universidades peruanas con respecto a las universidades latinoamericanas.

Según (COMEXPERU, 2011) en el año 2011, era evidente que había problemas en la educación universitaria en Perú. Las causas de estos problemas mencionan:

- La desigualdad de oportunidades para el acceso de la educación universitaria: Debido a que todas las oportunidades que existían en el país se centraban en Lima y las demás provincias no podían tener acceso a la misma calidad educativa.
- Incremento de las universidades nacionales y privadas: Esto no garantizaba un buen nivel competitivo, debido a que aunque haya un mayor número de universidades no se aseguraba que la educación sea de calidad.
- Ninguna universidad peruana se encontraba en las 30 primeras universidades del ranking de las 100 mejores universidades latinoamericanas en el año 2010.(Quacquarelli Symonds, 2010)

Según la publicación de SINEACE (Yanada, Rivera, & Castro, 2012), el Perú se encontraba por debajo del rango promedio con lo que respecta a calidad educativa en comparación con otros países.

- Se menciona una encuesta realizada el año 2010 denominada ENHAB (Encuesta Sobre Habilidades y Funcionamiento del Mercado Laboral Peruano), en el cual se manifestaba un deterioro en la calidad del sistema universitario debido a que el número de profesionales insatisfechos con sus estudios superiores se había incrementado.
- Se cita como uno de los indicadores, que la brecha salarial del mejor pagado y el peor pagado entre los profesionales en Perú era de 10 a 1, mientras que en Chile la proporción era sólo de 3 a 1, lo cual muestra una brecha muy grande de diferencia y era un índice negativo para la educación universitaria en nuestro país.
- En lo que respecta al entorno laboral, según el estudio de Manpower, mencionaba que el 42% de las empresas peruanas se encontraban con dificultades para encontrar un profesional con las características requeridas, siendo este porcentaje más alto que el promedio de América Latina (34%). Otro aspecto que indica que la calidad de educación superior en el Perú no se encontraba a nivel de la educación en otros países es que según la ENAHO (Encuesta Nacional de Hogares) la tasa de subempleo (que viene a ser otra ocupación diferente a la que se dedica una persona) tuvo un porcentaje muy elevado.

En el año 2013, se encontraban solo tres universidades peruanas dentro de las 100 primeras universidades en Latinoamérica las cuales fueron la Pontificia Universidad

Católica del Perú en el puesto 30, la Universidad Nacional Mayor de San Marcos en el puesto 57 y la Universidad Peruana Cayetano Heredia en el puesto 65.(Quacquarelli Symonds, 2013)

Según el informe de (INEI, 2015), en la cual se realizó una encuesta nacional a egresados universitarios y universidades se encontraron los siguientes indicadores:

- Solo el 8.2 % de las universidades encuestadas contaban con una certificación ISO, la cual es la base del sistema de gestión de la calidad, esta es una norma internacional que se centra en todos los elementos de administración de calidad con los que una universidad debería contar para mejorar la calidad de la educación.
- El 23% de egresados no se encuentran laborando con empleos relacionados a su carrera.
- El 37.3% de egresados considera que los servicios otorgados por la universidad no fueron buenos.
- De las 122 universidades encuestadas solo 8 contaban con acreditación por SINEACE.
- El 24.4% de egresados mencionan que el desempeño académico no fue bueno.
- Según el 67.4% de los encuestados comentan que los trámites administrativos no fueron buenos.

En el año 2015, se seguían contando con 3 universidades peruanas dentro del ranking de las 100 mejores universidades de Latinoamérica, se encontraban la Pontificia Universidad Católica del Perú en el puesto 21, la Universidad Nacional Mayor de San Marcos en el puesto 70 y la Universidad Peruana Cayetano Heredia en el puesto 74; aun así se debe resaltar a la Pontificia Universidad Católica del Perú que subió en la escala de posiciones perteneciendo así a las 30 mejores universidades latinoamericanas en cambio las otras universidades bajaron.(Quacquarelli Symonds, 2015)

En el año 2017, no varío mucho las posiciones de las universidades con respecto al 2016, ya que seguían solo 3 universidades peruanas en el top 100 de mejores universidades latinoamérica.(Quacquarelli Symonds, 2017)

Para el año 2019, la Pontificia Universidad Católica del Perú se coloca en el puesto 18, es decir entre los 20 mejores de Latinoamérica, y la Universidad Nacional Mayor de San Marcos sube cinco posiciones. (Quacquarelli Symonds, 2019)

Para el año 2020, la Pontificia Universidad Católica del Perú se coloca en el puesto 16, mejor que el año pasado y se mantiene en las 20 mejores de Latinoamérica, por otro lado, hay 4 universidades en el top 100 de las mejores universidades de latinoamérica. (Quacquarelli Symonds, 2020)

Ante la problemática de las universidades peruanas, se propone mejorar la calidad educativa en el año 2006 (SINEACE, 2016), se creó la ley 28740 (Ley de Creación del Sistema Nacional de Evaluación, Acreditación y Certificación de la Calidad - SINEACE). Su función era la de apoyar a las instituciones a conseguir la acreditación mediante la evaluación de éstas. En su reglamento se menciona la aseguración de los

servicios de calidad de las instituciones educativas (colegios, universidades, entre otros) ya sean públicas o privadas. En el año 2014, se creó la Ley Universitaria (LeyN° 30220) que declaraba en uno de sus puntos la reorganización del SINEACE, en esta ley se creó un nuevo modelo de acreditación, basada en una matriz de estándares.



Figura 1. Relación de dimensiones y factores de modelo de acreditación de programas de estudios universitarios (SINEACE, 2016)

El modelo de acreditación comprende las dimensiones que se encuentran en color rojo y los demás son los factores (véase Figura 1), cada dimensión contiene varios factores, de esta matriz de estándares se resaltan los siguientes estándares relacionados al control y seguimiento académico de alumnos y profesores:

- El estándar 14 (Selección, evaluación, capacitación y perfeccionamiento) corresponde al factor gestión de docentes y a la dimensión formación integral, estemencióna que el programa de estudios debe evaluar el desempeño del personal docente con el fin de identificar necesidades para capacitaciones. El perfeccionamiento se refiere a actualización, innovación pedagógica y manejo de tecnologías de información de los profesores.
- El estándar 20 (Seguimiento al desempeño de los estudiantes) corresponde al factor seguimiento al estudiante y a la dimensión formación integral, estemencióna que el programa de estudios debe contemplar el seguimiento del desempeño de los estudiantes y ofrece apoyo como servicios de tutoría o reforzamiento en cursos.
- El estándar 30 (Sistema de información y comunicación) corresponde al factor infraestructura y a la dimensión soporte institucional, estemencióna que el programa de estudios debe tener implementado un sistema de información y comunicación que apoye a la gestión académica y administrativa.

La calidad académica en las universidades puede mejorar mientras se puedan cumplir con los estándares definidos por el organismo de acreditación. A continuación, se

presentan estudios relacionados a los estándares mencionados anteriormente con estudios que reflejan su relación con el desempeño académico de alumnos y del docente:

- Relacionado al estándar 14, sobre el seguimiento del desempeño de docentes, indica que existe un buen porcentaje de alumnos encuestados que consideran que el desempeño académico de los docentes es bajo entre los puntos mencionados se encuentran los siguientes: identificar intereses de estudiantes, promover participación de los intereses de los estudiantes, conocimientos en fundamentos teóricos y/o tecnológicos. Como conclusión en base a indicadores de las encuestas que realizó se tiene una relación significativa entre el desempeño docente universitario y la satisfacción de los estudiantes del programa (Tolentino, 2014).
- Relacionado al estándar 20, sobre el seguimiento del desempeño de estudiantes: Hay un importante factor que es la ejecución curricular (control o seguimiento del sílabo o currículo del curso). Se tomó como objeto de estudio los resultados de las encuestas aplicadas a los alumnos del primer año de estudio de la Facultad de Odontología de la UNMSM, en donde la autora hace énfasis en la dependencia existente entre el cumplimiento continuo del sílabo con el rendimiento académico, ella menciona lo siguiente: “Existe relación directa entre la ejecución curricular y el rendimiento académico (...) donde la percepción positiva de los estudiantes sobre la ejecución curricular se correlaciona con un mejor rendimiento académico”(Moromi, 2002). Otro factor importante relacionado al estándar 20, viene a ser la asistencia a clases del alumnado universitario el cual es importante para obtener un rendimiento académico esperado. Se realiza un estudio donde se toma como fuente de datos los registros de asistencias y los resultados del examen final del curso de metodología de investigación. Se logra comprobar a través de técnicas estadísticas, que existe una dependencia entre las siguientes variables: asistencia a clases (teóricas, prácticas) y superación de la asignatura. Se concluye del estudio realizado, que la asistencia a clases tanto práctica como teórica influye en la superación o mejor rendimiento académico en los universitarios(Rodríguez & Herrera, 2009).
- Relacionado al estándar 30, se menciona que las instituciones de educación superior han tenido cambios con la incorporación de las tecnologías de información. Existe una necesidad de que se usen estrategias tecnológicas innovadoras para que las universidades realicen procesos de mejora de calidad apoyados en las tecnologías de información. Menciona que según Cukierman refiere a que los alumnos están acostumbrados a usar estas tecnologías de forma casi innata y que se debería aprovechar su uso para acceder a la información y al conocimiento(Kuz, Falco, & Giandini, 2016).

La inserción de las tecnologías de información en las universidades ha supuesto una transformación y optimización de procesos administrativos, de la gestión y de procesos de enseñanza-aprendizaje. También menciona que la integración de éstas proporciona la mejora del proceso educativo(Álvarez & Mayo, 2009).

De acuerdo a (SINEACE, 2020), la escuela profesional de obstetricia de la UNMSM se encuentra en proceso de autoevaluación la cual es realizada por ésta misma y se basa en el matriz de estándares y en la guía de evaluación, y como se menciona en (De la Cruz, Espinoza, & Cuba, 2019), de acuerdo alas reuniones con la directora de escuela en la Escuela Profesional de Obstetricia - Facultad de Medicina de la UNMSM se desea mejorar la calidad educativa siguiendo los estándares mencionados de la matriz de estándares de la ley universitaria del SINEACE, los cuales son: un adecuado seguimiento al estudiante, evaluación al docente y el de contar con un sistema informático.

Es importante mencionar que los alumnos de 2.^º a 5.^º año si llevan los cursos en su escuela mientras que los alumnos de 1.^{er} año llevan los cursos en Estudios Generales (EEGG).

Se realizaron encuestas (véase Anexo 1) a los alumnos de obstetricia que cuentan con 1 año de experiencia en los cursos impartidos por la Escuela Profesional de Obstetricia (EPO), es por ello que se delimitaron las encuestadas alumnos de 3.^{er}, 4.^º, 5.^º año académico y alumnos egresados menores de un año, ya que todos los mencionados cumplen con esa condición sin excepción.

Esta encuesta tiene un tamaño muestral probabilístico de 219 estudiantes de una población de 364 estudiantes, esta cantidad de muestras satisface el tamaño mínimo que requiere el nivel de confianza de 95% y margen de error de 5%.

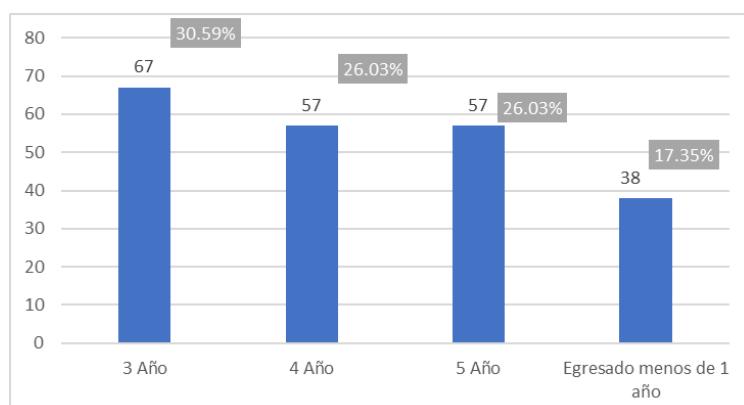


Figura 2. Gráfica lineal de pregunta: ¿En que año que se encuentran cursando actualmente o indicar si es egresado? (Elaboración propia)

Para resumir, se mencionan los resultados de las preguntas de la encuesta realizada por los alumnos:

- De acuerdo con la gráfica (véase Figura 3), donde 1 es “muy cerca” y 5 es “muy lejos”, alrededor del 90.8% de los encuestados considera que no viven cerca de su campus universitario y/o hospitales.

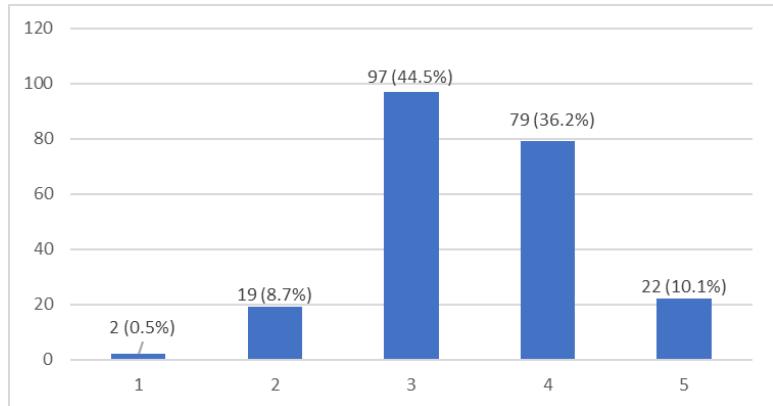


Figura 3.Gráfica lineal de pregunta: ¿Qué tan lejos viven los alumnos?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 4), donde 1 es “muy pocas” y 5 es “todas sin excepción”, el 85.84% de los encuestados considera que no se llegó a cumplir todas las clases del syllabus de los cursos.

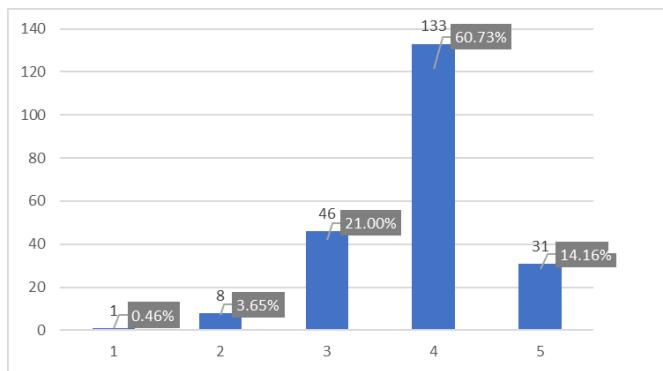


Figura 4.Gráfica lineal de pregunta: ¿(Cumplen/Cumplieron) los profesores con todas las clases del syllabus?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 5), donde 1 es “muy fácil” y 5: “muy difícil”, el 75.8% de los encuestados considera que no es fácil llevar el control y seguimiento de las clases avanzadas por el profesor.

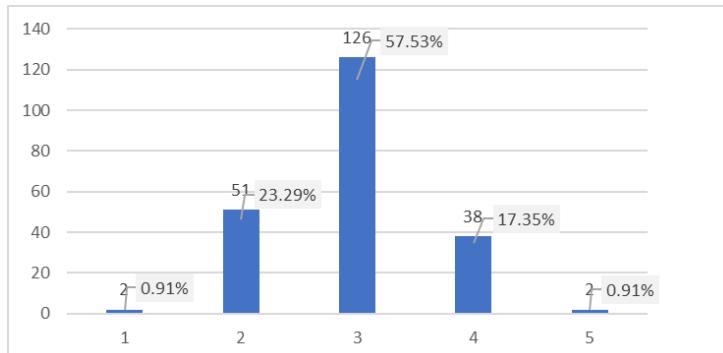


Figura 5.Gráfica lineal de pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las clases avanzadas por el profesor?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 6), donde 1 es “muy fácil” y 5 es “muy difícil”, el 63.47% de los encuestados considera que no es fácil llevar el control de sus asistencias de clases en los cursos.

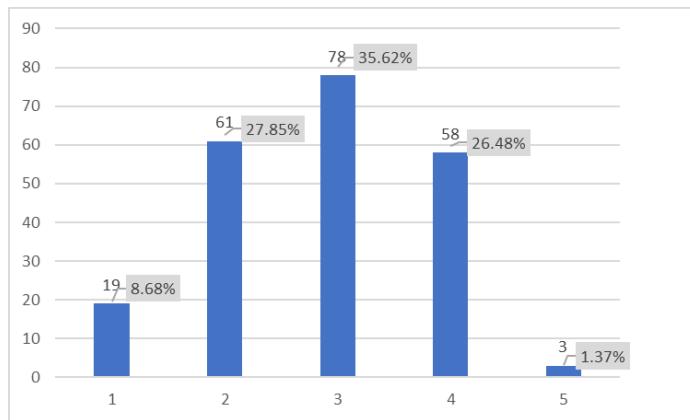


Figura 6.Gráfica lineal de pregunta: ¿Qué tan difícil (es/fue) llevar un control de asistencias que has realizado en tus cursos?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 7), donde 1 es “muy poco” y 5 es “demasiado”, el 86.30% de los encuestados considera que el retraso en el tiempo que se toma en enterarse de sus notas es regular a demasiado.

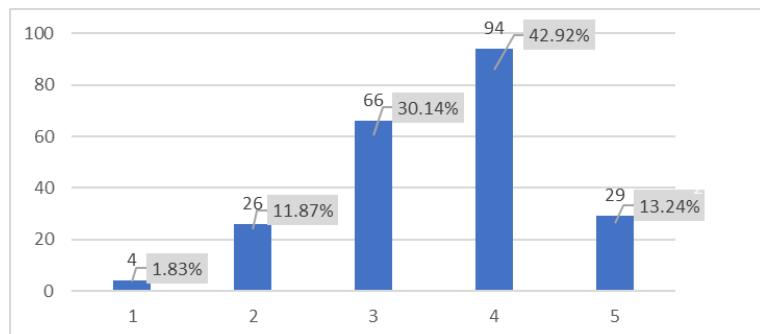


Figura 7.Gráfica lineal de pregunta: ¿Considera que (hay/hubo) retrasos en el tiempo en el que te enterabas de tus notas?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 8), donde 1 es “muy fácil” y 5 es “muy difícil”, el 85.39% de los encuestados considera que no es fácil llevar el control de sus notas en los cursos.

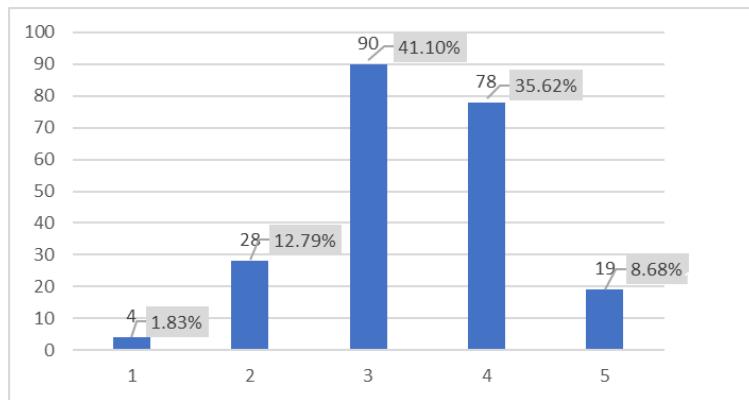


Figura 8. Gráfica lineal de pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las notas que has obtenido en tus cursos? (Elaboración propia)

- De acuerdo con la gráfica (véase Figura 9), donde 1 es “muy poco” y 5 es “bastante”, el 84.48% de los encuestados considera que es mejor evaluar al profesor por cada clase dado que se puede obtener una mejor retroalimentación.

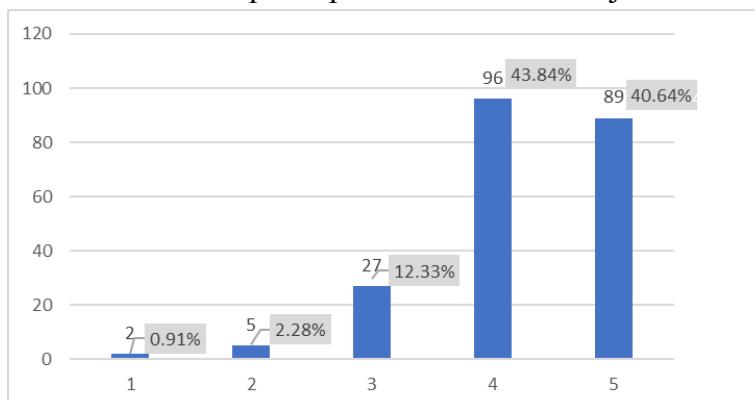


Figura 9. Gráfica lineal de pregunta: ¿Cree usted que es mejor evaluar al profesor por cada clase en vez de hacerlo una vez cada semestre, de tal forma que se pueda obtener una retroalimentación más rápida y realista? (Elaboración propia)

- De acuerdo con la gráfica (véase Figura 10), donde 1 es “no demora” y 5 es “demora mucho”, el 66.21% considera que existe demora en resolver sus dudas.

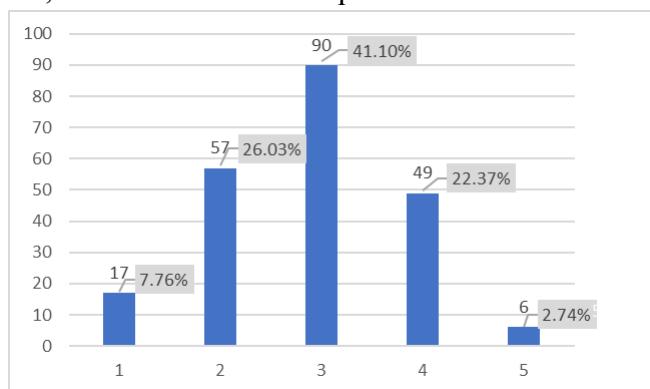


Figura 10.Gráfica lineal de pregunta: ¿Demora/Demoraba mucho en resolver las dudas sobre el curso?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 11), donde 1 es “muy poco” y 5 es “demasiado”, el 75.34% de los encuestados considera que la demora en enterarse de los anuncios que realiza el profesor es regular a alta.

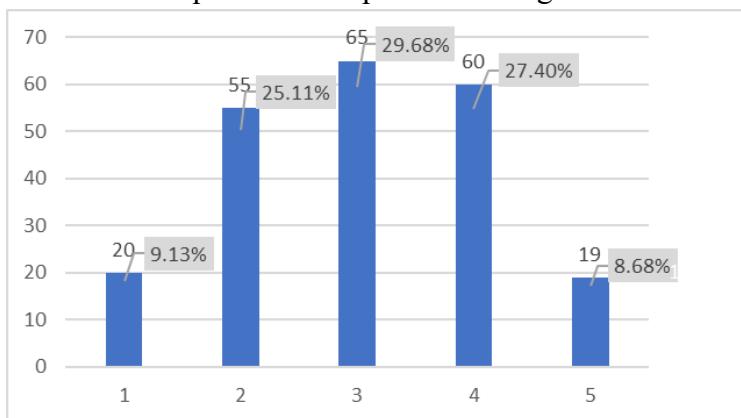


Figura 11.Gráfica lineal de pregunta: ¿Cuánto demora/demoraba en enterarse de los anuncios del profesor tales como postergación de clases, reprogramación o cualquier otro relacionado al curso?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 12), donde 1 es “demasiada disponibilidad” y 5 es “muy poca disponibilidad”, el 92.24% de los encuestados considera que no hay buena disponibilidad de la información de los cursos como asistencias o notas, no se puede consultar al instante.

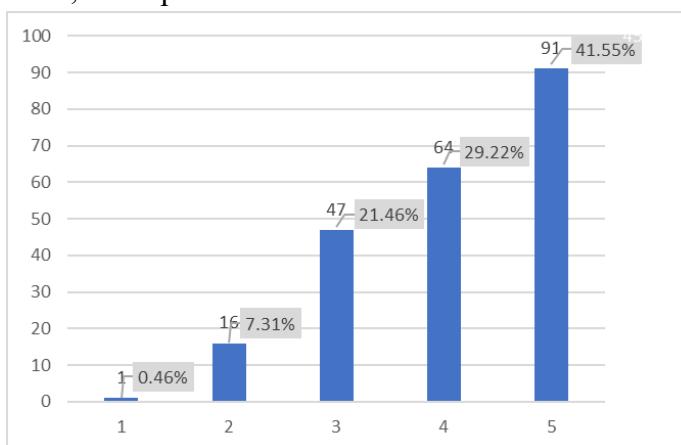


Figura 12. Gráfica lineal de pregunta: ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizadas disponible para poder consultar en cualquier momento(al instante)?(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 13), donde 1 es “muy poco” y 5 es “demasiado”, el 86.30% de los encuestados considera que la información de forma regular a alta es descentralizada y desorganizada como uso de papel, archivos excel, sistemas web, redes sociales.

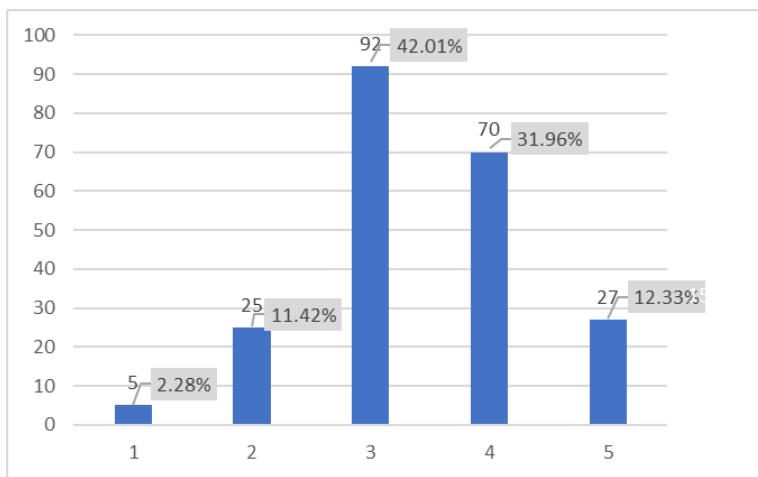


Figura 13. Gráfica lineal de pregunta: ¿Considera que la información del curso se encuentra descentralizada y desorganizada? Por ejemplo: el uso del papel, archivos excel, sistemas web, redes sociales.(Elaboración propia)

- De acuerdo con la gráfica (véase Figura 14), 97.72% considera que el manejo de la información sería más útil en celulares que en aplicativos webs.

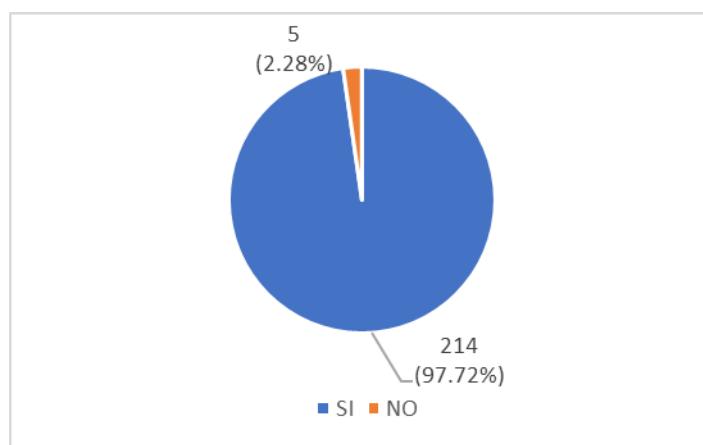


Figura 14. Gráfica circular de pregunta: ¿Consideraría que un aplicativo móvil(celular) que permita el manejo de información de los cursos sería más útil que el uso del papel o de sistemas que solo funcionen en web?(Elaboración propia)

A modo de resumen de los gráficos por cada pregunta de las encuestas y además de reuniones que se tuvo con la directora de escuela Clara Díaz, se presentan las siguientes dificultades vinculadas al control académico de la Escuela Profesional de Obstetricia:

- La información de las clases como: el registro de notas, la toma de asistencia a alumnos se registra en diferentes medios de información como: papel físico, archivo excel u otros, no se cuenta con un sistema centralizado que permita contar con dicha información de inmediato, esto causa que se generen retrasos en conseguir y entregar la información entre el profesor y los alumnos.

- Por lo general, no se llega a cumplir todos los temarios del sílabo de los cursos en su totalidad, esto debido a feriados, ausencia del profesor o simplemente que el profesor no haya podido culminar con todos los temas del syllabus.
- En la actualidad, solo algunos profesores acuden a aplicaciones informáticas (correo, facebook, whatsapp, etc.) para comunicar avisos a su delegado y/o alumnos, tales como registro de notas, falta a clase por enfermedad, postergación de clase por feriado u otro acontecimiento que es complicado para los alumnos de saber en un momento dado, muchas veces el alumno debe indagar por otros medios para poder enterarse, lo cual causa pérdida de tiempo y genera malestar en ellos.
- No se llega a medir la satisfacción de los alumnos por cada clase impartida del profesor, se hizo mención de que ésta se podría realizar mediante encuestas por cada clase.
- El resumen de la información impartida de las clases como la ejecución curricular (sílabo), las asistencias de alumnos y profesores, y las notas viene a ser tediosa de manejar para la dirección de escuela, esta información normalmente se registra a fin de ciclo por lo que también para la dirección de escuela sería importante contar con esta información en tiempo real en forma de estadísticas.
- Se mencionó que cuentan con sistemas informáticos web pero que no se llegan a usar mucho debido a que no en todos los salones se cuentan con computadoras menos en los locales externos a las facultades, pero si comentó que la mayoría de los profesores contaba con smartphones y sería más útil el uso de aplicativos móviles.

Para comprender mejor la problemática mencionada de la Escuela Profesional de Obstetricia – Facultad de Medicina de la UNMSM se presenta un árbol de problemas, el cual se encuentra detallado (véase Figura 15). Esta técnica permite reconocer el problema principal, sus causas y sus efectos.

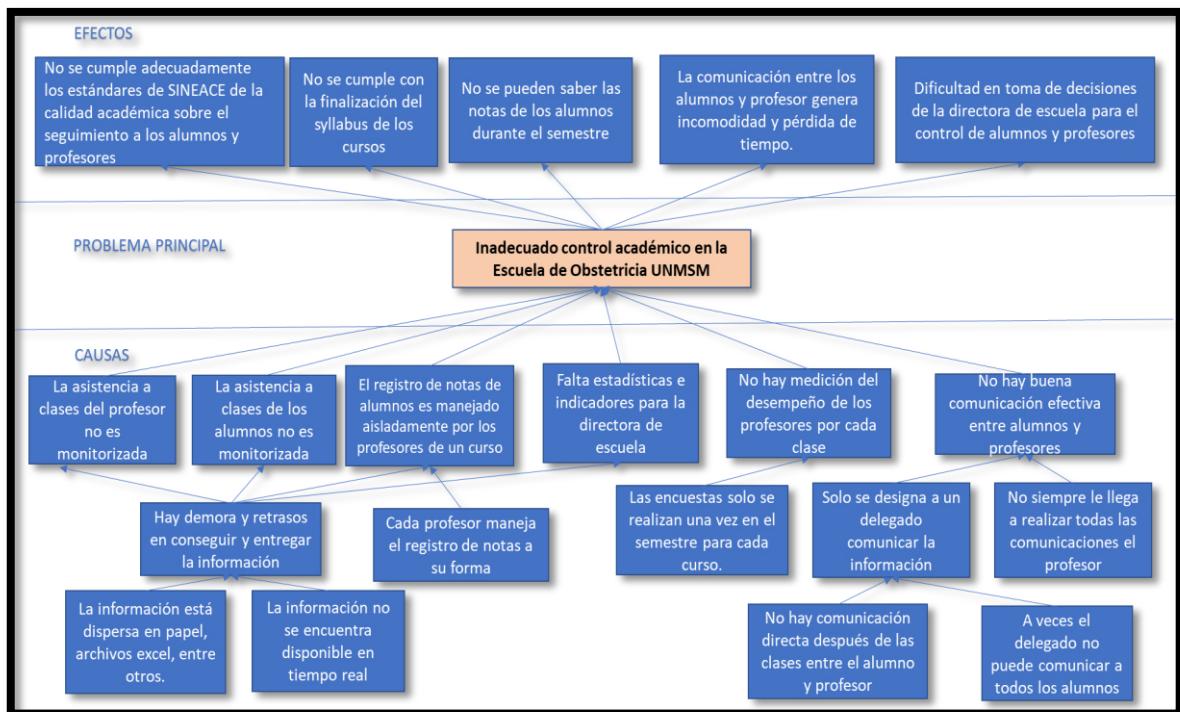


Figura 15.Árbol de problemas del inadecuado control académico en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia)

Además, se realizó el proceso AS-IS (Tal como está) ya que ayuda a generar un entendimiento de como se está realizando el proceso de negocio, del mismo modo permite establecer puntos críticos del proceso.

Para la visualización panorámica del proceso de negocio se hace uso del software Bizagi Modeler que nos permite plasmarlosintéticamente a través de un diagrama.

Se muestra el diagrama de proceso AS-IS del registro y consulta de asistencias de los alumnos y profesores de la escuela profesional de obstetricia de la Facultad de Medicina de la UNMSM (véaseFigura 16). En la cual se puede evidenciar que existen los siguientes problemas:

- El alumno no sabe sus asistencias, ni sus calificaciones, ya que no se encuentran disponibles a través de un medio informativo. Ocasionalmente el profesor lo deriva al delegado del grupo, lamentablemente este actúa de manera pasiva (en su gran mayoría), lo cual produce una ralentización en la comunicación.
- El registro de asistencias y notas es realizado manualmente a través de papel físicopara cada profesor del curso.
- Existe riesgo de pérdida de información ya que ésta se encuentra en papelfísico.
- La dirección de escuela carece de indicadores y/o estadísticas sobre las asistencias, calificaciones o cualquier otro de índole académico;los cuales no favorecen a la rápida toma de decisiones, por conseciente, tomar las medidas correspondientes que amerite el caso.

- El uso de papel fisicono conlleva a mantener un ecosistema sostenible.

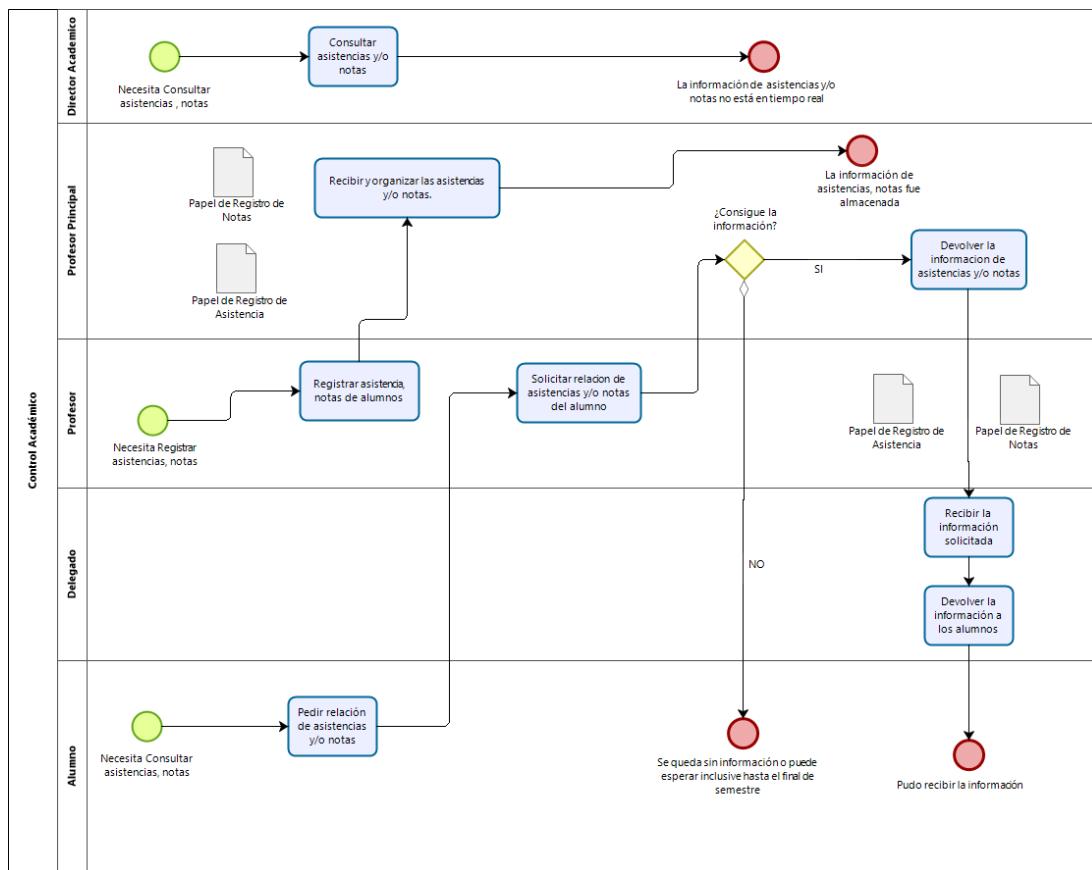


Figura 16.Diagrama de procesos AS-IS del registro y consulta de asistencias y notas en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM.(Elaboración propia)

Se muestra el diagrama de procesos AS-IS de los anuncios del profesor hacia los alumnos de su curso dentro de la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (véase Figura 17). En la cual se puede evidenciar que existen los siguientes problemas:

- Existen retrasos en la recepción de anuncios realizados por los profesores hacia los alumnos.
- Se depende mucho del delegado para emitir los anuncios.
- Los medios de comunicación podrían no ser los más adecuados.
- No existe un único canal por el cual los estudiantes se enteren de los avisos realizados por el docente.

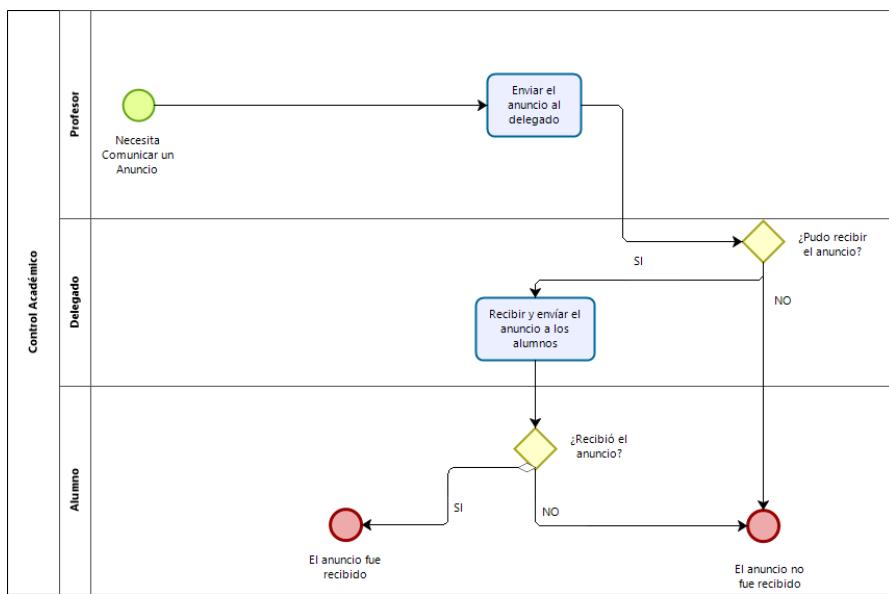


Figura 17. Diagrama de procesos en base a proceso AS-IS de los anuncios del profesor para los alumnos de su curso en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia)

Se muestra el diagrama de procesos de las preguntas y respuestas de los alumnos y profesores de un curso en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM (véase Figura 18). En la cual se puede evidenciar que existen los siguientes problemas:

- No hay una comunicación efectiva entre los docentes hacia los estudiantes y viceversa.
- No todas las dudas de los alumnos son resueltas durante una clase impartida.
- No existe un único medio por el cual los alumnos puedan expresar sus dudas respecto a una clase impartida.

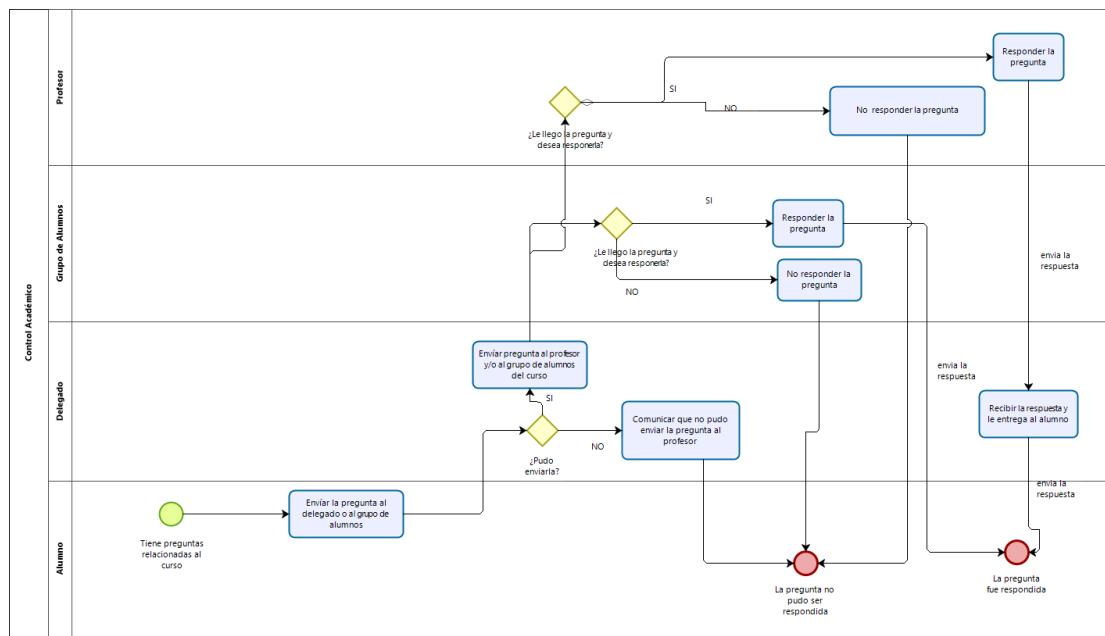


Figura 18. Diagrama de procesos en base a proceso AS-IS de las preguntas y respuestas de los alumnos y profesores de un curso en la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM. (Elaboración propia)

1.2 Definición del problema

El problema identificado es que existe un inadecuado control académico en la Escuela Profesional de Obstetricia – Facultad de Medicina de la Universidad Nacional Mayor de San Marcos.

Problemas secundarios

- No hay un adecuado control de la asistencia a clases del profesor.
- No hay un adecuado control de la asistencia a clases de los alumnos.
- No hay un adecuado control de las notas de los alumnos.
- No se cuenta con estadísticas e indicadores para la dirección de escuela en tiempo real.
- No se cuenta con la medición del desempeño de los profesores por clase.
- No hay un medio de interacción para que alumnos y profesores se puedan comunicar.
- La información no se encuentra disponible en línea.
- La información está descentralizada y desorganizada.
- Se tarda mucho en la búsqueda de información.

1.3 Objetivos

En base al análisis del árbol de los problemas se hace la creación del siguiente árbol de objetivos el cual plasma los objetivos principales y secundarios que se quieren llevar a cabo (véase Figura 19).

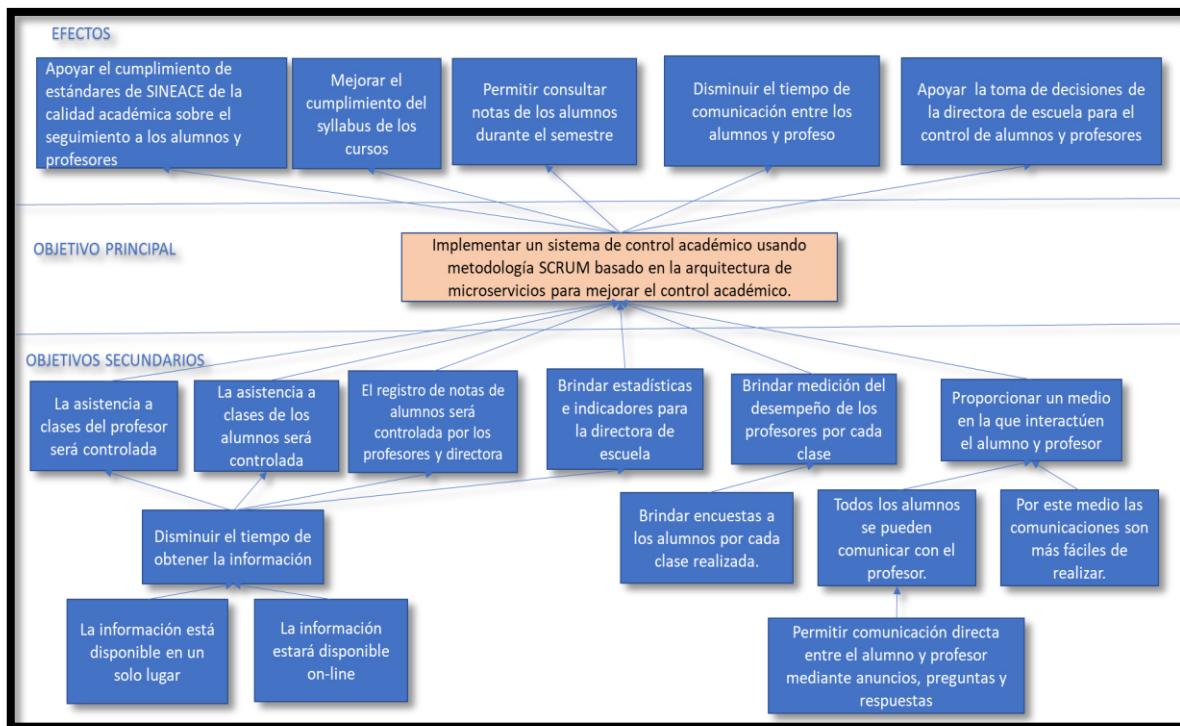


Figura 19. Diagrama delárbol de objetivos (Elaboración propia)

Objetivo principal

Desarrollar un aplicativo móvil que use la arquitectura de microservicios en base a la metodología de desarrollo SCRUM que pueda mejorar el control académico. Caso: Escuela Profesional de Obstetricia - Facultad de Medicina de la Universidad Nacional Mayor de San Marcos.

Objetivos secundarios

- Permitir el control de la asistencia a clases del profesor.
- Permitir el control de la asistencia a clases de los alumnos.
- Permitir el control de las notas de los alumnos.
- Brindar estadísticas e indicadores para la dirección de escuela
- Brindar medición del desempeño de los profesores por clase al permitir que los alumnos realicen encuestas por cada clase.
- Proporcionar un medio de interacción para que alumnos y profesores se puedan comunicar mediante anuncios, preguntas y respuestas.
- Brindar información disponible en línea.
- Organizar y centralizar la información en el sistema para todos los interesados.
- Reducir el tiempo de búsqueda de información.

1.4 Justificación

Este trabajo se enfoca en realizar un software que permita mejorar la eficacia y eficiencia del control académico de alumnos y profesores de la Escuela Profesional de

Obstetricia de la Facultad de Medicina de la UNMSM, con el fin de poder mejorar su calidad académica de acuerdo con los estándares de acreditación del SINEACE en las que se mencionan el control de alumnos y profesores y el poder contar con un sistema informático que apoye a la gestión académica.

1.5 Alcances y limitaciones

1.5.1 Alcances

- La solución propuesta será usada solo por alumnos, profesores y la dirección de escuela de la Escuela Académica Profesional de Obstetricia de la Facultad de Medicina de la UNMSM.
- La solución se desarrollará en el último semestre activo.
- Los usuarios tendrán acceso a notificaciones en tiempo real.
- El aplicativo se desarrollará para móviles Android e IOS.
- Se realizarán pruebas internas en el desarrollo de software y pruebas con los usuarios para validar las funcionalidades del software.
- El sistema de información abarcará los siguientes puntos:
 - Visualización de los cursos designados en el semestre para los alumnos y profesores.
 - Registro de asistencias de alumnos y profesores en las clases impartidas del curso
 - Registro de encuesta de los alumnos por cada clase impartida del curso.
 - Registro y visualización de las notas en el curso.
 - Registro y visualización de los anuncios en el curso.
 - Registro y visualización de preguntas y respuestas en el curso.
 - Visualización de estadísticas para la dirección de escuela.

1.5.2 Limitaciones

Limitación espacial:

El aplicativo de control académico cubre geográficamente el Perú debido a que abarca el área de la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM, las prácticas en diferentes hospitales y los internados se realizan en diferentes ciudades.

Limitación social

El aplicativo de control académico beneficiará a los alumnos (2.^º a 5.^º año), profesores y la dirección de escuela de la Escuela Profesional de Obstetricia de la Facultad de Medicina de la UNMSM.

Limitación técnica

No aplica, ya que se cuentan con los conocimientos adecuados sobre hardware y software sobre la arquitectura de microservicios y de reactnative.

1.6 Organización de la tesis

La tesis que se presenta está organizada de la siguiente manera, en el capítulo II se describe el marco teórico (Definición de términos, metodologías de desarrollo, arquitectura de software). En el capítulo III, se hace enfoque en el estado del arte

(Taxonomía, artículos, tesis, tecnología, casos de éxito y benchmarking). En el capítulo IV, se presenta la resolución del problema (cuadros comparativos y la descripción de la solución tecnológica). En el capítulo V, se presentan las conclusiones.

2 MARCO TEÓRICO

2.1 Definición de términos

2.1.1 Control

Control:Significa la comprobación, inspección, fiscalización, intervención. Regulación manual o automática, sobre un sistema(Real Academia Española, 2001).

Según (Koontz, Weihrich, & Cannice, 2012), el control es medir y corregir las actividades de los subordinados para asegurar que los eventos se ajusten a los planes.De acuerdo con(Haimann, 2005)el control es el proceso de verificación para determinar si se están cumpliendo los planes o no, si existe un progreso hacia los objetivos y metas. (Zona Económica, 2019)

Se puede resumir al control como un proceso que verifica y corrige si los planes van de acuerdo con lo establecido.

2.1.2 Sistemas de control de gestión

(Horngren, Datar, & Foster, 2000), definen al sistema de control de gestión como “Un sistema que adquiere y usa información que ayuda a la coordinación del planeamiento y las decisiones del control organizacional con el objetivo de mejorar las decisiones colectivas dentro de la organización”.(De la Cruz, Espinoza, & Cuba, 2019)

Por su parte(Abernethy & Chua, 1996), indican que es un sistema que combina mecanismos de control diseñados e implementados por los administradorespara incrementar la probabilidad de que los líderes de la organización se comporten de una manera consistente con los objetivos de esta.(De la Cruz, Espinoza, & Cuba, 2019)

Por lo tanto, menciona (Beuren & Teixeira, 2014) que un Sistema de control de Gestión debe ser capaz de medir y gestionar el desempeño organizacional, alineados con la generación de valor para la organización, particularmente dirigido a grupos que desempeñen un papel estratégico. La necesidad de controlar el desempeño basado en los flujos de información consistente es la motivaciónde ser en los sistemas de control de gestión.(De la Cruz, Espinoza, & Cuba, 2019)

(Beuren & Teixeira, 2014) menciona que la evaluación del desempeño organizacional es trascendental para cualquier tipo de organización, uno de los motivos más importantes por el que las organizaciones deben implementar un sistema de evaluación y control de gestión de sus recursos humanos, es para saber si sus trabajadores están efectivamente contribuyendo al logro de los objetivos institucionales establecidos (Sánchez, 2011), al ser un conjunto de medidas que tratan de estar alineadas con la estrategia de la organización.(De la Cruz, Espinoza, & Cuba, 2019)

Además, (Beuren & Teixeira, 2014) menciona que se busca la satisfacción de los stakeholders e influye en todos los actores del negocio a mejorarsus actividades integrando la estrategia, losprocesos y los recursos.(De la Cruz, Espinoza, & Cuba, 2019)

(Beuren & Teixeira, 2014) menciona que una de las funciones de la evaluación de desempeños es la de proveer información valiosa para la toma de decisiones, siendo este el apoyo principal para el proceso de planeamiento y control, así como mantenerse alineados con los objetivos trazados de la organización. Otra importante funcionalidad es la “señalización”, esto quiere decir, mostrar a los empleados la importancia de los aspectos estratégicos establecidos y brindar información financiera a los stakeholders, tales como la innovación, operaciones, niveles de satisfacción del cliente, entre otros. (De la Cruz, Espinoza, & Cuba, 2019)

Como está mencionado en (De la Cruz, Espinoza, & Cuba, 2019), para (Ferreira & Otley, 2006) una forma de evaluar a los sistemas de control de gestión es a través del framework “Gestión del desempeño y control”, cuya finalidad es capturar la estructura y funcionalidad de los sistemas de control de gestión. A continuación, (véase Tabla 1), se listan los 12 tópicos centrales y las preguntas con las que éstas son formuladas:

Tabla 1
Tópicos centrales y preguntas de sistema de control de gestión.

Tópicos	Preguntas
Misión y visión	¿Cuál es la visión y misión de la organización y como se llama la atención de los gerentes y empleados? ¿Qué mecanismos, procesos y redes son usados para transmitir los propósitos y objetivos generales de la organización hacia sus miembros?
Factores claves de éxito	¿Cuáles son los factores claves que se consideran fundamentales para el éxito general de la organización y como les llama la atención a los gerentes y empleados?
Planes y estrategias:	¿Cuál es la estructura de la organización y que impacto tiene en el diseño y uso de los sistemas de gestión del rendimiento? ¿Cómo influye y como está influenciado por el proceso de gestión estratégica?
Estructura de la organización	¿Cuáles son las medidas de desempeño clave de la organización?
Indicadores claves del desempeño:	¿Cuáles son las medidas de desempeño clave de la organización que se derivan de sus objetivos, factores clave de éxito y estrategia y planes? ¿Cómo se especifican y qué papel juegan en la evaluación de desempeño? ¿Existen omisiones significativas?
Establecimiento de objetivos	¿Qué nivel de desempeño debe alcanzar la organización para cada una de sus medidas de desempeño clave (identificadas en la pregunta anterior), como se encarga de establecer los objetivos de desempeño apropiados para ellos y que tan difíciles son esos objetivos?
Evaluación del desempeño	¿Qué procesos, si los hay, sigue la organización para evaluar el desempeño individual, grupal y organizacional? ¿Las evaluaciones de desempeño son principalmente objetivas, subjetivas o mixtas y que tan importantes son la información

		y los controles formales e informales en estos procesos?
Sistema de recompensas		¿Qué recompensas financieras y/o no financieras, obtendrán los gerentes y otros empleados al alcanzar los objetivos de desempeño u otros aspectos evaluados del desempeño (o, a la inversa, que sanciones sufrirán al no lograrlos)?
Flujo de información		¿Qué flujos específicos de información (retroalimentación y avance), sistemas y redes tiene implementada la organización para respaldar el funcionamiento de sus sistemas de gestión de desempeño?
Tipos de uso de los flujos		¿Qué tipo de uso se hace de la información y de los diversos mecanismos de control establecidos? ¿Se pueden caracterizar estos usos en términos de diversas tipologías en la literatura? ¿Cómo difieren los controles y sus usos en diferentes niveles jerárquicos?
Fuente: (De la Cruz, Espinoza, & Cuba, 2019), adaptado de (Ferreira & Otley, 2006) y citado en(Beuren& Teixeira, 2014).		

2.1.3 Calidad educativa

Según la ley de educación 28044, define la calidad educativa como el "nivel óptimo de formación que deben alcanzar las personas para enfrentar los retos del desarrollo humano". De una manera parecida la OREALC(Oficina Regional de Educación para América Latina y el Caribe) menciona que la calidad en educación como un medio para que el ser humano se desarrolle plenamente y gracias a ella crezca y se fortalezca como persona que contribuya al desarrollo de la sociedad transmitiendo y compartiendo sus valores y cultura.(SINEACE, 2016)

2.1.4 Acreditación y la mejora continua

La acreditación tiene como objetivo cumplir los estándares o requisitos establecidos por el organismo acreditador, se puede entender como el reconocimiento público al cumplimiento de estándares por parte de una institución. La mejora continua se da por medio de la autoevaluación de la organización de una institución, debe ser tal que pueda impulsar los cambios y mejoras, estos no son generales sinomás bien particulares de acuerdo con cada institución.(SINEACE, 2016)

2.1.5 Matriz de estándares de acreditación de Perú

Esta matriz fue creada por la SUNEDU la cual se conforma de 4 categorías generales o dimensiones: la dimensión central que viene a ser la formación integral, la dimensión de gestiónestratégica, dimensión de soporte institucional y la última es la dimensión de resultados. (SINEACE, 2016). A continuación, se muestran las dimensiones:

- Dimensión de gestiónestratégica: Evalúa la planificación y la conducción de la institución o programa de estudios, tomando en consideración el uso de la información para la mejora continua.

- Dimensión de formación integral: Es el eje central. Evalúa el proceso de enseñanza y aprendizaje a los alumnos, el soporte a los estudiantes y docentes, como los procesos de investigación y responsabilidad social.
- Dimensión de soporte institucional: Evalúa los aspectos relacionados con la gestión de recursos, infraestructura y el soporte para lograr el bienestar de los miembros de la institución educativa
- Dimensión de resultados: Verifica los resultados del aprendizaje y el perfil de egreso como también los objetivos educacionales de la institución.

2.2 Metodología de desarrollo

2.2.1 Metodología RUP

El Proceso Unificado Racional (RationalUnifiedProcess en inglés, y sus siglas RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (Fernandez & Cadelli, 2014)

RUP no es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí.(Fernandez & Cadelli, 2014)

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto RUP como proceso de desarrollos explícito en la definición de software y su trazabilidad, es decir, contempla en relación causallos programas creados desde los requerimientos hasta la implementación y pruebas. RUP identifica claramente a los profesionales (actores) involucrados en el desarrollo del software y sus responsabilidades en cada una de las actividades.(Fernandez & Cadelli, 2014)

2.2.2 Metodología SCRUM

Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Pone su atención y hace foco sobre valores y prácticas de gestión, en vez de requerimientos, prácticas de desarrollo, implementación y otras cuestiones técnicas. Esta metodología delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles, es decir, que es flexible y los integrantes del equipo pueden optar por organizar la forma de interactuar entre ellos. La terminología “Scrum” procede del deporte llamado rugby, donde se designa al acto de preparar el avance del equipo en unidad pasando la pelota a uno y otro jugador. (Fernandez & Cadelli, 2014)

Scrum es adaptable, ágil, autoorganizado y con pocos tiempos muertos. Se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles. Toma el cambio como una forma de entregar al final del desarrollo algo más cercano a

la verdadera necesidad del cliente. Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común. (Fernandez & Cadelli, 2014). Se basa en los principios ágiles:

- Privilegiar el valor de la gente sobre el valor de los procesos.
- Entregar software funcional lo más pronto posible.
- Predisposición y respuesta al cambio.
- Fortalecer la comunicación y la colaboración.
- Comunicación verbal directa entre los implicados en el proyecto.
- Simplicidad;
- Eliminación de artefactos innecesarios en la gestión del proyecto

2.3 Desarrollo de aplicación móvil

Proceso por el cual se construyen programas móviles útiles para smartphones y tablets. Existe una relevancia de acuerdo la importancia de los smartphones en la sociedad, se esperan 14 millones de aplicativos para el año 2020. (Alférez, 2018)

Android e IOS son las plataformas que dominan la industria con un porcentaje de 98.71% de la cuota de mercado (MarketShare Statistic for Internet Technologies, 2018), cada uno posee su propio ecosistema y ofrece contenido, desarrollo y publicación de aplicativos y servicios.(Alférez, 2018)

Las nuevas tendencias han emergido para proyectos que no tienen recursos para desarrollar en ambas plataformas individualmente, dado que desarrollar en aplicaciones nativas conllevar a mayores costos y mayor tiempo.(Alférez, 2018)

De acuerdo con(Alférez, 2018), los sistemas multiplataformas tienen las siguientes características:

- Solo usan un ambiente y lenguaje de programación para ambas plataformas.
- Usan una base de código compartido que incrementa el desarrollo y el mantenimiento de la lógica de negocio

De acuerdo con(Alférez, 2018) algunas desventajas pueden ser:

- Más difícil conseguir la experiencia fluida en ambas plataformas, el rendimiento puede disminuir.
- Para acceder a la plataforma subyacente de un dispositivo se torna más complejo y complicado el desarrollo.
- Existe inmadurez y falta de estándares entre los practicantes de la organización de los equipos de desarrollo.

De acuerdo con(Rinaldi, 2016) una forma de poder clasificarlos según la comunidad de desarrolladores es de la siguiente forma (véase Figura 20).

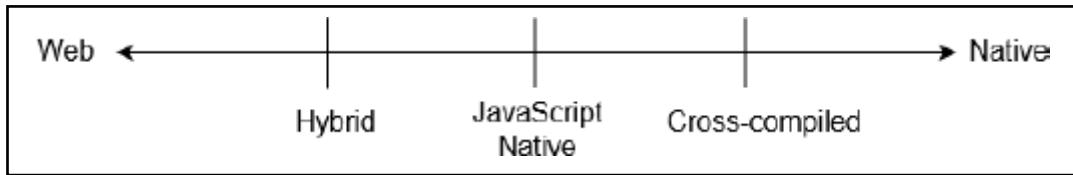


Figura 20. Clasificación de las tecnologías de desarrollo móvil (Rinaldi, 2016)

2.3.1 Web

En este tipo de clasificación se desarrolló un sitio web para el navegador web del dispositivo móvil que pueda usar las funciones táctiles y adaptarse a diferentes tamaños, el propósito es que se pueda reusar el código base y tomar ventajas de las habilidades en el desarrollo web dado que no son usadas las funciones del software o software de las plataformas específicas.(Alférez, 2018)

De acuerdo con(Alférez, 2018) se mencionan dos tipos:

- Diseño responsivo y adaptativo, que optimiza la experiencia del usuario a través de los diferentes dispositivos ajustando el tamaño de ventanas, resoluciones y contexto de uso.
- Aplicativos de web progresivos que es una tecnología la cual toma ventaja de los navegadores y las tecnologías web para comportarse de una forma similar a un aplicativo móvil en un dispositivo.

2.3.2 Híbrido

En este tipo de clasificación (véase Figura 21), se encuentran aquellos aplicativos construidos dentro de un webview que es una interfaz que renderiza el contenido web en un aplicativo, entonces se usan tecnologías como HTML5, CSS y JavaScript y además se provee un componente nativo que provee acceso a los recursos nativos de un dispositivo por ejemplo GPS, cámara entre otros, todo esto directo de JavaScript, gracias a interfaces de funciones escritas en código nativo del sistema operativo.(Alférez, 2018)

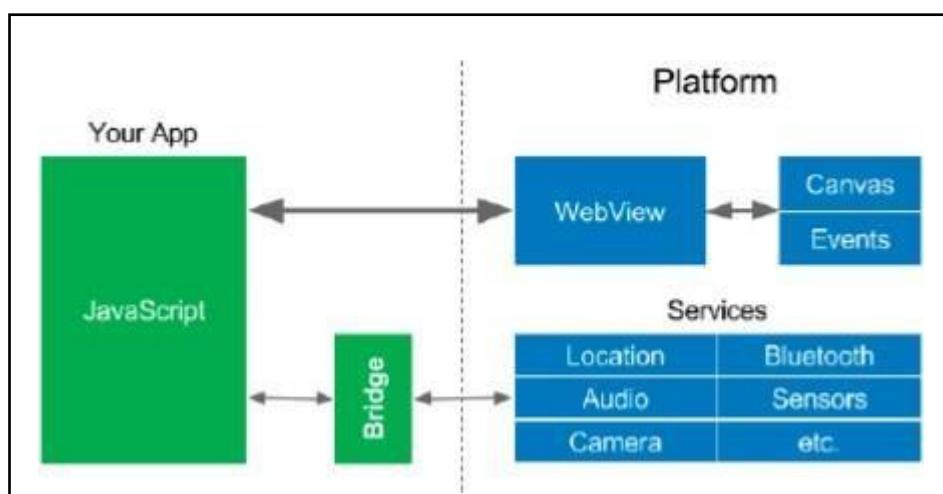


Figura 21. Arquitectura de aplicativos híbridos(Alférez, 2018)

Los inconvenientes vienen a ser un decremento en el rendimiento del dispositivo o que la experiencia de usuario nativa es difícil de conseguir dado que se agregan capas, la ventaja es que se pueden reusar las habilidades web a varias plataformas móviles con un solo código base.(Alférez, 2018)

Apache Cordova es un ejemplo de un componente que puede acceder a funciones del dispositivo, éste es un proyecto de código abierto mantenido por la fundación de Apache Software, entre sus funciones esta que puede establecer comunicación con APIs nativas y el webview mediante plugin de la arquitectura. (Alférez, 2018)

Ionic es un framework de código abierto creado por la compañía del mismo nombre que tiene licencia MIT. Usa Apache Cordova y combina Angular con su propia librería de interfaz gráfica que provee una colección de componentes que tienen la forma, sensación y funcionalidades nativas de cada plataforma móvil. También puede crear aplicativos de web progresivos. Ejemplos de empresas que usaron este framework son McDonald o compañía de ropa Diesel.(Alférez, 2018)

2.3.3 Javascriptnativo

En este tipo de clasificación (véase Figura 22) los aplicativos comparten el código base para varios dispositivos, pero no en vez de construir una página web se renderiza a los widgets nativos de las plataformas elegidas. (Alférez, 2018)

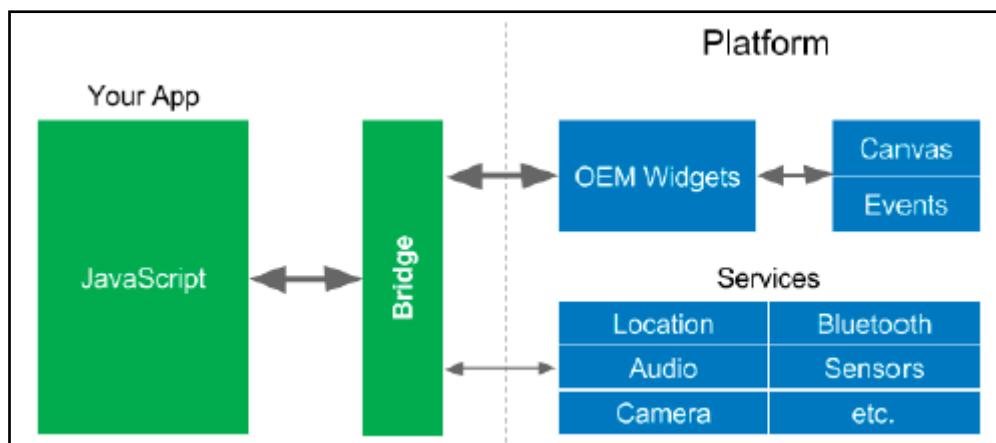


Figura 22. Arquitectura de aplicativos móviles basados en javascript(Leler, 2017)

De acuerdo con(Alférez, 2018), algunas principales soluciones son:

ReactNative es un framework Javascript introducido por Facebook en el año 2015 y está basado en la librería React. Se encarga de usar componentes nativos (bloques de construcción), la arquitectura y flujo de trabajo de las librerías de React. Ejemplos de empresas que usaron este framework son Facebook, Instagram o Airbnb.(Alférez, 2018)

Native Script es un framework desarrollado por Telerik basado en la librería de AngularJS, se diferencia con ReactNative que la primera trata de administrar UI transparentes y repetitivas entre las distintas plataformas, en cambio ReactNative soporta diferencia en el renderizado de las UI. Ejemplos de empresas que usaron este framework son SAP o Verizon.(Alférez, 2018)

2.3.4 Compilación cruzada

Los aplicativos son escritos en un lenguaje de programación base y luego compilados en código nativo para cada plataforma móvil soportada.(Alférez, 2018)

De acuerdo con(Alférez, 2018) entre las principales soluciones están Xamarin y Flutter:

- Xamarin fue desarrollado por la empresa del mismo nombre luego comprada por Microsoft en 2016, ofrece como lenguaje C#, librería de clases para trabajar alrededor de 3 plataformas (Android IOS y Windows Phone) compilado en aplicaciones nativas.
- Flutter es un framework móvil de interfaces gráficas de Google que permite desarrollar interfaces multiplataforma, ofrece como lenguaje Dart. Uno de los aplicativos que usaron este framework es la aplicación de música Hamilton.

2.3.5 Nativo

El desarrollo nativo involucra usar el SDK (Software Development Kit) de Android o IOS para codificar con sus propias herramientas, programación de lenguaje y APIS que funcionen en su plataforma. Se compilan en código maquina y puede interactuar directamente y tomar ventaja de características del sistema operativo (usar software o hardware específico de un dispositivo)(Alférez, 2018)

Android por su parte es un sistema operativo desarrollado y mantenido por Google basado en una versión modificada del Kernel de Linux. Los aplicativos pueden ser escritos con Java o Kotlin usando Android SDK.(Alférez, 2018)

IOS por otro lado tiene su sistema operativo desarrollado y mantenido por Apple Inc. y es exclusivo para su hardware a diferencia de Android, está basado en UNIX y las aplicaciones pueden ser escritas con Swift u Objective-C.(Alférez, 2018)

A continuación, se muestra la arquitectura de aplicaciones móviles nativas que muestra los componentes y la forma que interactúan el aplicativo y la plataforma. (véase Figura 23)

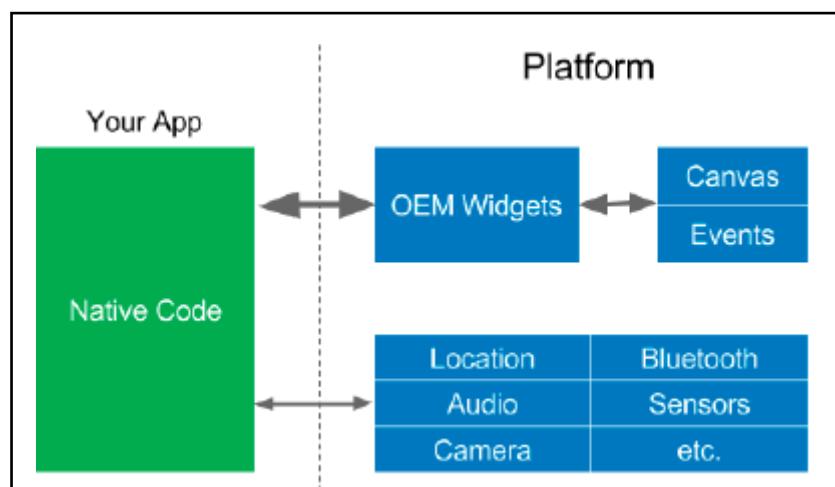


Figura 23. Arquitectura de aplicaciones móviles nativas (Leler, 2017)

2.4 Arquitectura

2.4.1 Arquitectura orientada a servicios (SOA)

SOA es una estrategia de TI que organiza las funciones contenidas en las aplicaciones empresariales en servicios interoperables basados en estándares que pueden combinarse y reutilizarse rápidamente para satisfacer las necesidades del negocio.(Pertcu & Iordan, 2009)

De acuerdo con(Cañaveral, 2012), SOA permite la reutilización de los activos de TI existentes en una empresa, interoperabilidad entre aplicaciones heterogéneas y tecnologías diversas. SOA proporciona un nivel de flexibilidad que antes no era posible en el sentido que:

- Los servicios son componentes de software con interfaces bien definidas que son independientes de la implementación, separando el qué (interfaz de servicio) con el cómo (implementación).
- Los servicios son autónomos, realizan tareas predeterminadas y ligeramente acopladas, por la independencia.
- Los servicios pueden ser dinámicamente resueltos, en cuanto a su estructura.
- Se pueden crear servicios compuestos a partir de otros servicios.

SOA utiliza el paradigma find-bind-execute (encontrar-unir-ejecutar), en este paradigma, los proveedores registran sus servicios públicamente. Este registro es utilizado por los consumidores para encontrar servicios que cumplan ciertos criterios. Es indispensable realizar aplicaciones que transformen cualquier funcionalidad en servicio para una correcta integración a la arquitectura SOA.(Cañaveral, 2012)

A continuación, se muestra los elementos que componen la arquitectura SOA. (véase Figura 24)

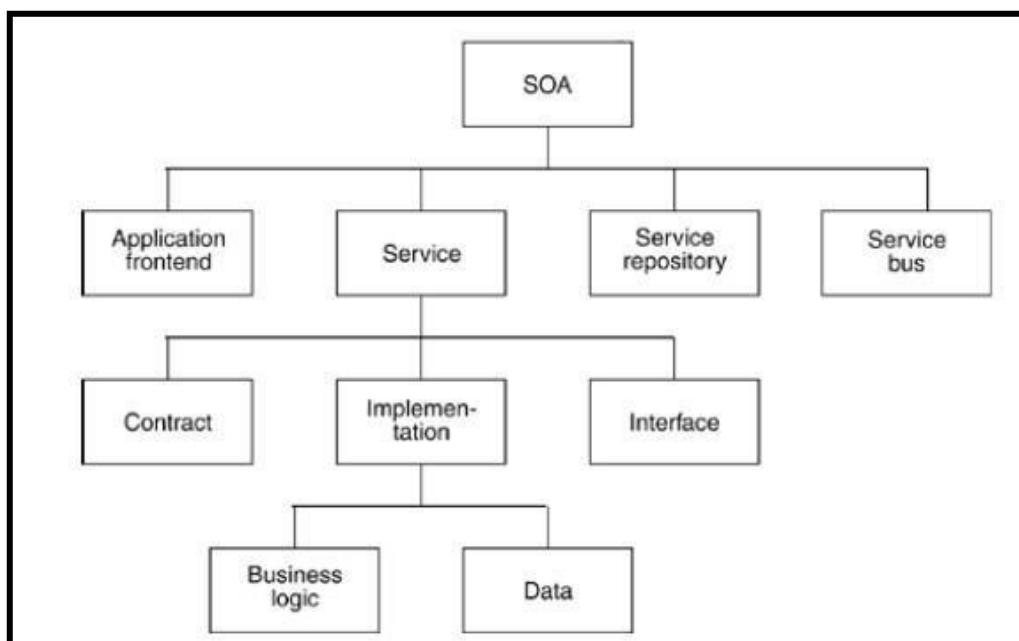


Figura 24. Elementos de una Arquitectura orientada a servicios(Krafzig, Banke, & Slama, 2004)

2.4.2 Arquitectura basada en Microservicios

Según Martin Fowler y James Lewis, dos principales “descubridores” de este tipo de arquitectura. Se define a microservicios como “un estilo arquitectural en el que múltiples servicios, cada uno corre de manera individual y desplegados de la forma más automatizada posible, se comuniquen mediante mecanismos ligeros generalmente un recurso API basado en HTTP”(Fowler & Lewis, s.f.)

Los microservicios tienen como fundamento principal dividir el software de la aplicación en pequeños servicios que sean lo más independientes uno de otro, estos deben ser fácilmente reemplazables o actualizables.(Perez-Herrera, 2015)

Entre las características más resaltantes se encuentra visión global de aplicación web desarrollada se compone de orquestadores que son los que manejan los microservicios, es el encargado de dirigir y manejarlas peticiones hacia otros microservicios, la comunicación predominantemente es de tipo REST debido a su facilidad de uso y rapidez a diferencia de SOAP.(Perez-Herrera, 2015)

Se mencionan ciertas características comunes entre los aplicativos microservicios:(Perez-Herrera, 2015)

- Se componen de servicios
- Organización en base al negocio
- Existen productos mas no proyectos
- Administración descentralizada
- Base de datos descentralizada
- Infraestructura automatizada
- Diseñado para evolucionar

Ventajas

De acuerdo con(Perez-Herrera, 2015), entre las ventajas se encuentran:

- Esta arquitectura refuerza la estructura modular lo cual es ventajoso para sistemas y/o equipos grandes
- Los servicios son más fáciles de implementar, menos propensos a causar fallos en el sistema cuando fallan.
- Proveen de diversidad tecnológica, con microservicios se pueden mezclar varias tecnologías: diferentes lenguajes de programación, diferentes almacenamientos de datos

Desventajas

De acuerdo con(Perez-Herrera, 2015), entre las desventajas se encuentran:

- Los sistemas distribuidos son más difíciles de programar, ya que las llamadas remotas son lentas y pueden ser riesgo para fracaso

- La consistencia es eventual, ya que todo no está en un solo lugar sino disperso en varios lugares, por lo tanto, se debe realizar una gestión de la consistencia eventual para cada uno de ellos.
- Se debe contar con gente con experiencia para manejar la gran cantidad de servicios.

Se puede apreciar la diferencia entre una arquitectura monolítica cual contiene todo a esto nos referimos con interfaz de usuario, lógica de negocio, capa de acceso a datos; y la otra arquitectura basada en microservicios que se subdivide en varios aplicativos cada uno con funciones específicas que se comunican entre sí con la vista y/o con la base de datos. (véase Figura 25)

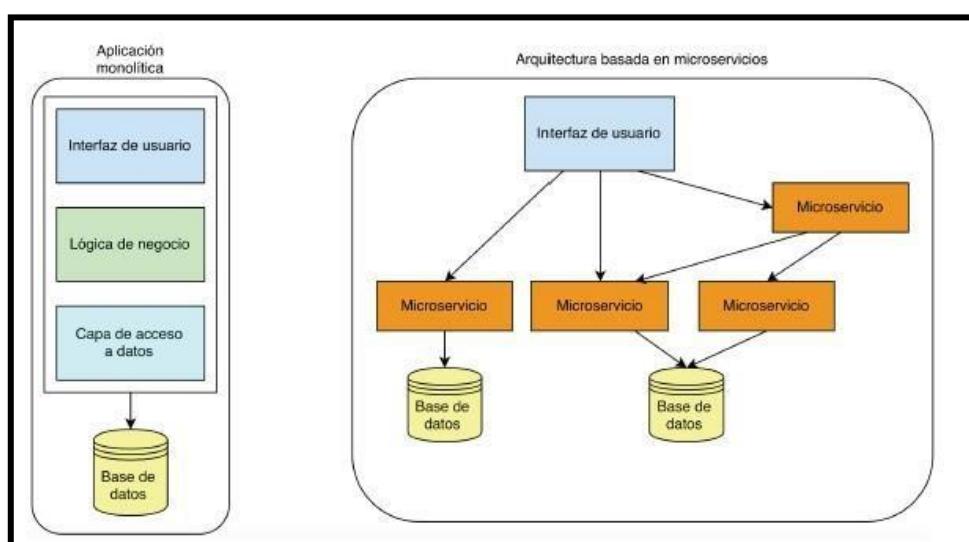


Figura 25. Arquitecturas monolíticas vs arquitectura de microservicios (Perez-Herrera, 2015)

3 ESTADO DE ARTE

3.1 Taxonomía

Según la taxonomía publicada por ACM (véaseFigura26), la presente tesis pertenece al área temática de investigación *Service-oriented architectures* (en español Arquitectura orientada a servicios) que pertenece a la clasificación *Enterprise computing* (en español Computación empresarial) que a su vez pertenece a la clasificación *Applied computing* (en español Computación aplicada).

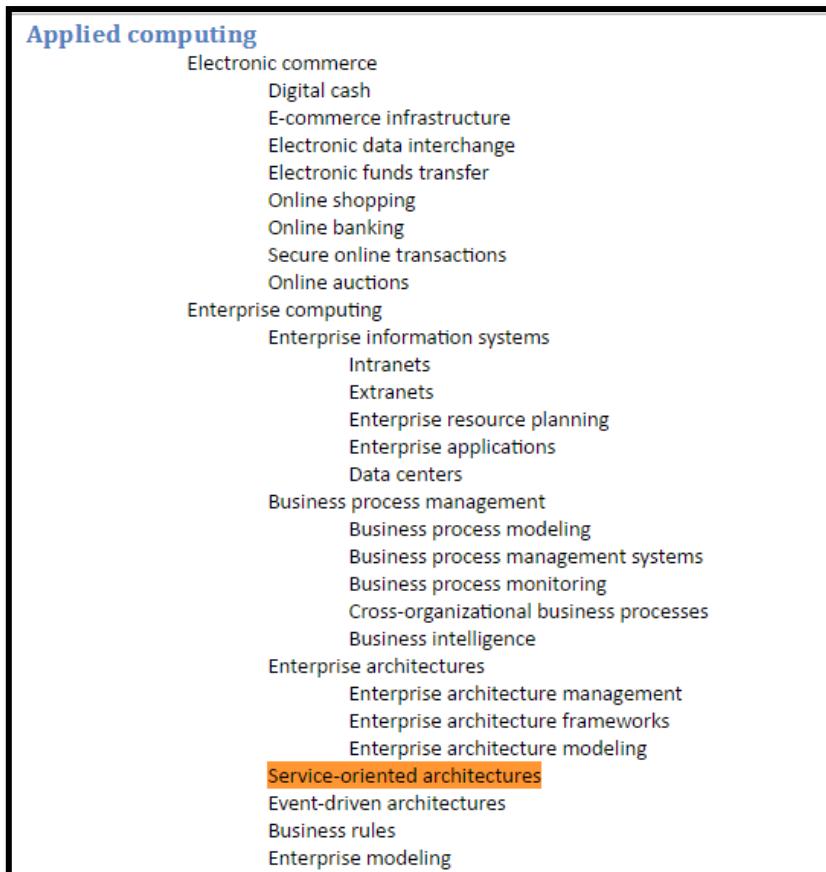


Figura26. Clasificación ACM 2012 (Association for Computing Machinery, 2019)

Según la taxonomía de IEEE (véase Figura 27), la presente tesis pertenece al área temática de investigación *Distributed information system* (en español Sistema de información distribuido) que pertenece a la clasificación *Distributed computing* (en español Computación Distribuida) que a su vez pertenece a la clasificación *Computers and information processing* (en español Computadoras y procesamiento de información) (IEEE, 2019)

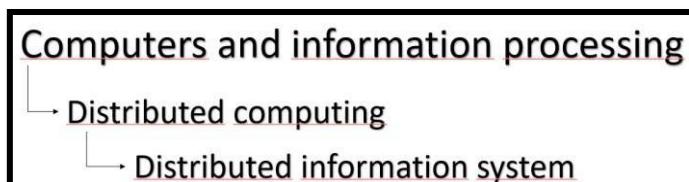


Figura 27. Taxonomía IEEE 2019 (IEEE, 2019)

De acuerdo a la línea de investigación de la facultad de ingeniería de sistemas e informáticaUNMSM (véaseFigura 28), la presente tesis se encontraría en el área temática *Ingeniería de Software* del Área de computación(Facultad de Ingeniería de Sistemas e Informática UNMSM, 2019). Según(Delgadillo & Fermín, 2018),esta área contiene los tópicos del desarrollo del software mencionados como: diseño, implementación, arquitectura, orientado a servicios, enfoque para dispositivo móvil, multiprocesamiento, distribuido, tiempo real los cuales estan relacionados con la presente tesis.

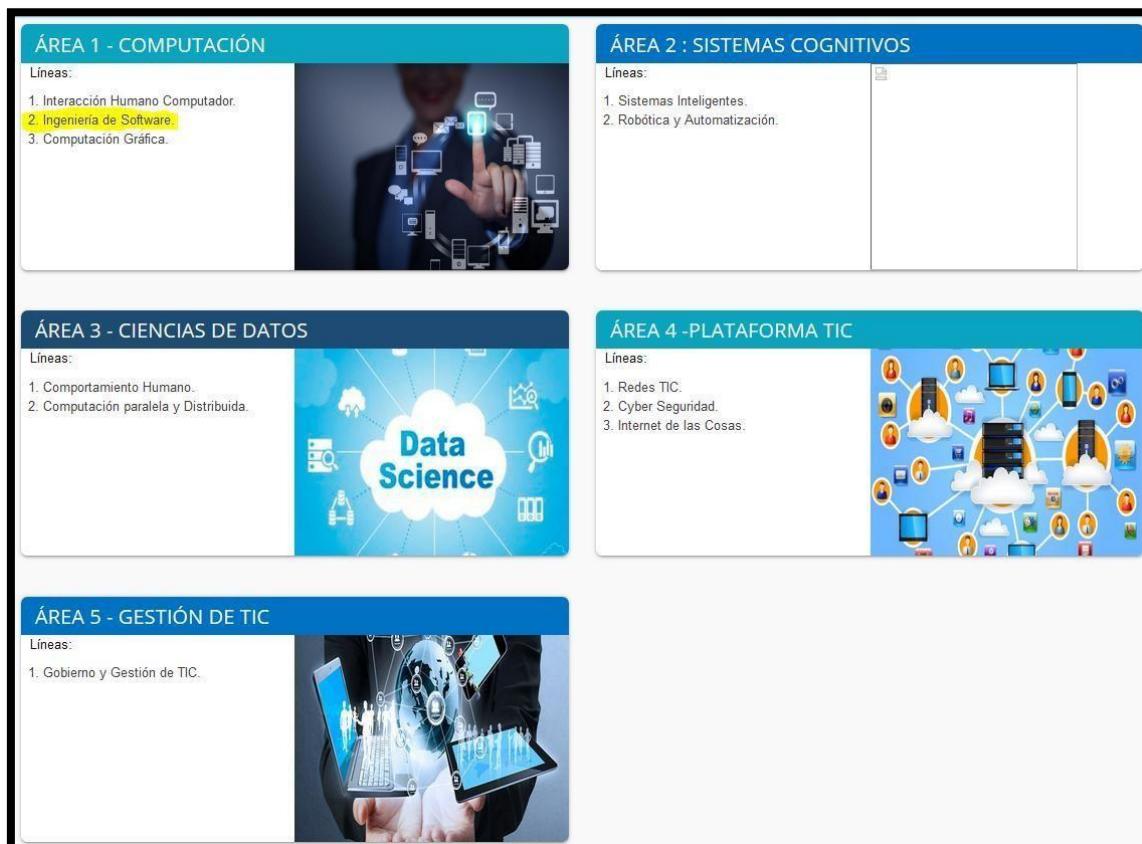


Figura 28.Líneas de investigación de la facultad de ingeniería de sistemas e informática(Facultad de Ingeniería de Sistemas e Informática UNMSM, 2019)

3.2 Método de revisión de la literatura

Para la selección del material bibliográfico y documental se hizo uso de Google Académico para conceptos como SOA y Microservicios. En cuanto a la selección de artículos,estas fueron escogidas a través de repositorios, bibliotecas digitales y bases de datos como:Scielo, ACM, IEEE, ScienceDirect, Redalyc, Scopus. Para la recopilación de tesis, estas fueron halladas en archivos, repositorios y bibliotecas digitales de diferentes universidades,tales como:

- ❖ Universidad Nacional Mayor de San Marcos(UNMSM).
- ❖ Universidad Nacional Federico Villareal (UNFV).
- ❖ Universidad Politécnica de Madrid (UPM).
- ❖ Universidad Politécnica de Cataluña (UPC).

- ❖ Universidad Israel (UISRAEL).
- ❖ Universidad Central del Ecuador (UCE).
- ❖ Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López (ESPAM).

Para la búsqueda se utilizaron descriptores como: sistema de control, arquitecturas de software, SOA, arquitectura orientada a servicios,microservicios, metodología ágil, Scrum, computación distribuida, sistemas móviles híbridos, ReactNative, Ionic. Adicionalmente, se tuvo en consideración los siguientes dos criterios para el filtro: los criterios de inclusión y exclusión.

Criterios de inclusión:

- Como criterio de inclusión se menciona que los artículos y tesis tengan una antigüedad menor a 8 años (2012 - 2019).
- Artículos que pertenezcan a revistas indexadas.

Criterios de exclusión:

- Como criterio de exclusión se menciona que los artículos y tesis tengan una antigüedad mayor a 8 años (menor al año 2012).
- Artículos que no pertenezcan a revistas indexadas.

3.2.1 Artículos

3.2.1.1 Una aproximación holística a las metodologías agiles desde la programación extrema

(Pardo, Triana, & Forero, 2014)

3.2.1.1.1 Estado de arte

Los autores centran su atención en las metodologías tradicionales de desarrollo argumentando que estas carecen de “agilidad”, caracterizadas por un mayor énfasis en la documentación en lugar de entregar valor al cliente progresivamente a través de módulos de software funcionando, de poseer una notoria deficiencia en cuanto a flexibilidad (ya que se sigue un plan general de proyecto) y una baja capacidad de adaptación al cambio lo cual incurre en mayores costos. Según los autores, todo ello puede generar un bajo nivel de satisfacción en el cliente, ya que ocasionalmente los resultados obtenidos no se corresponden con las funcionalidades esperadas en el producto final.

3.2.1.1.2 Motivación del autor

La motivación de los autores viene a ser dada en encontrar una metodología de desarrollo que permita reducir las desventajas ya mencionadas de las metodologías tradicionales. Por lo que optan por las metodologías de desarrollo agiles, en particular la XP (Extreme Programming), a través de un análisis bottom-up resaltando sus virtudes resultado de la estrecha relación entre el cliente y el desarrollador capturar y validar los requerimientos del cliente.

3.2.1.1.3 Descripción del aporte del autor

El aporte radica en cómo se debe llevar a cabo la metodología de desarrollo ágil “XP”, mostrándolo objetivos, el ciclo de vida, los valores que prevalecen durante el desarrollo

y los principios que la rigen. Adicionalmente, a modo de resumen, se menciona un cuadro comparativo entre las metodologías tradicionales versus las metodologías agiles, dejando en claro la superioridad de las metodologías agiles en lo que se refiere a nivel de satisfacción del cliente, la flexibilidad frente a un plan general de proyecto y la rápida adaptación al cambio.

3.2.1.4 Proceso para resolver el problema

Para contrarrestar las desventajas de las metodologías tradicionales, los autores exponen los objetivos, etapas, principios y valores de la metodología ágil de desarrollo "XP".

Objetivos en común:

- Establecer las mejores prácticas de Ingeniería de software en el desarrollo de proyectos.
- Optimizar la ejecución de los proyectos.
- Garantizar la calidad del software desarrollado.
- Potenciar al máximo el trabajo en equipo.
- Satisfacer los requerimientos del cliente.

Etapas: Planificación, diseño, desarrollo y pruebas

Valores: Comunicación, simplicidad, retroalimentación, coraje, valentía y respeto.

Principios:

- La mayor prioridad es la satisfacción del cliente.
- Bienvenidos los cambios, aun después del desarrollo.
- Entregas de software frecuentes.
- Trabajo en equipo.
- Construir proyectos alrededor de individuos motivados.
- Obtener la información de manera personalizada.
- El software que se trabaja es la mejor medida del progreso.
- Los procesos agiles promueven desarrollos sostenibles.
- El buen diseño mejora la agilidad.
- La simplicidad.
- Las mejores arquitecturas.
- Continua reflexión en búsqueda de la eficacia.

3.2.1.5 Conclusiones

Resulta evidente las virtudes ofrecidas por la metodología XP, ya que proponen un mayor énfasis en el proceso progresivo e incremental de entregar software funcionando, con ello, el cliente contempla la evolución del producto requerido siendo capaz de solicitar cambios durante el transcurso del desarrollo. Lo que lo lleva a ser más flexible, más tolerante al cambio y además ser más eficiente en cuestión de tiempos de desarrollo. Al tener un contacto más directo con el cliente, la captación y validación de requerimientos del cliente suele ser más refinado, por ende, se logra tener una mayor visión de lo que realmente está buscando el cliente. Todo esto se traduce a un alto grado de satisfacción del cliente.

3.2.1.6 Aporte

El aporte a nuestra tesis viene a ser las ventajas de las metodologías agiles frente a las metodologías tradicionales.

3.2.1.2 JavaScript in mobile applications: React Native vs Ionic vs NativeScript vs Native development

(Brito, Gómez, Santos, & Bernardino, 2018)

3.2.1.2.1 Estado de arte

Los autores mencionan el alto crecimiento en el uso de los teléfonos inteligentes y como estos han mejorado su capacidad de procesamiento (mejor calidad de procesadores, estabilización de los sistemas operativos, conectividad y almacenamiento mejorado). Como consecuencia, ello genera un gran interés en el desarrollo de aplicaciones móviles de una manera rápida sin mayores obstáculos. La construcción de las aplicaciones móviles se puede clasificar en 3 grandes grupos: nativas, web e hibridas. A diferencia de las aplicaciones móviles hibridas y web, las aplicaciones nativas requieren de un desarrollo único por plataforma, lo que incurre a mayores tiempos, costos de desarrollo y mantenimiento.

3.2.1.2.2 Motivación del autor

Los autores se encuentran motivados en encontrar una alternativa eficiente al desarrollo de aplicaciones nativas, es decir, en implementar una solución única que pueda ser distribuida en diversas plataformas (sin la necesidad de codificar en lenguajes diferentes), contar con una documentación de calidad, manejar una buena curva de aprendizaje, contar con herramientas que agilicen el desarrollo, tener buenos de respuesta y finalmente, la capacidad de mantenimiento y de actualizaciones.

3.2.1.2.3 Descripción del aporte del autor

El aporte reside en realizar un análisis comparativo entre las aplicaciones nativas e hibridas basadas en 7 criterios (evaluadas según escala Likert) señalando ventajas e inconvenientes que puedan beneficiar o interrumpir el desarrollo de una aplicación móvil.

3.2.1.2.4 Proceso para resolver el problema

Para la obtención del análisis comparativo, se tuvieron que evaluar aplicaciones nativas vs hibridas (ReactNative, Ionic y NativeScript) según los siguientes criterios:

- Aprendizaje y calidad de la documentación
 - ❖ En las aplicaciones nativas la curva de aprendizaje se vuelve complejo al tener que aprender al menos dos lenguajes de programación. Por otro lado, ambos sistemas operativos cuentan con documentación de calidad, grandes comunidades de desarrollo y versiones estables. En cuanto a las aplicaciones hibridas solo requieren de un único lenguaje de programación (JavaScript), lo cual lo hace más fácil de aprender. En cuanto a la documentación, ReactNative se destaca considerablemente con una documentación muy bien estructurada.
- Costo de desarrollo
 - ❖ Las aplicaciones nativas evidentemente tienen un costo más alto que las hibridas debido a que son implementadas en lenguajes diferentes.

Mientras que las aplicaciones híbridas poseen un costo mínimo, ya que solo se requiere programar en un único lenguaje. Sin embargo, cuando se requieren de prestaciones externas dentro del dispositivo móvil, los costos suelen aumentar debido a que no hay mucha documentación con respecto a estas integraciones.

- Emuladores y depuración
 - ❖ Las aplicaciones nativas cuentan con entornos de desarrollo integrados reconocidos en el mercado, tales como Android Studio o Xcode. Con respecto a las aplicaciones híbridas, el desarrollo resulta muy similar a la programación web al usar hot reloading, mecanismo que permite visualizar un cambio realizado poco después que se haya guardado un cambio en el código trabajado.
- Tiempo de respuesta y velocidad
 - ❖ Las aplicaciones nativas como es de esperar poseen un tiempo de respuesta casi instantáneo. En las aplicaciones híbridas, en particular React Native y NativeScript, cuentan con tiempos de respuesta casi similares a las totalmente nativas. Mientras que Ionic, cuenta con latencias perceptibles por cualquier usuario.
- Reconocimiento profesional
 - ❖ Las aplicaciones nativas tienen abarcado un gran sector del mercado, tales como YouTube, Google Maps son un claro ejemplo de ello. Por otra parte, aplicaciones híbridas basadas en React Native, han tenido una exponencial acogida en el mercado, ejemplos claros de ello son: Facebook, Instagram, Skype.
- Reutilización de código y trabajo en equipo
 - ❖ Las soluciones nativas no tienen ventajas o desventajas en cuanto a términos de reutilización. Para el caso de los multiplataforma (híbridas), se cuentan con componentes disponibles para ser reutilizados en otros proyectos.
 - ❖ El trabajo en equipo para las soluciones híbridas resulta ser multidisciplinar, ya que una tarea independientemente de su complejidad puede ser abordada por diferentes programadores. A diferencia de las soluciones nativas, en las cuales la ayuda mutua suele ser más complicada debido a que cada plataforma maneja un lenguaje diferente.
- Mantenimiento y actualizaciones
 - ❖ Al realizar una modificación a las soluciones nativas estas repercutirán directamente a dos diferentes proyectos (Android y iOS), lo cual se traduce en doble trabajo. A diferencia de las soluciones híbridas, las cuales solo dependen de un único proyecto, sin embargo, cuando se trate de un error específico en un sistema operativo la corrección suele ser más complicado de resolver en comparación con las soluciones nativas.

3.2.1.2.5 Conclusiones

Resulta claro que si lo que se requiere es desarrollar un aplicativo móvil básico y puntual sin recurrir en demasia a los recursos externos propiciados por el dispositivo

móvil, entonces debemos inclinarnos por las soluciones híbridas, ya que estas nos ofrecen un único canal para que nuestra aplicación se construya satisfactoriamente. Sin embargo, si es todo lo opuesto a lo anterior, es decir, obtener todo el rendimiento del procesador o querer integrar con servicios externos como: mapas, GPS, VoIP, Chat o Notificaciones Push resultaría más factible optar por las soluciones nativas.

3.2.1.2.6 Aporte

El aporte a nuestra tesis viene a ser las ventajas propiciadas por las aplicaciones híbridas con respecto a las aplicaciones nativas. Al ser nuestra solución un aplicativo móvil de consulta y/o registro de datos con al menos un servicio externo: envío y recepción de notificaciones, vemos conveniente que nuestra solución se trate de una aplicación móvil híbrida.

3.2.1.3 Disambiguation and Comparison of SOA, Microservices and Self-Contained Systems

(Cerny, Donahoo, & Pechanec, 2017)

3.2.1.3.1 Estado de arte

Los autores mencionan la evolución que se ha ido dando en las arquitecturas de software resultado de las exigencias propuestas por la industria. Actualmente, dos de las arquitecturas más renombradas son las arquitecturas: SOA y microservicios. Se detallan cómo están compuestos, qué componentes interactúan y se resaltan palabras clave como “Integración” para SOA y “Unidad independiente” para los microservicios. Luego se presenta un cuadro comparativo en el cual se expresan las principales inquietudes que debe afrontarse al implantar una arquitectura (SOA y Microservicios) que sirva como base para el desarrollo de un proyecto. Entre las ventajas más importantes, comprobaron que los Microservicios ofrecen un alto nivel de desacoplamiento, factibilidad en la entrega y despliegue continuo, eficiente escalabilidad; SOA por su parte, ofrece una mejor integración de servicios heterogéneos ya que cuenta con sofisticadas herramientas en su capa middleware.

3.2.1.3.2 Motivación del autor

La motivación viene a ser dada en expresar una clara imagen entre las arquitecturas SOA y microservicios, señalando las principales diferencias entre cada uno de estas. Asimismo, los autores advierten los desafíos que ocurrirían al implantar cualquiera de estas. Establecen además que arquitectura escoger según el alcance del proyecto. La intención es mantener una perspectiva basada en diferentes criterios que ayuden a despejar cualquier posible confusión entre una arquitectura y otra.

3.2.1.3.3 Descripción del aporte del autor

El aporte reside en realizar un análisis comparativo entre las arquitecturas SOA y microservicios basado en un conjunto de inquietudes que se formulan al implantar una arquitectura.

3.2.1.3.4 Proceso para resolver el problema

Para la obtención del análisis comparativo, los autores hicieron presentación de las arquitecturas de software con más renombre en las últimas décadas, SOA, microservicios y sistemas auto contenidos (Self-Contained System); señalando la naturaleza de estos, cómo están estructurados y cuáles son sus principales desafíos.

- **Arquitectura orientada a servicios (SOA):**

Esta arquitectura se caracteriza por tener un estilo orquestado (servicios tontos y tuberías inteligentes), por ejemplo, para resolver la solicitud de un cliente, este debe pasar a través de un middleware (ESB) que comprende el paso a paso entre la interacción de varios servicios predefinidos. A pesar de que esta tecnología provea de un amplio abanico de herramientas de integración, su debilidad radica en el impacto que surge al realizar algún cambio en cualquierservicio, el cual se podría propagar a las próximas capas, a otros ESB que dependan del servicio en cuestión o a la capa frontend.

- **Microservicios:**

Esta arquitectura tiene un estilo coreográfico (servicios inteligentes y tuberías simples) es decir cada servicio debe serresponsable de realizar sus propias tareas a cada momento, ademásdeben saber cómo interactuar con otros servicios, sin depender del direccionamiento de otros componentes. Las ventajas que ofrece esta arquitectura es la de eliminar el acoplamiento originado por el middleware (SOA), enintegrarse correctamente con la filosofía de DevOps (Entrega continua + Despliegue continuo)dado que los cambios originados en algún microservicio no repercuten en otros dominios, solo en ellos mismos, el cual permite a su vez una efectiva escalabilidad. Como contraparte se incrementa la complejidad en el modelo de datos, lo cual podría generar replicación de datos.

- **Self-ContainedSystem:**

La propuesta de esta arquitectura es la misma a lo señalado en los microservicios solo queagregando los módulos relacionados con la interfaz de usuario. El beneficio que se obtiene al implantar este tipo de arquitectura es la de responsabilizar el impactode un cambio originado en un microservicio a un solo equipo de trabajo cubriendo todas las capas del mini sistema (Lado cliente + Lado servidor). A diferencia de los microservicios donde la codificación de la interfaz de usuario es responsabilidad de terceros.

3.2.1.3.5 Conclusiones

Claramente existe una gran ventaja de la arquitectura de microservicios frente a SOA por las características mencionadas anteriormente. Según mencionan los autores, esta arquitectura calza mejor para aplicaciones relativamente pequeñas, bien particionadas y definidas. Sin embargo, SOAtiene un punto fuerte: sus herramientas de integración, siendo mejor en el desacoplamiento del contrato, por ende, estámás orientado agrandes, complejas y amplias organizaciones que incluyen sistemas legados, componentes compartidos a lo largo de la empresa.

3.2.1.3.6 Aporte

El aporte a nuestra tesis viene a ser las ventajas que ofrece la arquitectura de microservicios frente a la arquitectura SOA. Al ser nuestra solución un proyecto de un mediano alcance, sin sistemas legados al que incurrir, sin dependencias transversales; vemos conveniente optar por la arquitectura de microservicios, ya que sin lugar a duda

ofrece condiciones que van de la mano con las buenas prácticas de ingeniería de software.

3.2.1.4 Propuesta de arquitectura de microservicios, metodología Scrum para una aplicación móvil de control académico: Caso Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos.

(De la Cruz, Espinoza, & Cuba, 2019)

3.2.1.4.1 Estado del arte

Los siguientes hechos manifiestan una alarmante situación de la calidad académica en las universidades peruanas:

- La gran mayoría de las universidades peruanas están muy lejos del puesto 30 según (Quacquarelli Symonds, 2019).
- La desigualdad de acceso y oportunidades en las diferentes provincias producto del centralismo.
- El aumento de universidad privadas que no cumplen con los estándares establecidos para el proceso de licenciamiento de la Superintendencia Nacional de Educación Superior (SUNEDU).

La Escuela Profesional de Obstetricia (EPO) de la Universidad Nacional Mayor de San Marcos (UNMSM) es consciente de cumplir con los estándares propuestos para conseguir un mejor posicionamiento académico. La Escuela Profesional de Obstetricia (EPO) presenta ciertas falencias las cuales le impide lograr su propósito:

- No hay un seguimiento del silabo que permita comprobar que un docente haya cumplido a carta cabal con su temario durante todo un semestre.
- El registro de calificaciones o toma de asistencias se realizan a través de papel físico, archivos Excel u otros.
- La emisión de comunicados por parte de los docentes es realizada a través de aplicaciones externas como Facebook, WhatsApp, correo.

3.2.1.4.2 Motivación del autor

La motivación radica en proponer un diseño de sistema distribuido que permita corregir las falencias presentadas en la Escuela Profesional de Obstetricia – Facultad de Medicina de la Universidad Nacional Mayor de San Marcos. Asimismo, a través de esta propuesta se busca propiciar un canal único e integral que alberge las diferentes actividades relevantes realizadas en el día a día por estudiantes, docentes y la dirección de escuela.

3.2.1.4.3 Descripción del aporte del autor

El aporte reside en fórmular un diseño de sistema distribuido móvil que abarque el control académico y la gestión en la Escuela Profesional de Obstetricia – Facultad de Medicina de la Universidad Nacional Mayor de San Marcos.

3.2.1.4.4 Proceso para resolver el problema

El diseño del sistema propuesto está comprendido por dos partes: la creación de prototipos para el aplicativo móvil y el diseño de la arquitectura basado. Para la captación de requerimientos del usuario se apoyó de la metodología Scrum. Se tiene

proyectado acabar con el desarrollo dentro de 6 Sprints, siendo la duración de cada sprint: 1 mes. El primer sprint es destinado a la concepción de la arquitectura de Microservicios que soporte el aplicativo móvil, las tecnologías a escoger para el desarrollo móvil, mientras que los restantes sprints serán proyectadas para el desarrollo y validación de las tareas inferidas por las historias de usuario obtenidas juntamente con el Product Owner. En cuanto al diseño de la arquitectura, esta se basa en Microservicios. Se optó por esta arquitectura al comprobar a través de un análisis comparativo entre SOA y Microservicios, que estos últimos ofrecen mayores ventajas al considerar al proyecto como un proyecto incipiente, sin dependencias transversales y de escala mediana. La arquitectura propuesta presenta dos tipos de componentes: funcionales y no funcionales. Entre los componentes funcionales se tiene:

- **Autenticación:** Microservicio encargado de la seguridad de los microservicios en general. Basado en OAuth2, el cual cede un token de acceso cada vez que inicie sesión y este token permitirá acceder a los demás recursos propiciados por los demás servicios.
- **Asistencia:** Permite consultar, registrar, actualizar y finalizar las asistencias de las clases.
- **Encuesta:** Encargada de calificar el grado de satisfacción de un estudiante originado por la clase impartida de un docente.
- **Coparticipación:** Responsable de la interacción entre docentes y estudiantes, es decir, emisión de comunicados, sistema de preguntas y respuestas relacionadas a un tema en común.
- **Notas:** Permite consultar y registrar las calificaciones de los estudiantes a un nivel detallado según la fórmula especificada por el propio docente.
- **Periodo académico:** Contiene la malla curricular correspondiente a los diferentes cursos de un semestre académico. Posee la relación de alumnos inscritos, la relación de horarios, los grupos asignados, etc.
- **Notificación:** Responsable de notificar cualquier evento relevante realizado por algún actor del Sistema.
- **Reportes:** Responsable de presentar estadísticas referentes a asistencias, calificaciones y satisfacción por clase impartida. Con la finalidad que sean útiles para unidad estratégica en la toma de decisiones.

Entre los componentes no funcionales, se cuentan con:

- **Eureka-Server:** Permite el auto-registro y descubrimiento dinámico desde que un servicio es iniciado por primera vez.
- **Config-Server:** Encargada de alojar las propiedades de configuración de todos los microservicios. Con ello no existe la necesidad de reiniciar un microservicio al querer cambiar una propiedad de configuración.
- **Gateway:** Es la única puerta por la que un cliente puede acceder a un recurso propiciado por cualquier microservicio. Tiene la funcionalidad de rutear la solicitud de un cliente a un determinado Microservicio (Labor de Proxy).
- **Composición:** Su función principal es la de armar estructuras compuestas de dos o más respuestas de microservicios diferentes.

3.2.1.4.5 Conclusiones

Con la propuesta de este diseño se tiene un mejor punto de partida para la posterior implementación del sistema móvil distribuido. Al apoyarse en Scrum para la captación de requerimientos a través de historias de usuario y poder formalizarlas a través de criterios de aceptación asegura lo que el cliente realmente requiere. Por otro lado, la arquitectura de Microservicios formulada establece las bases de un desarrollo continuo y evolutivo que lo centraliza mayormente a culminar los requerimientos funcionales, por lo que generalmente son los que aportan mayor valor al cliente.

3.2.1.4.6 Aporte

Al ser parte de nuestra tesis, esto aporta directamente en la resolución del problema como un preludio que permita la satisfactoria implementación de la solución.

3.2.1.5 Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI

(Brito J. A., 2016)

3.2.1.5.1 Estado del arte

El autor indica que no había mucha literatura científica entre comparativo de CMMI y metodologías ágiles y que era importante hacerlas sobre CMMI dado que se establece un marco común para poder evaluarlas. Además, menciona que para las empresas es muy costoso y complejo.

3.2.1.5.2 Motivación del autor

La motivación del autor viene a ser dada por crear un instrumento fácil que permita al entorno empresarial precisar las diferencias de las metodologías ágiles con las áreas de CMMI; dado que se menciona que realizar los procesos de evaluación y mejora tradicionales de estos métodos para las empresas son costosas y complejas.

3.2.1.5.3 Descripción del aporte del autor

El aporte del autor viene a ser dado por el cuadro comparativo entre las metodologías XP, Scrum e Iconix que permite ver en qué áreas en las que se fortalecen una u otra, estas áreas de proceso figuran en los niveles del CMMI, además muestra gráficos sobre la comparativa de la evaluación del proceso de desarrollo de software y por último la comparativa del proceso de desarrollo de software y las metodologías mencionadas

3.2.1.5.4 Proceso para resolver el problema

Se plantea realizar instrumentos que permitan clasificar las metodologías ágiles entre los criterios “No soportada”, “Parcialmente soportada en menor medida”, “Parcialmente soportada”, “Parcialmente soportada en gran medida”, “Soportada”.

Se hizo una calificación por cada subpráctica entre las metodologías ágiles marcando OK o X.

Scrum demuestra fortaleza en las áreas REQD, PMC, REQM, VAL mientras que XP tiene un cubrimiento mayor en CMMI. Scrum está más enfocado en gestión mientras que XP en construcción y desarrollo del producto.

3.2.1.5.5 Conclusiones

El instrumento es una herramienta útil que usa los niveles de CMMI y sus procesos para la comparación de prácticas ágiles y así poder tomar una mejor decisión.

El instrumento no solo puede abarcar las metodologías ágiles sino también a otros métodos o procesos.

3.2.1.5.6 Aporte

El aporte a nuestra tesis viene a ser dada porque enfatiza las virtudes de SCRUM usando las prácticas de los distintos niveles de CMMI las cuales contiene una fuerte gestión y desarrollo aceptables en comparación a otras metodologías como XP o Iconix.

3.2.2 Tesis

3.2.2.1 Sistema web de gestión académica en el centro de idiomas de la ESPAM MFL

(Ganchoso & Vera, 2015)

3.2.2.2.1 Estado del arte

El autor menciona que la escuela superior ESPAM MFL (Politécnica Agropecuaria de Manabí Manuel Félix López) es un centro de idiomas que inicialmente atendía a una cantidad ínfima de alumnos, pero con el tiempo se dio un aumento de la demanda de cantidad de personas. Se hace mención que en tal centro de idiomas se realizaba la elaboración de la oferta académica y su difusión, los procesos de preinscripción, admisión, matrícula y gestión de expedientes. Este centro estaba a cargo de una sola persona y los registros se llevaban manualmente, por su parte los estudiantes realizaban sus trámites académicos de manera presencial.

3.2.2.2.2 Motivación del autor

La motivación de los autores viene a ser dada por quécree que pueda apoyar en la agilización del proceso de gestión en la escuela que se encuentran con los problemas descritos por medio de la automatización de tareas el cual mejorará el flujo, la confiabilidad y seguridad de la información, ahorro de tiempo, menos gasto en papel y beneficiará a los alumnos profesores y al director.

3.2.2.2.3 Descripción del aporte del autor

El aporte de los autores viene a ser dada por la implantación de un sistema que permita agilizar la gestión de procesos académicos del Centro de Idiomas de la ESPAM MFL esto debido a que el número de estudiantes generó mucha demanda y había poco personal para abarcarlo.

El autor menciona que la gestión académica tiene como objetivo diseñar e implementar un sistema que permita evaluar y mejorar la propuesta pedagógica de calidad que forme integralmente al estudiante, para esto se hace uso del diseño curricular, además incluye procesos de preinscripción, selección, admisión, planes de estudio, modelos pedagógicos ofertas de programas e información académica. Involucra las acciones del estudiante desde su ingreso, cuenta con el desarrollo y el egreso del sistema.

Así mismo el autor menciona que el sistema permitió un mejor manejo de información, disponibilidad, consistencia y seguridad de esta como también menos tiempo y menos gastos de movilización, también se disminuirá el uso de papel.

3.2.2.2.4 Proceso para resolver el problema

Para resolver el problema el autor divide el avance en los 3 pasos de la metodología Scrum mencionados anteriormente:

a) “Pre-Juego” (Primera fase)

En esta etapa del Scrum se realiza la planeación del sistema en 4 sprints:

- Sprint1: Modelado de datos, interfaz gráfica y gestión de usuarios.
- Sprint2: Módulo de administración.
- Sprint3: Módulo de gestión de notas.
- Sprint4: Generación de reportes.

A su vez se realiza el diseño de sistema el cual consiste en 3 capas: presentación de negocio, capa de negocio, capa de datos.

- La capa de presentación: Esta capa es encargada de interactuar con las interfaces de usuario (interfaz web)
- La lógica de la aplicación: Esta capa contiene la lógica de la aplicación.
- La capa de acceso a datos: Esta capa es encargada de un sistema de gestor de base de datos de tipo relacional.

Por último, en esta etapa de scrum se realizan las interfaces de usuario.

b) “Juego” (Desarrollo)

En esta etapa se desarrolla la arquitectura del proyecto evaluado en 3 capas y se avanza la programación de las 4 iteraciones, en esta etapa se usan pruebas para medir la calidad del software de manera unitaria.

c) “Post” (Final)

En esta etapa el autor realiza pruebas de aceptación para comprobar que los procesos se hayan realizado correctamente de acuerdo con los casos de uso. En esta última etapa se desarrolló la implantación del módulo en los servidores de la universidad. Además, se realizó una demostración con los usuarios los cuales estuvieron conformes con lo mostrado.

Por último, menciona los siguientes puntos:

- Scrum es una herramienta muy buena para desarrollo ágil el cual permite realizar avances incrementales en el tiempo, aunque el usuario cambie de requisitos.
- Las pruebas son fundamentales cuando se quiere aseverar la calidad del software y los cumplimientos con los requisitos del cliente.
- Los reportes mostrados en la tesis podrían mejorar y mostrar información más relevante de acuerdo con la información para toma de decisiones, al igual que la

implementación de notificaciones a los usuarios sobre información importante para ellos.

3.2.2.5 Observaciones

Los casos de uso no son usuales en SCRUM, se prefieren historias de usuario que indiquen el valor, costo, riesgo que significa para los usuarios, así como criterios de aceptación.

La solución está limitada a usuarios que usan web no es adaptativa a móviles.

3.2.2.6 Conclusiones

El sistema muestra la automatización de la gestión académica en la cual se daba control manualmente y al haber una gran cantidad de personas esta tarea se complicaba.

El uso de la metodología ágil SCRUM y uso de casos de uso para la abstracción del negocio permite que se pueda centrar más en la programación y en avances incrementales mediante sprints.

3.2.2.7 Aporte

Esta tesis aporta a la nuestra el uso de software implantado con éxito en la gestión académica de un instituto de estudios, se observa la automatización y optimización de procesos manuales y el uso de la metodología ágil SCRUM que influyó en el desarrollo del software.

3222 Sistema de seguimiento de syllabus

(Flor, 2014)

32221 Estado del arte

El autor centra su atención en el proceso del seguimiento del control del sílabo de la Universidad Tecnológica de Israel, debido a que las funcionalidades del sistema encargado de ello son muy precarias. Como consecuencia, se acarrean problemas, entre ellas: la poca veracidad y el inadecuado control de usuarios.

El autor hace énfasis, en establecer un proceso adecuado para el control del seguimiento del syllabus en dicha institución educativa.

32222 Motivación del autor

El autor critica la no-interactividad que presenta el sistema de control de syllabus, lo cual demanda tiempos de holgura en las diversas tareas de los usuarios, cabe mencionar que en la actualidad se requieren sistemas interactivos que faciliten las tareas de los usuarios, en la actualidad, se presenta un abanico de tecnologías para poder alcanzar tales objetivos.

El autor busca garantizar la calidad educativa a través de la implantación del sistema propuesto, además de poder tomar correctivos ante cualquier irregularidad en cuanto algún incumplimiento del Syllabus.

32223 Descripción del aporte del autor

El aporte del autor viene a ser dado por la implantación de un sistema informático que permita hacer el adecuado seguimiento del syllabus, para cumplir con ello, se plantea desarrollar un sistema de ambiente web de tal forma que para el usuario sea un sistema amigable.

Los módulos con los que cuenta el sistema son:

- Planificación del docente.
- Control de asistencia.
- Reporte asistencia alumnos.
- Reporte planificación curricular.
- Mantenimiento de tablas (Usuario administrador)

32224 Proceso para resolver el problema

Para resolver el problema el autor hace uso de la metodología de desarrollo XP que se divide en cuatro etapas:

- La primera fase, Planificación del proyecto; donde se crean las historias de usuario.
- La segunda fase, Diseño.
- La tercera fase, Codificación.
- Por último, la cuarta fase, Pruebas.

Siendo todos estos pasos fundamentales para el desarrollo de un producto de calidad.

La primera fase, es fundamental ya que de aquí se obtiene las historias de usuario, que resultaron de la entrevista con el cliente. Se debe estar atento a lo que realmente solicita el cliente. Dentro de esta fase también evaluamos que historias de usuario van o no van. También se ve que hay iteraciones, así como las reuniones diarias

En la segunda fase, es fundamental establecer el diseño (Arquitectura); ya que este es el soporte de las fases posteriores, si no tenemos una buena base en donde poder construir nuestro edificio, pues se derrumbará.

En la tercera fase, entra a tallar la codificación del software, los estándares de codificación, la programación en parejas, controlador de versiones.

La cuarta fase está estrechamente relacionada con la tercera, aquí entra a tallar las pruebas unitarias, según el autor, la unidad de prueba corresponde a una clase.

32225 Observaciones

El uso de XP si bien es una metodología ágil tiene en uno de sus principios la programación en pareja la cual no se realizó en esta tesis.

Con este aplicativo se logra mejorar el control de asistencia al docente, así como el cumplimiento con el syllabus.

Falto realizar la implantación del sistema en dicha universidad.

32226 Conclusiones

Con la implantación de este nuevo sistema, se alcanza mayor dinamicidad en comparación con el anterior sistema. Además de ello la generación de reportes para el seguimiento de syllabus y para la asistencia del alumnado permite un mejor control de

rendimiento académico. En cuanto a la automatización del proceso de seguimiento de syllabus, se evitan las tareas manuales presentadas en el anterior sistema. Por lo tanto, con la nueva implementación del sistema, se logra optimizar el proceso de seguimiento de syllabus en dicha institución.

32227 Aporte

Aporta a nuestra tesis en el sentido que se menciona que la implantación de un sistema pudo mejorar control de asistencias tanto del profesor como del alumno, además se observa que el uso de una metodología ágil como XP fue de mucha utilidad.

3223 Sistema de seguimiento estudiantil

(Freire, 2015)

32231 Estado del arte

Las técnicas y/o conocimientos que tiene el autor son las siguientes:

- Conocimientos de las ventajas y desventajas de los distintos gestores de base de datos, entre ellos MySQL, PostgreSQL.
- Conocimientos en arquitecturas en tres capas mencionando las tecnologías: JSF, EJB y JPA; todos estos ejecutados bajo la plataforma JAVA.
- Herramientas de desarrollo de software, herramientas de integración,
- Para la construcción del desarrollo de software se usan las metodologías orientado a Objetos, así como notaciones de desarrollo UML (Diagramas de casos de uso, Diagramas de secuencia).
- Metodologías de planificación de presupuesto para el costo del proyecto del software.

32232 Motivación del autor

En primera instancia, el tema es la automatización de los procesos manuales presentados en la institución educativa “La Colina”.

Según las propias palabras del autor de la tesis: “*El manejo de un sistema de control escolar es importante para los planes de cualquier institución educativa, actualmente representan el medio eficaz para agilizar los procedimientos que en ella se desarrollan, generan mayor productividad en las instituciones que disponen de los beneficios de este tipo de sistemas.*”

Como los procesos son realizados manualmente en el departamento administrativo, consumen mayores recursos para presentar los reportes y las diferentes estadísticas necesarias para la toma de decisiones referente al ámbito académico, desconociendo así los nudos críticos que originan el bajo rendimiento del estudiante en las diferentes materias y la reprobación de estas.

El sistema que ellos presentan permitirá desvelar esos nudos críticos, para que con ello se puedan optar a decisiones más acertadas.

32233 Descripción del aporte del autor

La solución planteada por este autor es la de poder hacer un “seguimiento” al propio estudiante, durante y después de realizar las actividades educativas, con el fin de ubicar a tiempo aquellos alumnos necesiten apoyo pedagógico, a través de una serie de pasos nombrado como “Proceso de seguimiento estudiantil”.

El sistema se divide en tres principales módulos los cuales son:

- Modulo: Docente.
- Modulo: Estudiante.
- Modulo: Padre de familia.

Todo englobado bajo la arquitectura de 3 capas Cliente – Servidor.

- JSF: Capa vista
- EJB: Capa controlador
- JPA: Capa persistencia

La solución está desarrollada con la metodología de orientación a objetos.

32234 Proceso para resolver el problema

El proceso de la solución está comprendido en los siguientes casos de uso, como se muestra en la siguiente imagen (véase Figura 29).

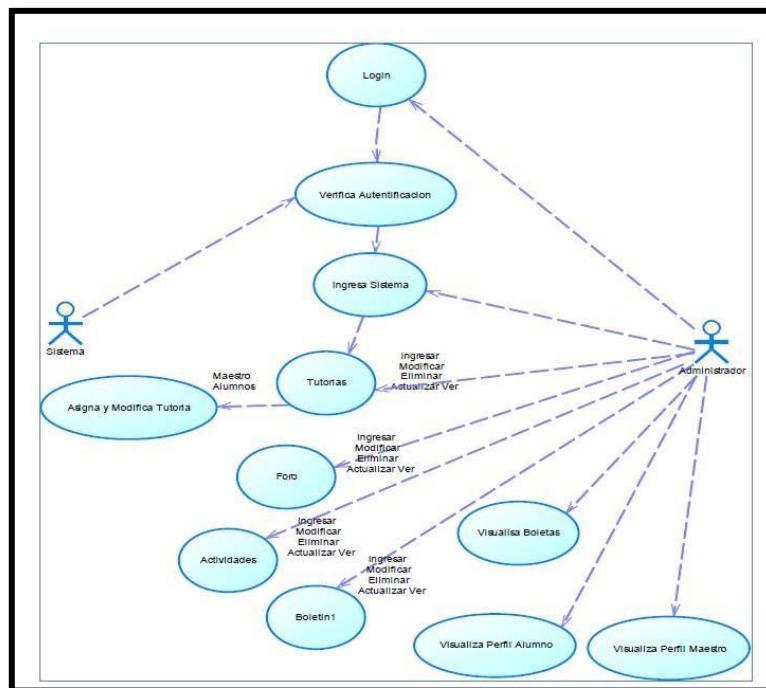


Figura 29. Casos de uso del sistema de seguimiento al estudiante(Freire, 2015)
32235 Observaciones

Según el autor, cada usuario debe acceder al sistema, pero eso no es tan interactivo; debemos de emplear notificaciones, que lleguen al dispositivo móvil del docente, apoderado del estudiante y al estudiante mismo.

Necesidad de reportes inteligentes que estén a disposición de los diversos usuarios que tiene el sistema, que no esté limitado a un navegador web, si no también poder ampliarse a un dispositivo inteligente como lo puede ser el uso en smartphones.

Por parte de la arquitectura de la solución se debería optar por una arquitectura orientada a servicios, ya que tiene un conjunto de ventajas como el de la escalabilidad.

32236 Conclusiones

Con la implantación del nuevo sistema se apoyará a cada uno de los alumnos en las diferentes asignaturas impartidas en el año lectivo, con este sistema se mantendrá al padre de familia informado sobre las actividades que realiza su hijo, fomentando así una gestión participativa. Con este sistema se logra elevar el nivel del rendimiento académico.

32237 Aporte

Aporta a nuestra tesis en el sentido que menciona las ventajas del control académico y cómo mediante el sistema se registra información y luego se procesa como reportes a diferentes roles ayuda a este fin.

3224 Diseño e implementación de una app sobre desarrollo sostenible con backend de arquitectura basada en microservices y de una reactnativefront-end app

(Chacón, 2016)

32241 Estado del arte:

La autora menciona que existe degradación del medio ambiente y que es necesario crear conciencia de cómo reducirla.

32242 Motivación del autor:

La autora está motivada porque desea apoyar en la reducción de la degradación del medio ambiente sin comprometer los recursos y posibilidades de futuras generaciones, así como el aprendizaje de nuevas tecnologías.

32243 Descripción del aporte del autor:

El aporte de la autora viene a ser dado por la implementación de un aplicativo móvil con el uso de microservicios que permita apoyar en la reducción de la degradación del medio ambiente.

Las etapas principales fueron análisis, diseño, implementación y pruebas. En el primero se definieron las funcionalidades como casos de uso, en el segundo se definió la arquitectura y el diseño de base de datos, en el tercero se avanzó con el desarrollo del software y por último la prueba del aplicativo en base a la verificación de los casos de prueba. El avance del desarrollo se dio con el uso de scrum de un periodo de 2-3 semanas en cada sprint.

Las funcionalidades del sistema propuesto abarcan las funcionalidades del registro y consulta de hechos(sucesos sobre la contaminación ambiental) y acciones dividido en secciones como agua, comida, transporte y temperatura y acciones como por ejemplo para la sección agua están acciones como ducha corta, lavar con agua fría; para la sección comida están acciones comer carne, comer pescado; para la sección temperatura están las acciones como encender calefacción, ponerse un sweater y por ultimo para la sección transporte están las acciones como usar coche o usar bicicleta.

La arquitectura de la solución está compuesta por una capa de presentación y una capa lógica, en la primera se encuentra el aplicativo móvil y en la segunda la arquitectura de microservicios. (véase Figura 30)

Se eligió aplicativo móvil debido a que menciona que en su país de procedencia España había mayor preferencia de uso de celulares sobre las computadoras. Sobre la tecnología del desarrollo móvil que eligió esta se basó en un aplicativo con el framework React Native que permite crear aplicaciones nativas en Android e IOS mediante el uso de Javascript, dado que era más práctico en vez de crear 2 aplicaciones diferentes.

Sobre el desarrollo del backend se usó la arquitectura de microservicios en vez de la arquitectura monolítica debido a que ofrecía las ventajas de menor dependencia, mejor encapsulamiento de las funciones, posibilidad de escalar, probar, implementar los microservicios de manera aislada. Los servicios como la base de datos hicieron uso de Amazonaws es decir uso servicios alojados en la nube que proveen alojamiento para aplicativos y bases de datos

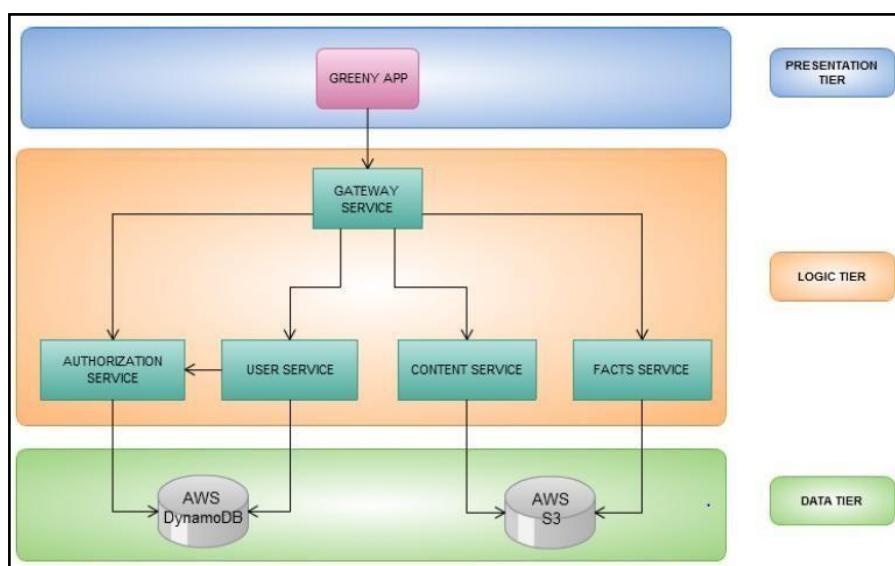


Figura 30.Diseño interno de capas del sistema (Chacón, 2016)

32244 Observaciones:

Si bien se usó scrum, se trabajó a su vez con casos de uso, diagrama de secuencias y otros documentos, se podría considerar que se usó documentación de RUP algo que en scrum no es necesario. No se especifica que hace cada sprint solo se menciona

genéricamente que duran 2-3 semanas cada sprint. Sobre la arquitectura de microservicios solo se usó el api gateway y no componentes como eureka que permite el registro y descubrimiento de servicios o config-server que permiten el registro de configuración de los servicios.

32245 Conclusiones:

Pudo cumplir los objetivos trazados con las pruebas generales mediante la validación de los casos de uso. Le sirvió de aprendizaje dado que eran todas nuevas tecnologías para la autora.

32246 Aporte

Aporta en nuestra tesis en el sentido que muestra el uso de reactnative para dispositivos móviles el cual es útil tanto para dispositivos que funcionan con Android e IOS y su coordinación con los microservicios cada uno funcionando de manera aislada.

3225 Arquitectura basada en microservicios

(Perez-Herrera, 2015)

32251 Estado del arte

Al autor le resulta muy interesante y novedoso el tema de la arquitectura de microservicios. Menciona que estas aplicaciones web basadas en esta arquitectura son el futuro y que se deben tener en cuenta muchos detalles ventajas y desventajas al momento de implementarlas.

Las características de la arquitectura de microservicios que él resalta son las siguientes:

- Independencia: Cada microservicio debe ser independiente entre sí, divididos por el límite del negocio.
- Automatización de despliegue: Permite que las aplicaciones puedan correr individualmente.
- Descentralización en el manejo de la base de datos: Ya que se permiten dividir el sistema en microservicios se hace más conveniente trabajar con una base de datos descentralizada cada cual con la función que le corresponde.

Las ventajas que el autor menciona de este tipo de arquitectura son las siguientes:

- Disminución del impacto provocado por fallos, debido a que el fallo se pueda encontrar en un microservicio el cual puede ser más fácil detectar y más rápido de corregir debido a que el servicio tiene un tamaño reducido.
- Heterogeneidad de sistemas: Permite que cada microservicio se pueda programar en diferentes lenguajes de programación de acuerdo con el tipo de funcionalidad que más se ajuste a las necesidades del negocio.

- Escalabilidad eficiente (se puede realizar en cada microservicio dependiendo del nivel de uso que tengan estos se le asignaran más recursos para que puedan manejarla para eso menciona el uso de un balanceador de carga)
- Flexibilidad y facilidad en actualización (el microservicio debe ser posible de actualizarse a nuevas versiones incluso realizar cambios bruscos, incluso si se ve que su uso ya no es adecuado se puede quitar o reemplazar por uno con mejores características.)
- Capacidad organizativa del equipo de trabajo: se puede dividir mejor el trabajo debido a que las funcionalidades están disgregadas, esto debido a que la coordinación va a ser más eficaz.
- Eficiencia: es posible que se ahorre en costes de mantenimiento y ahorro energético al usar la computación en la nube como lo están haciendo muchas compañías.

Las desventajas que el autor menciona de este tipo de arquitectura son las siguientes

- Complejidad de la aplicación: Debido a que la mayoría está acostumbrado al desarrollo de aplicaciones monolíticas, menciona que puede ser complicado para tratar temas como la comunicación entre servicios o dividirlos ya que son nuevos problemas que abordarían.
- Difícil migración desde aplicaciones monolíticas: En muchos casos resulta ser muy complejo, debido a que las funciones están muy entrelazadas haciendo muy difícil el cambio.
- Comunicación entre microservicios: Cuando más grande es el aplicativo más grande se volverán las comunicaciones y es posible que una de estas pueda fallar en el tiempo, para eso es conveniente una buena subdivisión y contemplar que hacer en caso de fallos
- Seguridad en las comunicaciones: Este tipo de arquitectura usa mucho las comunicaciones por lo cual deben realizarse técnicas de cifrados en los casos que se requieran.

32252 Descripción del aporte del autor

El autor propone la arquitectura mostrada (véase Figura 31) para el caso de una tienda online, en la cual se muestran los siguientes componentes:

La interfaz de usuario: la pantalla principal con la cual el usuario se comunica contiene 4 vistas: Vista inicial (Login), Vista de registro del usuario, Vista de presentación del catálogo de productos disponibles, y la vista del carrito de compras. Se hizo uso de lenguajes como HTML5, CSS, JavaScript ya que trabajan bien con NodeJS, además se usó el framework Twitter Bootstrap

Microservicio “gestión de usuario y peticiones”: este microservicio permite poder iniciar la sesión al usuario de acuerdo con los accesos que este tenga; además permite

realizar peticiones es decir cuando requiere una función en particular este microservicio se encarga de comunicarse con otro microservicio para resolver la petición realizada.

Microservicio “Búsqueda”: Este microservicio se encarga de realizar una petición HTTP tipo POST que se encarga de consultar un modelo en la base de datos de productos disponibles.

Microservicio “Ordenación de catálogo”: Este microservicio se encarga de ordenar el catálogo de productos disponibles en función de determinadas características como el precio o la talla de los modelos.

Microservicio “Carrito de compra”: Este microservicio se encarga de almacenar los productos seleccionados por el cliente, para esto y como no usa una base de datos relacional se guarda el nombre de usuario como identificador en la base de datos Redis. También tiene la funcionalidad de poder realizar la compra de los productos correspondiente a aquellos productos seleccionados en el carrito de compras.

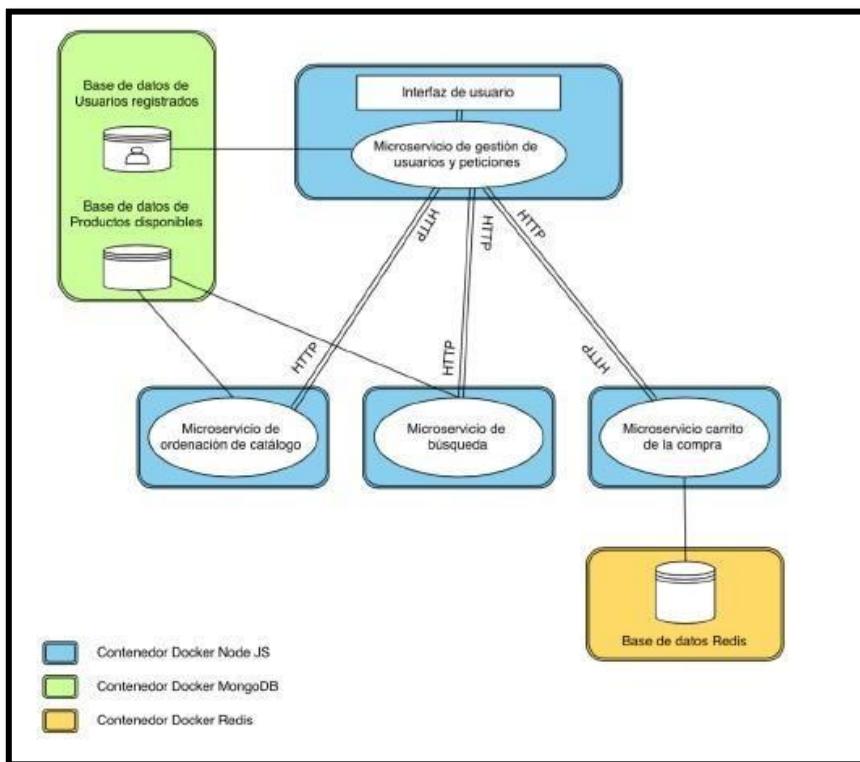


Figura 31. Arquitectura propuesta: Microservicios de venta online de productos(**Perez-Herrera, 2015**)

Entre la tecnología usada menciona lo siguiente:

- El uso de NodeJs ya que permite la programación por el lado servidor apoyándose en JavaScript, además permite la construcción despliegue y escalabilidad de servicios web de manera sencilla.

- El uso de Docker ya que ofrece máquinas virtualizadas que pueden correr al mismo tiempo, para el despliegue de los servicios y el despliegue de las bases de datos. A continuación, se muestra una imagen comparando el uso de recurso de una máquina virtual y contenedores Docker(véase Figura 32), se aprecia que una máquina virtual tiene un propio sistema operativo virtual en contraparte Docker cuenta con un solo motor que forma contenedores para que cada aplicación pueda correr en un entorno de esta manera se ahorran muchos recursos.

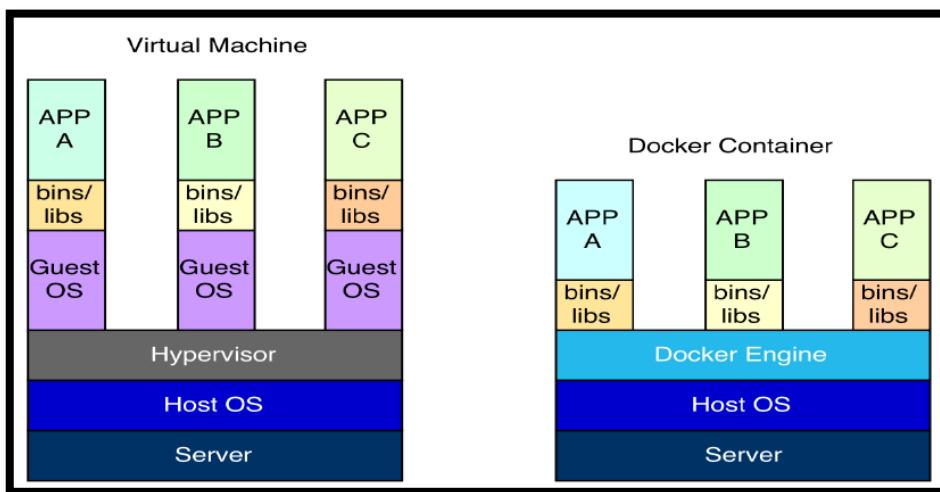


Figura 32. Diferencias entre máquina virtual y contenedor docker(Perez-Herrera, 2015)

32253 Observaciones

El trabajo de la tesis se limita en un estudio teórico y un ejemplo de aplicativo que pueda servir de guía para entender esta arquitectura.

Sobre la arquitectura de microservicios propuesta hay 2 servicios que accedan a una sola base de datos lo cual hace que estos no sean independientes entre sí.

32254 Conclusiones

- El artículo sintetiza varias características, ventajas y desventajas de los microservicios y realiza una exemplificación de éste en el caso de ventas online de productos.
- Hubiera sido bueno poder apreciar una aplicación real funcionando y que ventajas dio al negocio en sí.
- Se aprecia el uso de varias bases de datos dependiendo del problema que solucionan cada uno de estas.
- Se aprecia el uso de varios microservicios trabajando colaborativamente con NodeJS y Docker, hubiera sido bueno apreciar otros lenguajes como java o .net debido a que tienen bastante soporte en la actualidad.

32255 Aporte

Esta tesis aporta a la nuestra en el uso de la arquitectura de microservicios la cual se ve que tiene varias ventajas de las cuales podemos mencionar la disminución del impacto por fallos, heterogeneidad de sistemas, escalabilidad eficiente, flexibilidad al cambio, eficiencia en costes, mejor capacidad organizativa del trabajo.

3.3 Metodología de desarrollo

3.3.1 Metodología scrum

3.3.1.1 Definición

Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Pone su atención y hace foco sobre valores y prácticas de gestión, en vez de requerimientos, prácticas de desarrollo, implementación y otras cuestiones técnicas. Esta metodología delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles, es decir, que es flexible y los integrantes del equipo pueden optar por organizar la forma de interactuar entre ellos. (Fernandez & Cadelli, 2014)

La terminología “Scrum” procede del deporte llamado rugby, donde se designa al acto de preparar el avance del equipo en unidad pasando la pelota a uno y otro jugador. Scrum es adaptable, ágil, autoorganizado y con pocos tiempos muertos. Esta metodología ágil fue desarrollada por Jeff Sutherland y elaborada más formalmente por Ken Schwaber. Poco tiempo después Sutherland y Schwaber se unieron para refinar y extender Scrum. Se la ha llegado a conocer como una herramienta de hiperproductividad. Schwaber se dio cuenta entonces de que un proceso necesita aceptar el cambio, en lugar de esperar predictibilidad. Se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles. Toma el cambio como una forma de entregar al final del desarrollo algo más cercano a la verdadera necesidad del cliente. Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.(Fernandez & Cadelli, 2014)

De acuerdo con(Fernandez & Cadelli, 2014), se basa en los principios ágiles:

- Satisfacer al cliente mediante tempranas y continuas entregas de software que aporten valor. Entregar software funcional lo más pronto posible.
- Predisposición y respuesta al cambio dado que presentan una ventaja competitiva.
- Entregar frecuentemente software que funcione y no planificaciones o documentación de análisis o diseño.
- La gente del negocio y los desarrolladores deben trabajar juntos, el proceso de negocio debe ser guiado por el cliente por lo que interactuar con el equipo es frecuente.
- El proyecto debe contar con individuos motivados y brindarles apoyo.

- Fortalecer la comunicación y la colaboración.
- El software que funciona es la medida del progreso.
- El desarrollo debe ser sostenible por los desarrolladores y usuarios deben colaborar para que esto suceda.
- Atención continua en la calidad técnica y buen diseño mejora la agilidad.
- La simplicidad es esencial, camino simple consistente con objetivos claros.
- El equipo debe ser autoorganizado, ellos determinan los objetivos que persiguen.
- Reflexión sobre cómo llegar a ser más efectivo en intervalos de tiempo, adaptarse a nuevos entornos.

3.3.1.2 Características de scrum

De acuerdo con(Fernandez & Cadelli, 2014), entre las características que se resaltan de scrum son la siguientes:

- Más que una metodología de desarrollo es una herramienta para gestionar proyectos.
- No contiene definiciones en áreas de ingeniería.
- Con visión de que el trabajo es efectuado por equipos autoorganizados y autodirigidos, logrando motivación, responsabilidad y compromiso.
- Está basada en un proceso constructivo iterativo e incremental donde las iteraciones tienen duración fija.
- Contiene definición de roles, prácticas y productos de trabajo escritas de forma simple.
- Está basada en un conjunto de valores y principios.

3.3.1.3 Flujo scrum

En la siguiente imagen (véaseFigura 33), se puede visualizar el flujo de trabajo de la metodología Scrum. Es decir, desde el product backlog(lista de requerimientos), sprint backlog (requerimientos priorizados para el sprint, las iteraciones en el ejemplo de 1 semana, la reunión diaria en el ejemplo cada 24 horas, y por último los derivables(entregables) del sprint.

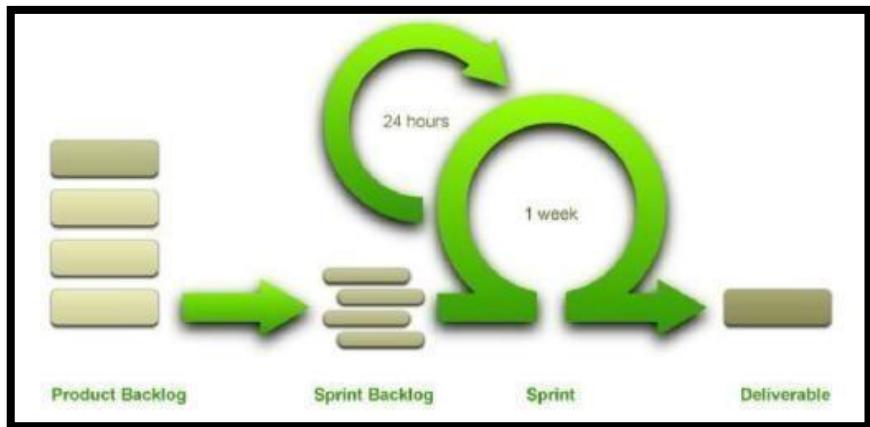


Figura 33. Flujo de trabajo de scrum(Fernandez & Cadelli, 2014)

3.3.1.4 Elementos de scrum

De acuerdo con(Fernandez & Cadelli, 2014), los elementos que componen Scrum son diversos. A continuación, se mencionan y posteriormente se realiza una explicación de estos.

- Roles: Product Owner, Scrum Master, Team (Equipo)
- Lista de requerimientos
- Product Backlog
- Sprint: Planificación (Planning), Sprint Backlog, Scrum Estimaciones, Builds continuos, Revisión del Sprint, Reunión retrospectiva
- Valores: Foco, comunicación, respeto y coraje

3.3.1.5 Roles

De acuerdo con(Fernandez & Cadelli, 2014), la dimensión del equipo total de Scrum no debería ser superior a veinte personas. Si hay más, lo más recomendable es formar varios equipos. No hay una técnica oficial para coordinar equipos múltiples, pero se han documentado experiencias de hasta 800 miembros, divididos en Scrum de Scrum definiendo un equipo central que se encarga de la coordinación, las pruebas cruzadas y la rotación de los miembros. Scrum tiene una estructura muy simple. Todas las responsabilidades del proyecto se reparten en 3 roles:

- a) **Productowner** (Dueño del producto) Representa a todos los interesados en el producto final. Es el responsable oficial del proyecto, gestión, control y visibilidad de la lista de acumulación o lista de retraso del producto (Product Backlog). Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar.

Sus áreas de responsabilidad son:

- Financiación del proyecto.
- Requisitos del sistema.
- Retorno de la inversión del proyecto.

- Lanzamiento del proyecto.
- b) **Scrum Master (Líder del proyecto)** Responsable del proceso Scrum, de cumplir la meta y resolver los problemas. Así como también, de asegurarse que el proyecto se lleve a cabo de acuerdo con las prácticas, valores y reglas de Scrum y que progrese según lo previsto. Interactúa con el cliente y el equipo. Coordina los encuentros diarios, y se encarga de eliminar eventuales obstáculos. Debe ser miembro del equipo y trabajar a la par.
- c) **Team (Equipo)** Responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software, es decir, de convertir el product backlog en un software entregable. El equipo tiene la autoridad para reorganizarse y definir las acciones necesarias o sugerir eliminación de impedimentos. Características del equipo:
- Autogestionado
 - Autoorganizado
 - Multifuncional
 - El número ideal para la conformación de un equipo es entre 8 y 10 personas

3.3.1.6 Lista de requerimientos

La primera actividad es armar una lista exhaustiva de los requerimientos originales del sistema. Luego se procede a ver qué requerimientos son realmente necesarios, cuáles pueden posponerse y cuáles eliminarse. Para ello debe identificarse un representante con capacidad de decisión, priorizar los requerimientos en base a su importancia y acordar cuáles son los prioritarios para la fecha de entrega. La poda de requerimientos es una buena práctica implícita en modelos ágiles, se hace lo que el cliente realmente desea, no más, se debe realizar una priorización de requerimientos.(Fernandez & Cadelli, 2014)

3.3.1.7 Productbacklog

Se puede armar en base a los requerimientos priorizados. Este es una forma de registrar y organizar el trabajo pendiente para el producto (actividades y requerimientos). Es un documento dinámico que incorpora constantemente las necesidades del sistema. Por lo tanto, nunca llega a ser una lista completa y definitiva. Se mantiene durante todo el ciclo de vida (hasta finalizar el sistema) y es responsabilidad del ProductOwner.(Fernandez & Cadelli, 2014)

3.3.1.8 Sprint scrum

Está basado en el control empírico de procesos. Se utiliza cuando la capacidad de predicción es vaga, la incertidumbre alta o el proceso es demasiado complejo para ser modelado y definido. En el enfoque empírico de control de procesos se establecen reglas simples y se crea una disciplina de inspección frecuente para adaptarse rápidamente a situaciones imprevistas o problemas. Un Sprint es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad. Constituye el núcleo de

Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas “carreras”.(Fernandez & Cadelli, 2014)

De acuerdo con(Fernandez & Cadelli, 2014), las características del Sprint son las siguientes:

- Duración máxima de 30 días.
- Durante el Sprint no se puede modificar el trabajo que se ha acordado en el Backlog.
- Sólo es posible cambiar el curso de un Sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por alguna de las razones siguientes:
 - La tecnología acordada no funciona.
 - Las circunstancias del negocio han cambiado.
 - El equipo ha tenido interferencias.

3.3.1.9 Planificación

De acuerdo con(Fernandez & Cadelli, 2014), se planifica en detalle el trabajo al inicio de cada Sprint asumiendo que los objetivos no van a cambiar durante el mismo, de esta manera se atenúa el riesgo. Los aspectos para tener en cuenta sobre la planificación de un Sprint:

- Una determinación general de alcance, frecuentemente basada en una EDT (Estructura de División del Trabajo).
- Estimaciones de esfuerzo de alto nivel realizadas durante la etapa de concepción del proyecto.
- Esfuerzo dedicado a labores de soporte o de preparación de los ambientes requeridos por el proyecto.
- Esfuerzo asociado a las reuniones diarias, de planificación y de revisión.
- Requerimientos de recursos de infraestructura o logísticos (máquinas, redes, licencias, papel, pizarras, etc.).
- Habilidades presentes y necesarias en el equipo.
- Restricciones asociadas al conocimiento del negocio, la tecnología o externas (legales, reglamentarias, estándares, etc.).
- Rol del Scrum Master durante la planificación:
 - Dirige la planificación.
 - Es vínculo entre el equipo y el ProductOwner del proyecto.
 - Registra problemas y riesgos detectados durante la planificación.

- Registra las tareas, asignaciones y estimaciones.
- Inicia el Backlog del Sprint.

3.3.1.10 Sprint backlog

De acuerdo con(Fernandez & Cadelli, 2014), el sprint backlog se refiere al trabajo o tareas determinadas por el equipo para realizar en un Sprint, en otras palabras, es la conversión de tareas a un producto funcional. Entre las características se resaltan:

- Las tareas se estiman en una duración entre 1 a 20 horas de trabajo. Las de mayor duración deben intentar descomponerse en subtareas de ese rango de tiempo.
- La estimación se actualiza día a día.

3.3.1.11 Scrum diario (Daily)

Scrum asume que el proceso es complejo y que es necesario inspeccionarlo frecuentemente, por eso se realiza una reunión diaria de seguimiento. El encuentro diario impide caer en el dilema señalado por Fred Brooks: “¿Cómo es que un proyecto puede atrasarse un año? Un día a la vez”.(Fernandez & Cadelli, 2014)

El foco de la reunión es determinar el avance en las tareas y detectar problemas o “bloqueos” que estén haciendo lento el progreso del equipo o que eventualmente impidan a un equipo cumplir con la meta del Sprint. La idea es que ningún problema quede sin resolver o, por lo menos, sin iniciar alguna acción de respuesta dentro de las 24 horas después de su detección. La reunión es además un espacio definido para que cada miembro del equipo comunique a los demás el estado de su trabajo y por lo tanto reafirme el compromiso. (Fernandez & Cadelli, 2014)

3.3.1.12 Rol del scrum master durante el scrum

De acuerdo con(Fernandez & Cadelli, 2014), dentro de los roles del Scrum Master durante el Scrum encontramos los siguientes:

- Dirigir la reunión y mantener el foco de atención.
- Realizar preguntas para aclarar dudas.
- Registrar, escribir y/o documentar los problemas para su resolución después de la reunión.
- Asegurarse que los miembros cuenten con el ambiente adecuado para la reunión.

3.3.1.13 Estimaciones

Las estimaciones se realizan por primera vez en la reunión de planificación al inicio del Sprint. Posteriormente, las tareas se reestiman todos los días y se registran sus cambios en el Backlog del Sprint.Esta actividad puede ser realizada inmediatamente antes o después del Scrum diario.(Fernandez & Cadelli, 2014)

De acuerdo con(Fernandez & Cadelli, 2014), algunas claves para la estimación son las siguientes:

- Siempre se realizan estimaciones de esfuerzo, no de duración.
- Siempre se estima el esfuerzo total pendiente para terminar la tarea, no se estima el esfuerzo consumido.
- Se buscan unidades manejables, lo usual es que estén en un mínimo de 2 horas y un máximo de 20. Si la tarea es muy corta se trata de juntarla con otras relacionadas. Si la tarea es muy grande se trata de descomponerla.

3.3.1.14 Builds continuos y pruebas básicas

De acuerdo con(Fernandez & Cadelli, 2014), la estrategia que generalmente se utiliza es la de Builds continuos y “smoke test” (prueba básica para la funcionalidad del sistema). El procedimiento de Builds continuos es el siguiente:

- 1) Los programadores desarrollan según el Backlog del Sprint, y al finalizar, notifican al integrador.
- 2) El integrador toma el código y lo integra con el resto del producto.
- 3) Se compila el software y se prueba “por arriba” el producto, para verificar que no se haya roto.
- 4) Si se encuentran problemas se devuelve al desarrollador. Se notifica al equipo que hay una nueva versión “estable” del código para usar como base.

3.3.1.15 Revisión del sprint

El objetivo de la reunión de revisión es presentar el producto o porción del producto desarrollada por el equipo a los usuarios. La reunión se utiliza para detectar inconformidades mayores que se vuelven elementos del Backlog de producto y que eventualmente se resuelven en el siguiente Sprint. A la reunión asisten el equipo, el Scrum Master, el ProductOwner y todas las personas implicadas en el proyecto. (Fernandez & Cadelli, 2014)

3.3.1.16 Reunión retrospectiva scrum

Involucra el concepto de mejora continua a través de las reuniones de retrospección. Las reuniones buscan detectar los puntos positivos y negativos del Sprint para generar propuestas de mejora para futuros Sprints. Las reuniones de retrospección son el concentrador del aprendizaje organizacional sobre el Scrum. Los puntos positivos y negativos se registran y se definen ítems de acción para cada uno. Los ítems de acción definidos se toman en cuenta en los siguientes Sprints. A este tipo de reuniones asisten el equipo y el Scrum Master, y opcionalmente el ProductOwner del producto.(Fernandez & Cadelli, 2014)

3.3.2 Metodología programación extrema

3.3.2.1 Definición

La metodología XP (extreme programming: programación extrema) es una metodología ágil es un conjunto de buenas prácticas para el desarrollo de software, se enfoca en resultados y en reducir procesos del desarrollo de software.(Pardo, Triana, & Forero, 2014)

3.3.2.2 Objetivos

De acuerdo con(Pardo, Triana, & Forero, 2014), se mencionan los siguientes objetivos que esta metodología busca:

- Establecer mejores prácticas de desarrollo de software.
- Optimizar la ejecución de proyectos.
- Garantizar la calidad del software.
- Potenciar el trabajo en equipo al máximo.

3.3.2.3 Fases

De acuerdo con(Pardo, Triana, & Forero, 2014), la metodología extrema está dividida en cuatro fases(véaseFigura 34):

1. **Planificación:** Es la fase donde se hace un buen entendimiento entre la parte empresarial y técnica mediante reuniones, historias de usuario, planes de entrega, planes de iteración. Está delimitado por el alcance de la presente versión.
2. **Diseño:** En esta fase se encuentran elementos como metáforas, tarjetas CRC (clase, responsabilidad y colaboración) y reciclaje.
3. **Codificación:** En esta fase se definen las parejas de programación, integración continua (integrar y probar el código en la versión resultante) y la propiedad colectiva (nadie es dueño del código, otro programador puede hacer cambios si es conveniente).
4. **Pruebas:** Se definen 2 tipos en esta metodología:
 - La prueba unitaria: Se hace las pruebas del código paso a paso, para reducir el impacto de errores cuando se hacen modificaciones.
 - La prueba de aceptación: Evalúa si el producto puede adaptarse a otros ambientes, se resaltan las funcionalidades que tienen más importancia por el usuario.

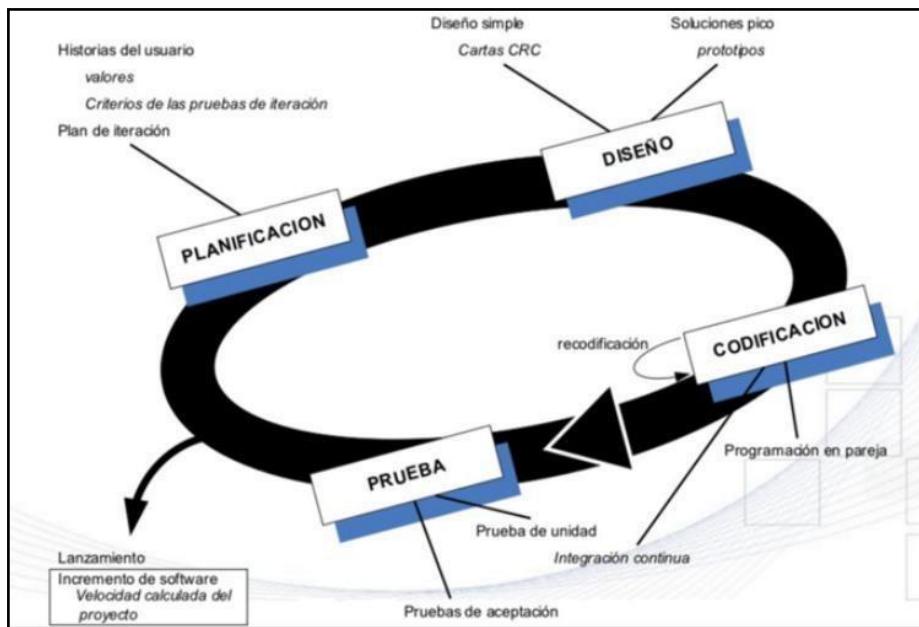


Figura 34.Fases de XP(Pardo, Triana, & Forero, 2014)

3.3.2.4 Procesos de desarrollo

De acuerdo con(Pardo, Triana, & Forero, 2014), los procesos de desarrollo están integrados con las fases de la metodología XP, se menciona continuación los siguientes procesos:

- Relación con el cliente:** Se considera al cliente importante en el proceso de desarrollo, éste participa en las reuniones, abordan las historias de usuario (lista de requisitos e importancia), además asisten a reuniones de planificación.
- Planificación del proyecto:** Es importante contar aquí con las historias de usuario ya que reflejan las necesidades del negocio. Se planifican la velocidad del proyecto, entrega, iteraciones y para cada iteración se definen y ordenan las tareas.
- Programación y pruebas:** La programación se realiza en parejas debido a que se considera que garantiza una mejor calidad del producto, para las pruebas se toma como base las historias de usuario y sus tareas para así validar las pruebas de aceptación y poder generar una nueva versión.

3.3.2.5 Valores

De acuerdo con(Pardo, Triana, & Forero, 2014), en la programación extrema se promueven los siguientes valores:

- Comunicación:** Debe haber una buena comunicación entre los equipos que desarrollan para intercambiar información correcta de manera continua y rápida.
- Simplicidad:** Se recomienda un enfoque simple con relación al proceso y la codificación.
- Retroalimentación:** Se recomienda contar con retroalimentación entre los actores que se obtienen cuando se realizan pruebas, integraciones, versiones y entregas frecuentes.

4. **Coraje:** Las personas deben expresar su punto de vista sin inhibiciones en el desarrollo del proyecto.
5. **Valentía:** Se menciona que no se debe temer a la programación ni postponerla.
6. **Respeto:** Las decisiones de equipo se deben respetar ante decisiones individuales.

3.3.2.6 Principios

De acuerdo con(Pardo, Triana, & Forero, 2014), se mencionan los siguientes principios:

- La prioridad es la satisfacción al cliente.
- Se aceptan los cambios y se toman como ventaja competitiva
- Las entregas de software son frecuentes
- Debe haber un fuerte vínculo entre programadores y clientes.
- Los proyectos deben tener individuos motivados
- Obtener la información directamente del cliente.
- El software avanzado es la mejor medición del progreso.
- Se promueve desarrollo de software dado las horas definidas sin tiempos adicionales.
- El buen diseño de software mejora la agilidad, para que aumentar nuevas funcionalidades no sea tedioso.
- La simplicidad es importante, para no hacer de más basta con cumplir los requisitos del cliente.
- Mejorar la arquitectura surge en las relaciones dinámicas entre los miembros del equipo.
- Reflexionar sobre la eficacia, la cual es reflejada por la simplicidad y la flexibilidad.

3.3.3 Metodología Iconix

3.3.3.1 Definición

De acuerdo con(Porras, 2019), Iconix es una metodología ágil con principios de desarrollo incremental e iterativo, presenta un proceso puro, práctico y simple con el análisis y representación del problema de forma eficaz, usa UML (Lenguaje de Modelamiento Unificado) de forma reducida es por ello por lo que no genera tanta documentación. Entre sus características se destacan:

- A diferencia de extreme programming (XP) comprende etapas de análisis y diseño
- Uso simplificado de UML: modelo de dominio, casos de uso, diagrama de robustez, diagrama de secuencia, diagrama de clases.
- Aporta buen nivel de trazabilidad entre requisitos y el producto de software

Se muestra la concepción de la forma de trabajo en Iconix (véase Figura 35), en la cual se puede diferenciar el modelo dinámico del estático. Aquí se producen productos de

software que se desarrollan incrementalmente y de forma paralela. El modelo estático se incrementa y es refinado por el modelo dinámico.

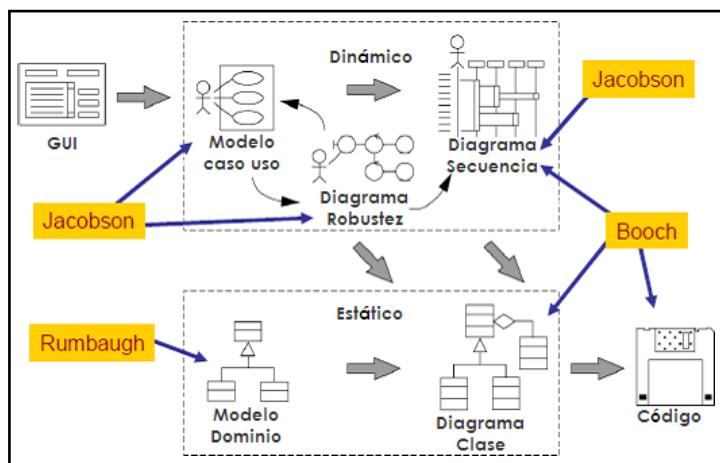


Figura 35. Esquema de metodología Iconix(Porras, 2019)

Las fases que comprende son las siguientes: análisis de requisitos, diseño preliminar, diseño detallado, implementación, pruebas y revisiones las cuales se detallarán a continuación.(Porras, 2019)

3.3.3.2 Fase: Análisis de requisitos

En esta fase se definen los requisitos funcionales y no funcionales, se construye el modelo de dominio, se realizan prototipos de interfaces gráficas, se definen los casos de uso y se organizan por paquetes.(Porras, 2019)

El modelo de dominio consiste en encontrar objetos (que representen cosas y conceptos en el mundo real), sirve como entrada para los casos de uso. No es igual que el modelo de clases o modelado de datos, está más enfocada en el dominio del problema.(Porras, 2019)

Los casos de uso son la parte central del desarrollo, consiste en un documento narrativo que describe la secuencia de eventos que realiza un actor que usa el sistema para culminar su proceso. Cada actor representa el rol de una entidad externa del sistema puede ser un usuario, hardware u otro sistema, no representa a un cargo de una persona se centra en los roles que cumple en el sistema.(Porras, 2019)

3.3.3.3 Fase: Diseño preliminar

Aquí se considera el análisis de robustez que permite analizar los casos de uso e identificar un conjunto de objetos por cada caso de uso. Estos objetos pueden ser interfaz (capa de presentación de interacción entre actores y el sistema), entidad (objetos del modelo de dominio), control (representa mediante una acción es intermedio entre la entidad e interfaz).(Porras, 2019)

Luego de contar con el análisis de robustez se realiza una verificación que consiste en que el diagrama de robustez, el modelo de dominio y descripción de casos de uso coincidan entre sí.(Porras, 2019)

3.3.3.4 Fase: Diseño detallado

En esta fase participan los diagramas de secuencia que definen el comportamiento de un caso de uso, en la cual se muestra la colaboración dinámica entre objetos del sistema, y se realizan interacciones entre estos mediante una secuencia de pasos. Toma como entrada los casos de uso y el diagrama de robustez.(Porras, 2019)

A continuación, se muestra una imagen de cómo se realiza la construcción del diagrama de secuencia (véase Figura 36).

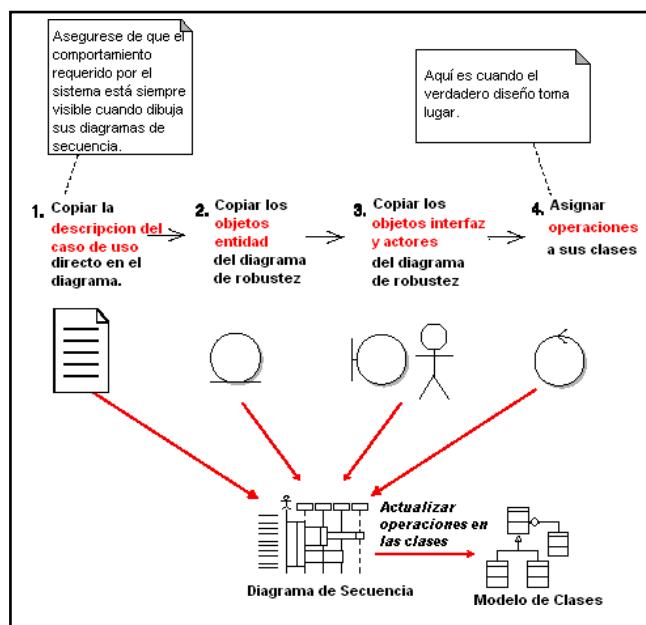


Figura 36.Construcción del diagrama de secuencia(Porras, 2019)

3.3.3.5 Fase: Implementación

Primero se debe definir la arquitectura técnica en base a los diagramas de secuencia, luego codificar tal como está definido en los diagramas de secuencia, además se debe contar con pruebas unitarias. (Porras, 2019)

3.3.3.6 Fase: Pruebas y revisiones

Las pruebas se derivan de los casos de uso y originan casos de prueba que verifican los diferentes escenarios y que la implementación fue correcta. (Porras, 2019)

De acuerdo con(Porras, 2019), hay 2 tipos de pruebas que se usan en esta fase:

- Prueba unitaria: Prueba componentes individuales de software, se ejecuta en cada integración de software en el desarrollo.
- Prueba de aceptación: Prueba conducida por el cliente que valida los requisitos del software.

3.4 Aplicativo móvil

3.4.1 Ionic

3.4.1.1 Definición

Es un kit de herramientas de código abierto que permite crear aplicaciones móviles y de escritorio usando tecnologías web como HTML, CSS y JavaScript. Está centrado en la experiencia del usuario frontend o la interacción de un aplicativo (controles, interacciones, gestos y animaciones). Fácil de aprender y se integra bien con otras librerías o frameworks como Angular o puede ser usado sin un framework frontend solo con scripts incluidos. Para la fecha de esta tesis, Ionic Framework tiene integración oficial con Angular, pero el soporte a Vue y React están en desarrollo. (Ionic Framework Documentation, 2019)

3.4.1.2 Componentes UI

Ionic Framework es una librería de componentes de UI (interfaz gráfica) los cuales son reusables y sirven para armar los bloques de construcción del aplicativo, todas desarrolladas con los estándares web HTML, CSS y JavaScript. Estos componentes son preconstruidos y permiten un alto grado de personalización para que cada aplicativo pueda tener su propio estilo. (Ionic Framework Documentation, 2019)

3.4.1.3 Continuidad de plataforma

Es una característica integrada en el framework Ionic que permite que los desarrolladores de aplicativos puedan usar el mismo código base para múltiples plataformas. Cada componente Ionic se adapta al estilo de cada plataforma en el cual están corriendo. Por ejemplo, IOS tiene su propio lenguaje de diseño y Android usa el lenguaje de diseño llamado “diseño material”, y para los aplicativos webs progresivos (PWA) se usa por defecto “diseño material”, pero en sí los temas que usará cada plataforma son completamente configurables. (Ionic Framework Documentation, 2019)

3.4.1.4 Navegación

Los aplicativos webs tradicionales usan historia lineal esto quiere decir que pueden navegar entre páginas dado que se almacena todo en una pila de historial lineal del navegador. En contraste los aplicativos móviles utilizan la navegación paralela “no lineal” teniendo por ejemplo pilas de navegación por pestaña en el caso de una interfaz con pestañas.

Ionic se adopta a ambos enfoques al soportar tanto navegación lineal como paralela, para la fecha de la tesis se recomienda usar el enrutador de Angular que está funcionando con Ionic 4 para versiones anteriores, Ionic manejaba su propio enrutador personalizado. (Ionic Framework Documentation, 2019)

3.4.1.5 Acceso nativo

La importancia de contar con aplicativos realizados con tecnologías web es que pueden correr en cualquier plataforma como escritorio, celulares, tablets, etc. Esto dado a que el mismo código base de los aplicativos Ionic pueden funcionar en varias plataformas

porque está basado en estándares web y APIs comunes que son compartidas por todas éstas. (Ionic Framework Documentation, 2019)

Para los aplicativos móviles se hacen uso de “webviews” que son proveídos por SDKs(Herramientas de desarrollo de software) de IOS y Android que permiten renderizar cualquier aplicativo Ionic y permitir el acceso nativo a funciones nativas del SDK. Para acceder a las funciones nativas del SDK se pueden usar proyectos como Capacitor y Cordova, esto permite a desarrolladores usar herramientas comunes de desarrollo web y tener acceso a características nativas del dispositivo como acelerómetro, cámara, GPS y muchos otros. (Ionic Framework Documentation, 2019)

3.4.1.6 Temática

El framework Ionic está construido usando CSS lo que permite tomar ventaja de la flexibilidad de las variables de CSS. Esto hace posible un buen diseño del aplicativo según los estándares web, pero además se pueden cambiar y personalizar y así crear un diseño propio. (Ionic Framework Documentation, 2019)

3.4.1.7 Arquitectura

Como se puede apreciar en la imagen (véase Figura 37), Ionic hace uso de servicios y directivas para poder usar las tecnologías web(html5, angular, css) para el diseño de la interfaz de usuario del aplicativo y para acceder a las funciones de éste usan directivas y servicios dados por apache cordova que pueden acceder a funciones nativas del dispositivo.



Figura 37. Arquitectura de Ionic para plataforma móvil(Christoph, Rösch, & Schuster, 2018)

3.4.1.8 Ventajas

De acuerdo con(Ospiva, 2019), entre las ventajas que ofrece ionic están:

- Es fácil de correr el aplicativo en un dispositivo real o un emulador, dado que ionic provee de empaquetado de aplicación nativa, servidor de desarrollo y recarga en vivo.
- El servidor de desarrollo ofrece recarga en vivo que permite que los nuevos recursos y cambios se vean en el navegador cuando el código cambia, esto mejora los tiempos de desarrollo.

- El servidor de desarrollo ofrece un reporte de errores como su traza para ubicar el problema.
- Existen una gran cantidad de componentes UI creados listos para usar en el aplicativo Ionic.
- Se provee de configuración específica de acuerdo con la plataforma para que sea fácil definir el estilo que tendrán las diferentes plataformas.

3.4.1.9 Desventajas

De acuerdo con(Ospiva, 2019), la principal desventaja que tiene es sobre el rendimiento del aplicativo ionic ya que es un framework híbrido y necesita de webviews por esto el rendimiento es menor en comparación con un aplicativo nativo.

3.4.2 NativeScript

3.4.2.1 Definición

Es un framework de código abierto para construir aplicativos móviles multiplataformas para Android e IOS, el cual es creado y mantenido por Telerik. Está escrito en una combinación de Javascript, XML y CSS (véaseFigura 38), el componente JavaScript corre la lógica de negocio, acceso a datos y el control del flujo del aplicativo, la Proción en XML define la interfaz gráfica de usuario (UI) y por último el CSS es usado para dar estilo al UI tal cual como un aplicativo en HTML. NativeScript se renderiza sus componentes en XML en elementos nativos de UI del aplicativo de acuerdo con la plataforma ya sea Android o IOS. (Branstein & Branstein, 2017)

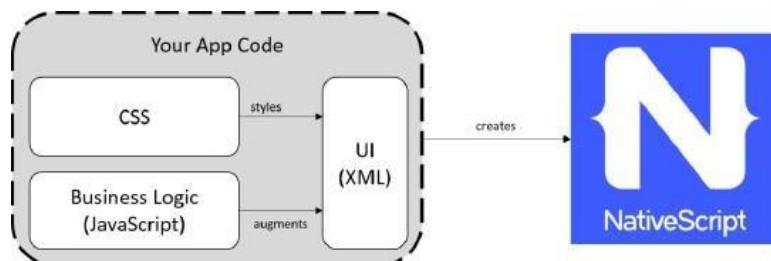


Figura 38. Uso de javascript,css y xml(Branstein & Branstein, 2017)

3.4.2.2 Ventajas

A diferencia de Xamarin que es un framework híbrido cross-compilednativescript usa JIT compiled que permite que los aplicativos sean compilados en tiempo de ejecución y no antes de la ejecución del aplicativo. A diferencia de compilar los códigos en un aplicativo nativo, native script tiene una máquina virtual en JavaScript el código corre en dicha máquina. (véaseFigura 39) (Branstein & Branstein, 2017)

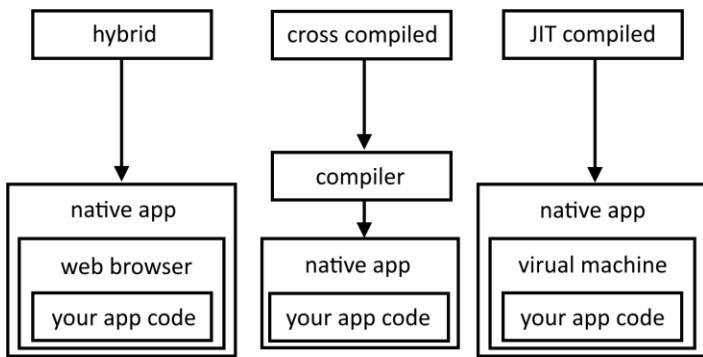


Figura 39. Como corre compilación JIT en los dispositivos (Branstein & Branstein, 2017)

El tipo de aplicativos que se pueden desarrollar se menciona que son de consumidores o líneas de negocios es decir aplicativos para empresas, redes sociales, etc.

3.4.2.3 Desventajas

Para juegos que usen gráficos 3D no es conveniente debido a que nativescript agrega una capa de abstracción para esos casos si se considera escribir en código nativo de Android o IOS. (Branstein & Branstein, 2017)

3.4.2.4 Arquitectura

Como se muestra en la imagen (véase Figura 40), se pueden apreciar los componentes de NativeScript. El código del aplicativo como se mencionó que contiene la lógica de negocio que responde a eventos e interacciones, CSS, y UI. El componente NativeScript CLI agrupa el código, el entorno de ejecución de Nativescript, sus módulos y la máquina virtual de JavaScript para que pueda correr en dispositivos, simuladores o emuladores Android e IOS para esto hace uso de sus SDK de cada plataforma para construir y compilar a un aplicativo nativo. (Branstein & Branstein, 2017)

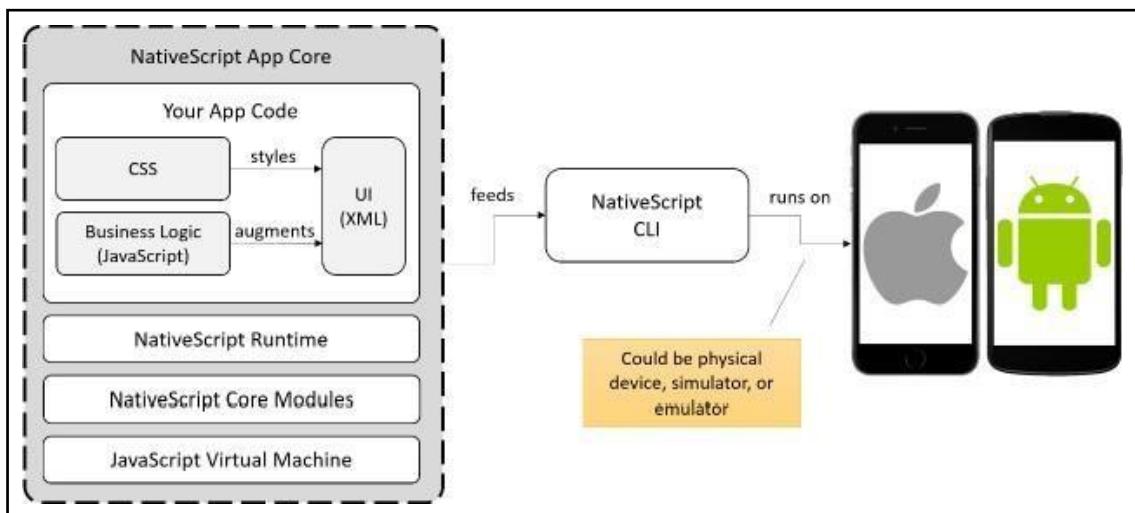


Figura 40. Arquitectura de NativeScript(Branstein & Branstein, 2017)

3.4.3 ReactNative

3.4.3.1 Definición

Fue creado y actualmente está siendo desarrollado por Facebook, está basado en el framework React, el cual es una de las librerías de Javascript más usadas según una encuesta realizada por StackOverflow en el año 2017. Permite a los desarrolladores crear aplicaciones con un código fuente en común para ambas plataformas del mercado actual: Android e IOS.(Alférez, 2018)

Su principal objetivo es la de crear aplicaciones con la experiencia nativa, es decir sus interfaces y la usabilidad sean las mismas que cuando se desarrollan APIs nativas.(Alférez, 2018)

3.4.3.2 React

Es una librería de Javascript que permite crear interfaces de usuario, contiene componentes programados de forma declarativa, manipula con APIs el DOM (modelo de objeto de documento) que tiene la estructura de un árbol de objetos que tiene una página HTML, se encarga de realizar cambios cuando su estado o propiedades cambian, actualizando la vista.(Alférez, 2018)

3.4.3.3 Estructura de componentes

- **Estados:** El estado del componente es la data que cambia durante el tiempo de ejecución e influencia su comportamiento. Por ejemplo, el campo del texto del usuario es parte de un estado.
- **Propiedades:** Son los atributos de los objetos con los que un pariente puede describir a un hijo. Por ejemplo, un componente que renderiza un hijo que contiene un texto definido en sus accesorios
- **Ciclo de vida:** Son las diferentes etapas que pasa un componente cambia su comportamiento, inicialmente empiezan con estado, propiedades y su incorporación en la estructura del árbol de interfaces de usuario. Cuando se cambian el estado o propiedades de los componentes éstos se redibujan así mismos.
- **Función de renderizado:** Esta función soporta la lógica de la apariencia de un componente basado en su estados y propiedades. React usa código en sintaxis XML y código Javascript para definir las interfaces de usuario como etiquetas HTML y demás componentes.

3.4.3.4 Flujo de datos unidireccional

El enlace de datos es el proceso que establece una conexión entre la interfaz gráfica del aplicativo y la lógica de negocio. React maneja las llamadas de un solo flujo de datos (véase Figura 41), los usuarios no pueden cambiar el estado directamente, se necesita

lanzar un evento en los componentes hijos para propagar a través de un árbol, a su vez los padres actualizarán la vista de arriba hacia abajo. (Alférez, 2018)

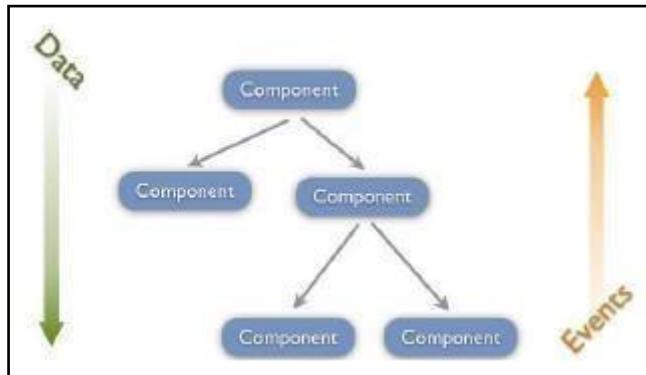


Figura 41. Flujo de datos en react(Alférez, 2018)

3.4.3.5 Reconciliación

React usa un Virtual DOM (véase Figura 42) ya que la directa manipulación de DOM es ineficiente ya que se consideró para páginas estáticas renderizadas por un servidor, ese virtual DOM está separado del servidor y se usa para detectar si ha habido cambios usando algoritmos de diferenciación, este proceso se denomina reconciliación. Cuando ocurran cambios en el estado se traducirá en operaciones del DOM que permitan re-renderizar el contenido, optimizar el proceso de actualizar la vista. En el caso de ReactNative el renderizado se realiza a dispositivos móviles.(Alférez, 2018)

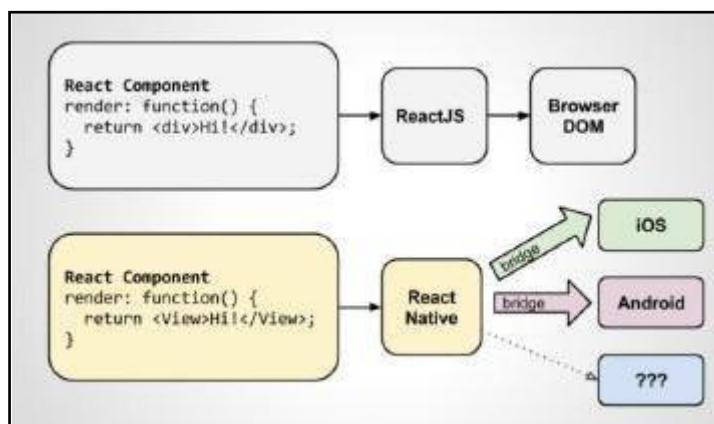


Figura 42.React y ReactNative Virtual DOM (Alférez, 2018)

3.4.3.6 Flux

React permite describir la interfaz de usuario como una función del estado de los componentes. Para manejar este estado se vuelve más complicado cuando se hace más grande el aplicativo dado que hay un estrecho vínculo entre interacciones de usuario y cambios de estado. Por eso se creó Flux basado en el patrón de diseño observador. Está compuesto de 3 partes: dispatcher (despachador), store (almacén), view (vista).(Alférez, 2018)

El Flujo de datos en Flux (véase Figura 43) funciona desde que el usuario genera acciones que se envían a través de un despachador central, luego los almacenes mantienen la lógica y el estado y son responsables de mantener el estado consistente de acuerdo a las acciones recibidas y emitir cambios con eventos y por último las vistas escuchan los eventos y se encargan de re-renderizar y propagar los datos a través del árbol obtenido del almacén. Una de las implementaciones más usadas es Redux bastante aceptada entre los desarrolladores. (Alférez, 2018)

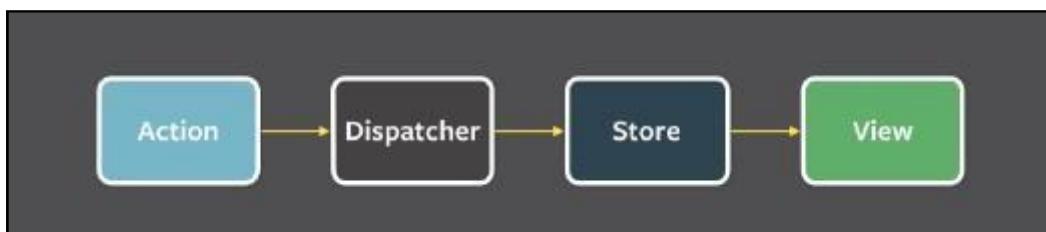


Figura 43.Flujo de datos en Flux(Alférez, 2018)

3.4.3.7 Partes internas

ReactNative provee al desarrollador 2 herramientas: los componentes de interfaces para aplicaciones móviles como listas o escogedor de fechas y las APIS que permiten acceder a funcionalidades de los sistemas operativos como gestos o almacenamiento local.Su ejecución está basada en dos ambientes: el lado nativo (Java en Android y Objective-c/Swift en IOS), la cual mantiene la responsabilidad de renderizar y manipular las vistas, y el lado JavaScript que describe la lógica y la interfaz de usuario.(Alférez, 2018)

Una vez que el aplicativo corre se empiezan procesos paralelos como la carga del paquete JavaScript que es creada por el código y sus dependencias, la carga de los módulos nativos, y la inicialización de una nueva instancia de la máquina virtual de JavaScript.(Alférez, 2018)

La configuración de JSON es el ambiente de ejecución que contiene un arreglo de los módulos necesitados, entonces cuando el método del módulo sea llamado JavaScript creará los objetos en tiempo de ejecución llamará el respectivo código nativo a través del bridge(puente) el cual transformará la llamada JavaScript a llamadas específicas de las plataformas. Luego el código JavaScript será ejecutado en una máquina virtual, el cual cargará las funciones scripts y parámetros de configuración del layout, además se harán usos de eventos que enviarán al módulo nativo que despachará éstos, el proceso JavaScript será llamado y éste se repetirá. (véase Figura 44)(Alférez, 2018)

De acuerdo con (Alférez, 2018)los principales hilos de los que está compuesto son los siguientes:

- Hilo principal: Es el libro por defecto y trata con la interfaz nativa de usuario.
- Sombra de la cola: Representa las propiedades de la computación del layout
- En IOS cada módulo nativo tiene su propio hilo en cambio en Android comparten un grupo de hilos

- Hilo de JavaScript corre el código Javascript en la máquina virtual de JavaScript.

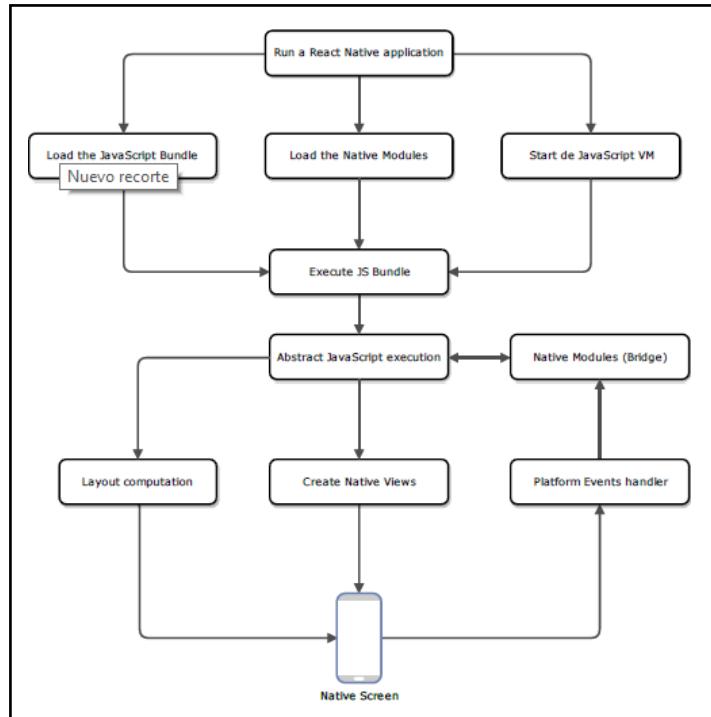


Figura 44. Esquema de proceso de ejecución en reactnative(Alférez, 2018)

3.4.3.8 De código a renderizado nativo

El código JavaScript se comunica con el lado nativo a través de un puente(bridge) por medio de un mensaje con un sistema de paso de mensajes con un componente llamado Administrador UI. Mensajes son manejados por la cola de mensajes que procesan los métodos asíncronos en una ejecución de un modelo de un solo hilo.(Alférez, 2018)

De acuerdo con(Alférez, 2018) los pasos que ocurren cuando hay eventos o cambios en la aplicación son:

- El código corre a través de React, que es el proceso de reconciliación es llevado a cabo resultando en una diferencia entre el actual estado y el estado anterior de la aplicación.
- La diferencia es modelada como comandos imperativos del Administrador UI (mensajes).
- Estos mensajes son enviados para construir un árbol de sombra iterativo, el cual es la representación de la interfaz gráfica que es mutable en el lado nativo. (como el DOM para web). Para ese fin la interfaz de administrador de vistas el cual existe para cada tipo primitivo de ReactNative, traduce comandos del Administrador UI a los nodos sombras. Por ejemplo: una vista de un texto de ingreso de texto. Si el desarrollador tiene una vista personalizada en código nativo, un administrador de vista debe ser implementado para enlazar de JavaScript al componente de vista nativo en la plataforma.

- Yoga, un motor de diseño que implementa un subconjunto de css optimizado para plataformas móviles, es responsable de calcular propiedades precisas de las vistas en las pantallas del dispositivo. Para cada nodo de sombra se crea un nodo Yoga que permite comunicar con el código nativo de la plataforma específica y crea la interfaz de usuario en el dispositivo.

3.4.3.9 Puentejar la comunicación entre javascript y código nativo

La ejecución de los hilos en ReactNative está dividido en dos: unos corren en el lado nativo (principal, colas de sombras e hilos de módulo nativo) y el hilo de JavaScript corre en la máquina virtual de JavaScript, la comunicación de ambos sitios es denominado bridge(puente). (véase Figura 45)(Alférez, 2018)

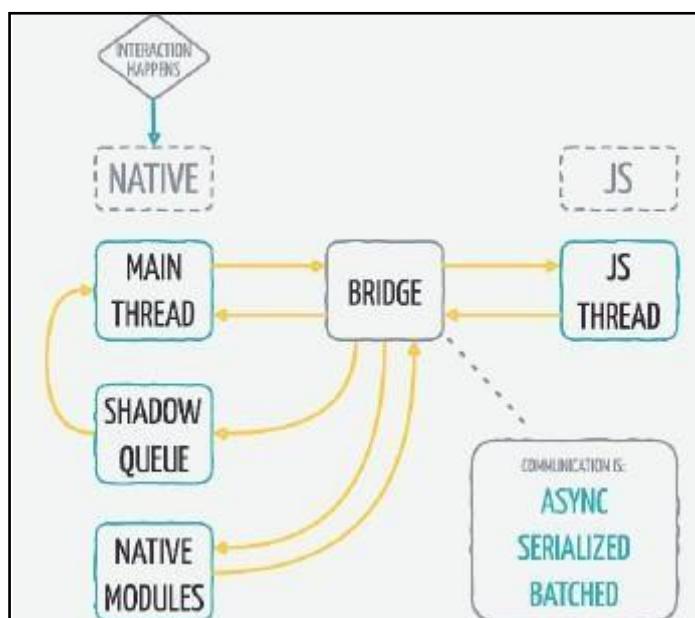


Figura 45.Puentejar en ReactNative(Alférez, 2018)

Se usan para la comunicación a través de interruptores según su contexto, el proceso es involucrado cuando un hilo almacena su estado para que su ejecución se pueda reanudar más tarde. Esto permite que ReactNative pueda usar widgets nativos, para esto se renderiza de acuerdo con la interfaz de usuario de las plataformas específicas de componentes IOS y Android.(Alférez, 2018)

Puenteando (Bridging) es un sistema basado en paso de mensajes tiene comunicación de tipo asíncrona, los mensajes son serializados y actualizados en batch para aumentar su rendimiento. Cada interacción de usuario, creación de vistas o peticiones de red pasan a través de este puente. (véaseFigura 46)(Alférez, 2018)

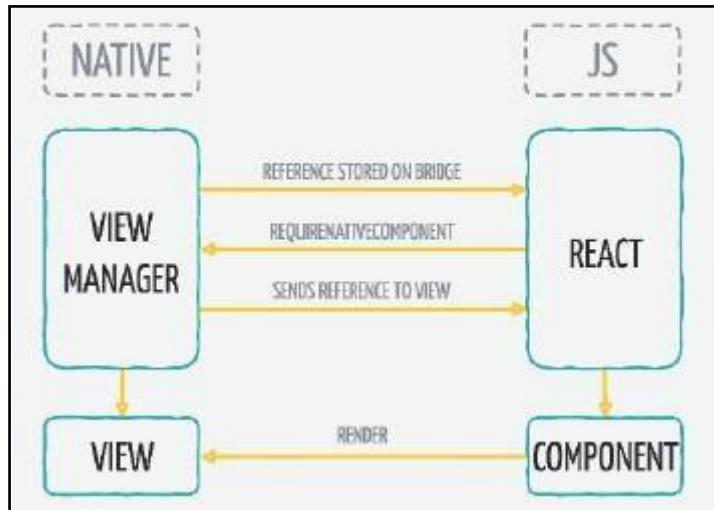


Figura 46. Componente de interfaz nativa de usuario en ReactNative(Alférez, 2018)

Los módulos nativos (véase Figura 47) permiten acceder a la plataforma subyacente que es el hecho de escribir código nativo y acceder con JavaScript, solo exporta métodos y constantes más no componentes de interfaz de usuario.(Alférez, 2018)

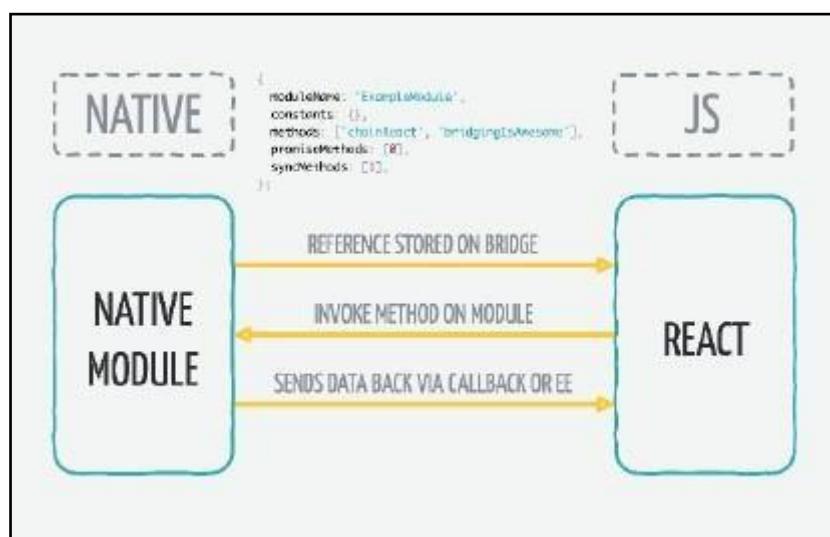


Figura 47. Componente de interfaz de módulo nativo en ReactNative(Alférez, 2018)

De acuerdo con(Alférez, 2018)puentar es muy importante en ReactNative, un proyecto puede necesitar un puente personalizado en el caso que:

- Exista una integración de SDK de terceros requeridos.
- Alto rendimiento es obligatorio.
- ReactNative se integra dentro de un proyecto existente o con módulos personalizados existentes.
- Acceso a plataforma API (hardware del dispositivo) es requerido.

Ambas plataformas son rápidas computacionalmente en su propio ambiente, pero el cuello de botella en el rendimiento ocurre cuando se usa el puente, por eso es por lo que

react crea un DOM virtual para optimizar la cantidad de información que pasa a través del puente.(Alférez, 2018)

3.4.3.10 Ambiente de desarrollo

ReactNative tiene algunas dependencias y configuraciones iniciales antes de comenzar a desarrollar. Por otra parte, Expo es un framework desarrollado por la compañía del mismo nombre está construido sobre ReactNative y provee de componentes con interfaces de usuario, servicios y librerías para acceder a las funcionalidades del dispositivo (cámara, GPS) con código multiplataforma.(Alférez, 2018)

De acuerdo con(Alférez, 2018) para codificar en principio se puede usar cualquier editor no es necesario un IDE. Existen 3 tipos de ambientes con respecto a cómo se organizan los archivos de código y si se utilizan o no:

- Expo Project: Se configura con la interfaz de la línea de comandos expo y establece el ambiente para que esté listo para codificar, fundamentalmente con la librería expo. Este framework mantiene la responsabilidad de compilar el código JavaScript y servirlo.
- ReactNative con ExpoKit: Crea un aplicativo reactnative por un script provisto por Facebook que establece sus principios, pero que también está basado en Expo. El responsable de la compilación es React, Expo se puede considerar como una librería adicional.
- ReactNative sin ExpoKit: Es creado por la interfaz de línea de comandos nativos de react y crea un proyecto reactnative sin alguna herramienta adicional. A diferencia de los otros tiene la opción de configurar en cada ambiente y usar IDES como Android Studio, Xcode en Android e IOS respectivamente.

Desarrollo de la aplicación

Un aplicativo en reactnative contiene 3 carpetas: Nodemodules (librerías JavaScript), Android e IOS, los proyectos que pueden ser aplicaciones comunes nativas pueden ser compiladas y desplegadas con herramientas nativas de cada plataforma. Está compuesto de Package.json y App.json el primero contiene las dependencias y versiones, el segundo la raíz de la jerarquía de las interfaces de usuario, el pariente de todos los componentes.(Alférez, 2018).

Reactnative como se mencionó corre en dos formas en la plataforma nativa y el código JavaScript. En la etapa de desarrollo el segundo es ejecutado en la computadora del desarrollador y se ejecuta en tiempo real para el primero. El proyecto esta preconfigurado y puede ser automáticamente recargado es decir si hay un cambio, el ambiente hará ese cambio en la lógica de Javascript entonces no es necesario compilar de nuevo, esto incrementa el tiempo de desarrollo.(Alférez, 2018).

Los componentes de módulos nativos pueden ser compartidos por la comunidad, es decir reactnative toma ventaja de los componentes o librería publicados por otro desarrollador.ReactNative no usa etiquetas HTML, pero si componentes provistos por reactnative o definidos por el desarrollador. Los estilos son definidos en un objeto

JavaScript que siguen la convención de CSS en la web y el algoritmo flexbox.(Alférez, 2018).

Depuración

De acuerdo con(Alférez, 2018)reactnative ofrece algunas funcionalidades que facilita e incrementa el proceso de depuración:

- Recarga automática (Recarga en vivo y recarga en caliente)
- La inspección de componentes react de jerarquía y parámetros.
- Monitor de rendimiento

Además, (Alférez, 2018) menciona que la aplicación puede ser probada en dispositivos reales y/o dispositivos simulados:

- En dispositivos reales se requiere de unaconfiguración especialsegún el ambiente ya sea expo o reactnative, además en caso de ser un dispositivo IOS unaPC MAC es requerida.
- En dispositivos simulados, los proyectos están simulados en un ambiente de reactnative se compila con un dispositivo virtual con el cual cada IDE debe estar configurado.

Diferentes herramientas se necesitan para depurar el ca los proyectos de reactnative que son 3(Javascript, Android, iOS) como puntos de interrupción, variables de inspección, etc.Para el código nativo se requiere IDES para depurar como Android Studio o Xcode, por otro lado, la lógica de JavaScript puede ser analizada por herramientas de reactnative, Google Chrome como depuración remota, Nuclide un plugin de Atom.(Alférez, 2018).

Pruebas

De acuerdo con(Alférez, 2018), reactnative tiene una librería denominada Jest que es usada por Facebook para probar componentes en ambos aplicativos react o reactnative, además pueden ser usadas para probar un frameworkJavascript. Permite al desarrollador:

- Afirmar valores
- Pruebas instantáneas: un estado de referencia de un componente es almacenado y luego su estado actual y la referencia son comparados a través de la ejecución de la aplicación.
- Pruebas de funcionalidad asíncronas.
- Estados Mock y eventos simulados como toques.

Además(Alférez, 2018) menciona quereactnative también incluye pruebas de integración según la plataforma (dispositivos Android e IOS) y pruebas fin-a-fin son llevadas a cabos para obtener productos de buena calidad.

3.4.4 Flutter

3.4.4.1 Definición

Según (Flutter Official Documentation, 2018), Flutter es un SDK de aplicativo móvil para construir aplicativos con alto rendimiento y alta fidelidad para IOS y Android desde un mismo código fuente que permita a desarrolladores entregar aplicativos con buen rendimiento que se sientan naturales en distintas plataformas. Está escrito en Dart, y corre en la máquina virtual de Dart en cual trabaja directamente con el código fuente no con código en bytes, soporta compilación JIT durante el desarrollo que permite cambios en caliente, y AOT para producción ya que brinda inicio rápido y rendimiento esperado de una aplicación. (Alférez, 2018)

En la imagen (véaseFigura 48), se muestra la arquitectura del sistema Flutter en la cual la parte en verde son capas que pueden ser personalizadas por el desarrollador para conseguir una funcionalidad específica a continuación se definirán sus componentes de acuerdo con(Alférez, 2018):

- El motor(engine) es el tiempo de ejecución que está construido con código C++. Usa Skia como librería de renderizado de gráficos 2D y el API Core Dart, expone una librería dart:ui que permite precisar las coordenadas de cada elemento del canvas.
- La capa de renderizado es una capa de abstracción que permite construir la vista como un árbol de objetos que se actualiza cuando un cambio ocurre, consiste en 3 pasos: el cálculo de diseño que posiciona los elementos, pintura que permite definir como se verán los objetos y componerlos que junta los elementos en un solo dibujo para que aparezcan en la pantalla como una sola cosa.
- La capa material/cupertino usa la capa de composición de los widgets para implementar controles definidos que siguen las líneas de diseño de la plataforma específica.
- Los widgets son muy importantes en Flutter son semejantes a los componentes de React, La capa de widgets soluciona el problema de tener que construir y mantener el árbol, ya que en vez de crear cambiar al árbol crea un árbol de widgets inmutables cada vez que se actualiza. Cuando se crea un árbol paralelo de elementos se crea una referencia a sus actuales widgets y sus objetos de renderizados correspondientes.



Figura 48. Arquitectura del sistema Flutter(Alférez, 2018)

3.4.4.2 Plataforma específica de código

Flutter adopta el concepto de plugins que permiten acceder a las características del hardware de un dispositivo. Estos se comunican con la plataforma a través de canales que manejan comunicación con el paso asíncrona de mensajes. El desarrollador puede escribir código nativo y exponerlo en un API basado en mensajería. (Alférez, 2018)

3.4.4.3 Ambiente de desarrollo

Flutter posee su propio SDK que funciona con una interfaz de línea de comandos, requiere de Android SDK e IOS SDK para compilar en sus plataformas. Cualquier editor de texto se puede usar para escribir código además de integrarse en IDEs específicos (Android Studio y Xcode). Ofrece también la opción de cambios en caliente que permiten recargar los cambios sin reiniciar o perder el estado del aplicativo. Está compuesto de cuatro carpetas: código Android, código IOS, librerías, pruebas. (Alférez, 2018)

3.4.4.4 Interfaz de usuario

Las interfaces de usuario no están compuestas de widgets nativos de la plataforma, no necesita un puente para comunicar con la plataforma (véase Figura 49), pero si construye y calcula las vistas en su contexto y renderizar en un canvas nativo, esto hace que las interfaces de usuario sean coherentes entre las plataformas y diferentes versiones. El desarrollador puede personalizar interfaces de usuario que se sincronizan en todas las plataformas. (Alférez, 2018)

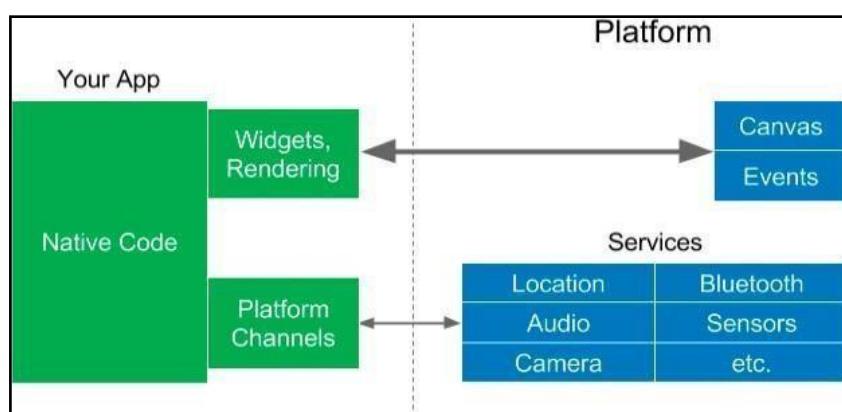


Figura 49. Arquitectura de aplicativos móviles usando Flutter(Alférez, 2018)

3.4.4.5 Desarrollo de aplicativo

Flutter no cuenta con un lenguaje declarativo marcado para describir la interfaz. Dart es usado para definir el diseño sin dividir el código entonces se hace una combinación de lógica e interfaz de usuario en el mismo código, esto promueve el cambio en caliente. Los widgets pueden mantener su propio estado para manejar interacciones de usuario, estos son inmutables y manejan los estados mutables una clase de estado separando al estado del widget de su apariencia. (Alférez, 2018)

3.4.4.6 Depuración y pruebas

Flutter CLI provee una herramienta lint(analizador Dart), y una herramienta de depuración y perfil(observatorio Dart). Ambas corren por defecto cuando se ejecuta la aplicación, cada capa del framework Flutter provee una función para volcar la información en la consola, permitiendo al desarrollador llevar un análisis profundo de la ejecución. (Alférez, 2018)

Además, provee módulos y herramientas para la prueba unitaria y la prueba de los widgets, que pueden correr en la máquina virtual Dart. Además, se pueden llevar acabo las pruebas de integración con Flutter Driver, que permite conectar a un aplicativo corriendo en un dispositivo real o simulador, emite comandos y ejecuta scripts de prueba en la computadora del desarrollador. (Alférez, 2018)

3.4.5 Xamarin

3.4.5.1 Definición

Es un framework para el desarrollo de aplicativos móviles multiplataforma comprado y actualmente desarrollado por Microsoft. Se dirige a las plataformas Android, IOS y Windows está basado en C#, el cual es un lenguaje desarrollado por la propia compañía su ejecución está basada en el framework .NET. Xamarin provee ecosistema de plugins para acceder a las funciones del dispositivo específico. (Alférez, 2018)

.Net framework es usado solo en el ambiente de ejecución de Windows, Xamarin por otro lado usa Mono, una implementación de código abierto de .NET que provee compatibilidad en otras plataformas, esto permite que desarrolladores c# puedan escribir código multiplataforma dirigidas a Windows, macOS, Linux, Android e iOS. (véaseFigura 50) (Alférez, 2018)

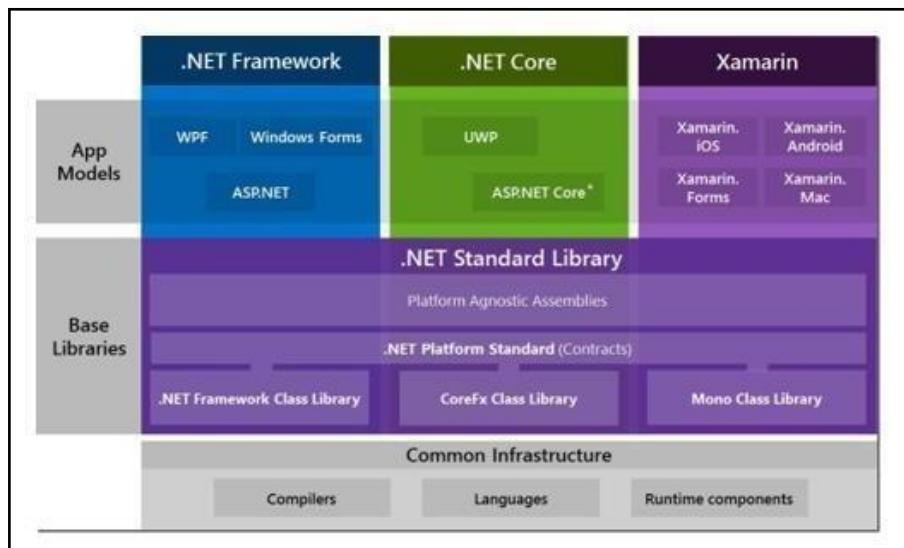


Figura 50. Ecosistema .NET (Alférez, 2018)

De acuerdo con(Alférez, 2018) el código C# es compilado en una representación intermedia denominada Microsoft IntermediateLanguage (MSIL), que no depende del

sistema operativo o el procesador elegido. Hay dos tipos de traducción a código máquina que se usan de acuerdo con Microsoft:

- Just in Time (JIT): Compilado en memoria en la ejecución como cuando corre
- Aheadof Time (AOT): Compilación ejecutada antes de la ejecución y almacenada dentro del paquete de aplicaciones.

Hay dos tipos de códigos que juegan un rol: Código Manejado (MSIL) llevado a cabo por el tiempo de ejecución mono y el código nativo que corre en plataforma específica (tiempo de ejecución de objective-c o Android). (Alférez, 2018)

En IOS se requiere una computadora MAC con macOS como sistema operativo para compilar los aplicativos, existe además una restricción que impide generar código de tiempo de ejecución en un dispositivo, se usa la compilación AOT para producir código binario nativo IOS y las clases .NET usadas. (véase Figura 51) (Alférez, 2018)

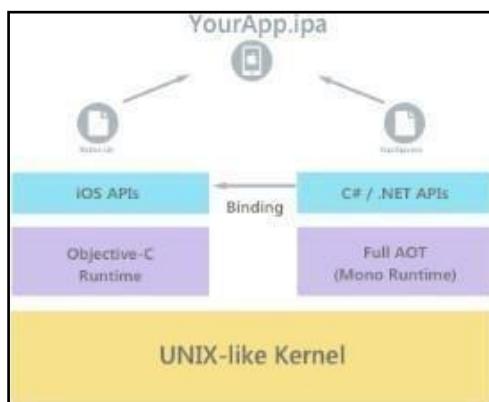


Figura 51. Enlace IOS (Alférez, 2018)

Para Android se usa el ambiente Mono que corre lado a lado con el programa en tiempo de ejecución de Android. Ambos permiten exponer APIs y comunicarse con Managed Callable Wrappers (permiten manejar el código y llamar funciones Android) y Android Callable Wrappers (es un puente JNI que permite el programa en tiempo de ejecución Android pueda llamar al código manejado). (véase Figura 52)

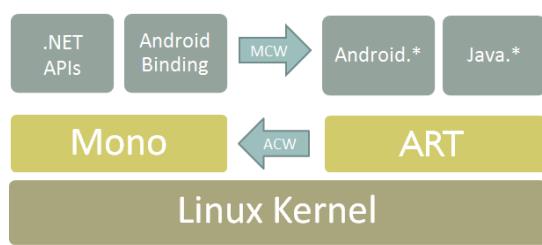


Figura 52. Enlace Android (Alférez, 2018)

Cuando se inicia el desarrollo de una aplicación móvil, Xamarin crea diferentes proyectos: uno para el código compartido (sharedcode) y otro para la plataforma objetivo (Xamarin.Android, Xamarin.IOS y UWP). Hay 2 enfoques para el nivel del código compartido: Xamarin y Xamarin.Forms (véase Figura 53). (Alférez, 2018)

El primer enfoque se enfoca en compartir el ambiente (lenguaje de programación, aplicación en tiempo de ejecución y librerías) y la lógica de negocio entre las plataformas, pero cada interfaz de usuario debe ser desarrollada por cada plataforma. (Alférez, 2018)

Por otra parte, el segundo enfoque añade una capa de abstracción que permite construir UIs nativas para cada plataforma compartiendo solo un código base. (Alférez, 2018)

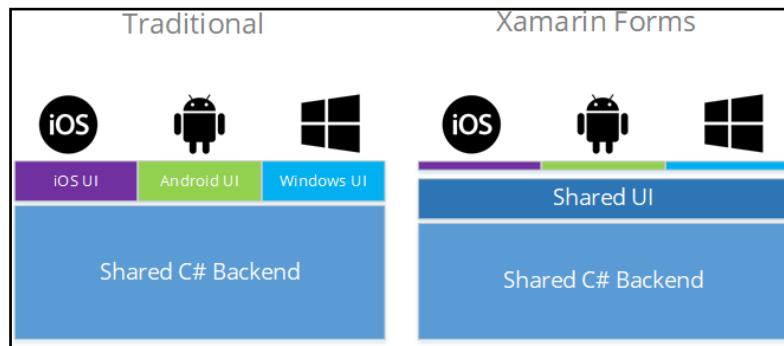


Figura 53. Pilas de Xamarin y Xamarin.Forms(Alférez, 2018)

De acuerdo con(Alférez, 2018) el enfoque Xamarin.Forms tiene las siguientes características:

- Interfaz de usuario son desarrolladas en un solo un código fuente o junto con uso de la XAML (lenguaje de marcado descriptivo), con esto diferencia la lógica de la interfaz gráfica. Las páginas aquí son las principales unidades que corresponden a Activities en Android y View Controllers en IOS.
- XAML usa la arquitectura MVVM(Modelo-Vista-VistaModelo) (véase Figura 54). El enlace de datos puede ser definidos en código o archivos XAML vinculando propiedades source y target. Hay 4 tipos: default, twoWay, oneWay, oneWayToSource.

De acuerdo a la imagen(véase Figura 54), se muestra que la vista es responsable de manejar como se muestra la pantalla, el Vista-Modelo maneja la lógica del enlace de datos de las vistas y notifica de sus cambios, y el modelo que mantiene los datos de la aplicación, lógica de validación y de negocio y el modelo de datos.

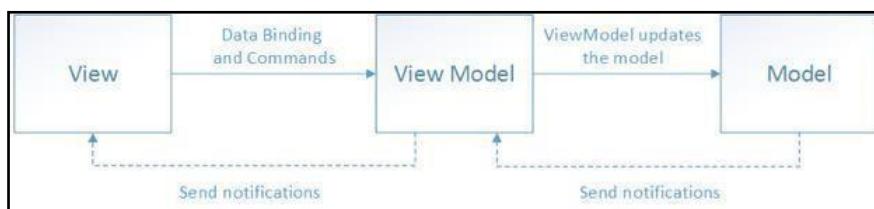


Figura 54. Patrón de Modelo-Vista-ModeloVista(Alférez, 2018)

Se permite que el desarrollador pueda llamar funcionalidades específicas de la plataforma al definir interfaces y proveer de implementación de cada plataforma específica (véase Figura 55). (Alférez, 2018)

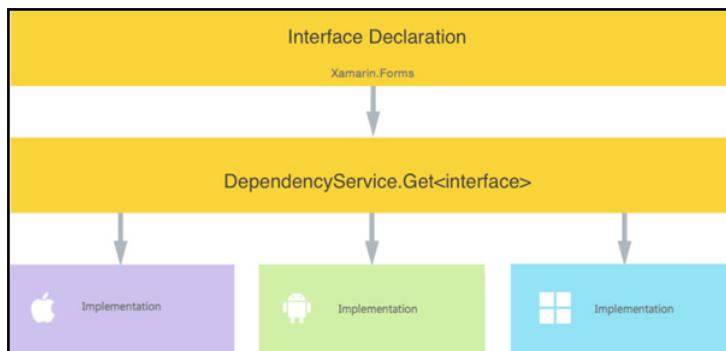


Figura 55. Dependencia de servicios Xamarin(Alférez, 2018)

3.4.5.2 Ambiente de desarrollo

Xamarin está integrado en el IDE Microsoft's Integrated Development Environment de Visual Studio. Permite depurar y compilar la solución además incluye herramientas para mejorar la creación de vistas nativas a cada plataforma. Para probar en un dispositivo Apple se necesita una MAC para compilar el código. (Alférez, 2018)

Cada vez que hay un cambio se debe compilar todo el proyecto de nuevo. Microsoft ofrece Xamarin Live Player que provee la capacidad de correr y recargar en vivo, aunque solo lo hace de interfaces de usuario. (Alférez, 2018)

3.4.5.3 Desarrollo de aplicación

De acuerdo con(Alférez, 2018) hay 2 formas que se pueden compartir el código: Proyectos compartidos y librerías estándar .NET:

- En el primero el código compartido del proyecto es compilado como parte de varios proyectos y puede incluir directivas para activar o desactivar partes de la lógica de cada plataforma.
- En el segundo el código es compartido en las diferentes plataformas como una librería basada en la librería estándar .NET.

Para usar las funcionalidades es del dispositivo como acceder a funciones de hardware disponibles se puede hacer o bien implementando la funcionalidad en cada plataforma o usar un plugin. Xamarin permite la adición de funcionalidad de la aplicación por medio de plugins los cuales son componentes o paquetes construidos por comunidades, estos no tienen soporte oficial. (Alférez, 2018)

3.4.5.4 Pruebas

De acuerdo con(Alférez, 2018), Xamarin ofrece las formas de pruebas:

- Pruebas unitarias en Xamarin.IOS
- Herramientas de pruebas de la lógica C# en .NET.
- Pruebas automatizadas gracias a Centro de aplicaciones de visual studio, además este centro permite la integración y entrega continua, construcción remota, establecer notificaciones y pruebas en dispositivos reales.
- Xamarinprofiler para analizar la aplicación y colecciónar data relevante al correr.

3.5 Arquitectura

3.5.1 Arquitectura de 3capas

3.5.1.1 Definición

La arquitectura predecesora de 2 capas solo teníalas capas de cliente y de servidor que hacíanque el cliente maneje la lógica de negocio mezclada con la interfaz de usuario haciendo que sea difícil la portabilidad, la solución fuecrearnuevas capascon la finalidad de separar la lógica de la aplicación de la interfaz de usuario y del mecanismo utilizado para el almacenamiento de datos (arquitectura de 3 capas). La acción de separar estas nuevas capas permite reducir el grado de acoplamiento que hay entre unidades o componentes, entre los cuales existe un alto grado de dependencia.(Ganchoso & Vera, 2015)

3.5.1.2 Capas

A continuación, se presenta la interacción de estas capas en este tipo de modelo de arquitectura(véaseFigura 56):

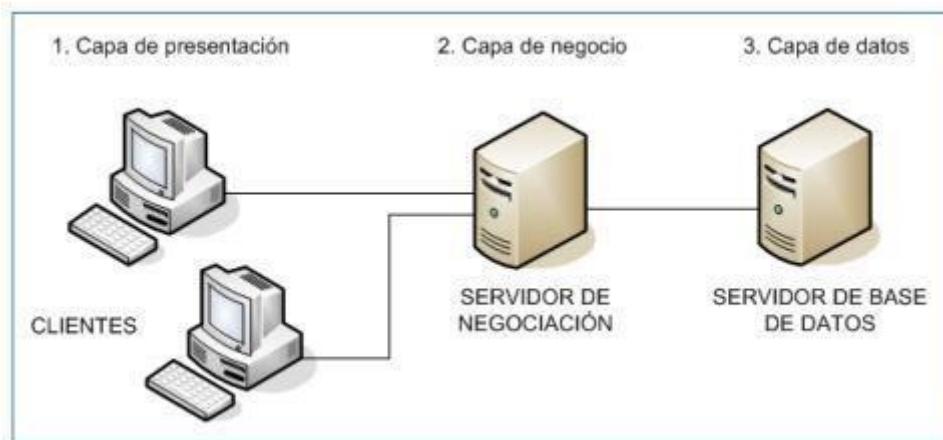


Figura 56. Arquitectura de 3 capas (Sarasty, 2015)

De acuerdo con(Ganchoso & Vera, 2015), este modelo presenta las siguientes capas:

La capa de presentación, encargada de interactuar con el usuario de la aplicación mediante una interfaz de usuario (ya sea una interfaz web, una interfaz Windows o una interfaz en línea de comandos, aunque esto último suele ser menos habitual en la actualidad), dentro del cual presenta y captura la información del usuario con un mínimo de proceso.

La capa lógica de la aplicaciones también considerada como la capa de negocio, debido a que es aquí donde se establecen todas las reglas que para las cuales se diseña el sistema. Esta capa es la encargada de recibir las solicitudes enviadas por la capa de presentación, además de interactuar con la capa de datos para solicitar al repositorio de datos el almacenamiento o recuperación de datos de ésta.

La capa de acceso a los datoses también nombrada la capa de persistencia, esta capa es la encargada de gestionar el almacenamiento de los datos, así como la recuperación de

los datos solicitados por la capa del negocio. Esta capa interactúa solo y directamente con la capa de negocio.

35.13 Es una arquitectura monolítica

De acuerdo a (Kalske, Mäkitalo, & Mikkonen, 2018), es la más común arquitectura monolítica que se usa, es más fácil de empezar que arquitecturas complejas como microservicios, pero a medida que crece el código fuente se hace mucho más difícil de realizar cambios dado que afecta a muchas partes del software, además de introducir alto acoplamiento y reducirlos puede ser complejo, generalmente cuenta con solo una base de datos y un solo lenguaje de programación.

35.14 Ventajas

De acuerdo con(Sarasty, 2015), algunas ventajas que se encuentran son las siguientes:

- La estructura de datos se puede modificar sin afectar a la interfaz de usuario.
- El código de la capa intermedia se puede reutilizar por varias aplicaciones, si fue desarrollado o diseñado modularmente.
- Es más sencillo realizar modificaciones sobre una capa sin afectar los módulos restantes, lo cual facilita el desarrollo, mantenimiento y uso.

35.15 Desventajas

De acuerdo con(Sarasty, 2015), entre algunas de las desventajas que se encuentran son las siguientes:

- Incremento en el tráfico de red, por lo que se requerirá un balanceo de carga.
- El cliente puede que no cuente con los recursos del servidor por falta o problemas de conexión.

3.5.2 Arquitectura orientada a servicios (SOA)

3521 Definición

SOA es un modelo de arquitectura que posiciona a los servicios como un mecanismo esencial para poder incrementar la eficiencia, agilidad y productividad de una empresa. La implementación de SOA puede consistir en la combinación de tecnologías, productos, APIs, extensiones para el soporte de una infraestructura, y otras partes.(Erl, 2019)

Los servicios vienen a ser los componentes unitarios de la arquitectura SOA, son aplicaciones que residen en servidores centralizados, éstos intercambian datos entre sistemas independientemente del lenguaje de programación en el que estén desarrolladas y de la plataforma en dónde se ejecuten.(Erl, 2019)

3522 Elementos de SOA

En la siguiente imagen (véase Figura 57), se muestran los elementos que componen la arquitectura SOA.

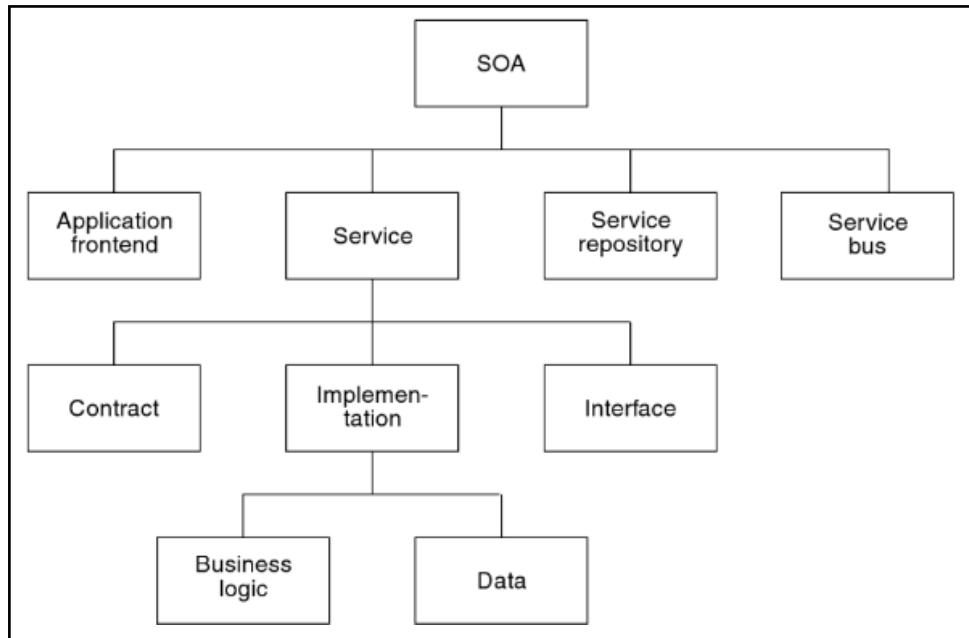


Figura 57. Elementos de SOA(Krafzig, Banke, & Slama, 2004)

3.5.2.2.1 AplicaciónFrontend

Es uno de los principales artefactos que tiene SOA, son los participantes activos. Ellos inician y controlan toda la actividad de los sistemas empresariales. Hay diferentes tipos de aplicaciones frontend. Una aplicación frontend con una interfaz de usuario gráfico, como una aplicación web o un cliente enriquecido que interactúa directamente con los usuarios finales, es el ejemplo más claro. Sin embargo, es enteramente posible que una aplicación frontend delegue mucho de su responsabilidad para un proceso de negocio a uno o más servicios. Últimamente, sin embargo, siempre una aplicación frontend inicia un proceso de negocio y recibe los resultados.(Krafzig, Banke, & Slama, 2004, págs. 58-65)

3.5.2.2.2 Servicios

De acuerdo con(Krafzig, Banke, & Slama, 2004, págs. 58-65), un servicio es un componente de software de significado funcional distinto que típicamente encapsula un concepto de negocio de alto nivel.

Tal como se muestra en la imagen (véase Figura 58), un servicio consiste en varias partes:

- **Contrato:** El contrato del servicio provee una especificación informal del propósito, funcionalidad, restricciones, y uso del servicio. La forma de esta especificación puede variar, dependiendo del tipo de servicio. Un elemento no obligatorio del contrato del servicio es una formal definición de interface basada en lenguajes como IDL o WSDL. A pesar de que no es obligatorio, una definición formal de interfaz de servicio agrega un beneficio significativo: Provee la abstracción e independencia de la tecnología, incluyendo el lenguaje de programación, middleware, protocolo de red y entorno de ejecución. Sin embargo, es importante entender que el contrato de servicio provee más

información que una especificación formal. El contrato puede imponer detalles semánticos de la funcionalidad y parámetros que no están sujetos a IDL o WSDL especificaciones.(Erl, 2019)

- **Interface:** La funcionalidad del servicio es expuesto por la interfaz del servicio a los clientes que son conectados al servicio usando una red. Aunque la descripción de la interface es parte del contrato del servicio, la implementación física de la interface consiste en “stubsservice” (trozo de código usado como sustituto de alguna otra funcionalidad, interfaces de servicios), que son incorporados dentro de los clientes de un servicio.
- **Implementación:** La implementación del servicio físicamente provee la lógica del negocio requerido y los datos apropiados. Es la realización técnica que satisface el contrato del servicio. La implementación del servicio consiste en uno o más artefactos tales como programas, datos de configuración y bases de datos.
- **Lógica de negocio:** La lógica del negocio que es encapsulado por un servicio es parte de su implementación. Se pone a disposición a través de las interfaces de servicio. Sin embargo, programar en contra de las interfaces es indeseable, si o ni uno aplica un enfoque orientado a servicios.
- **Data:** Un servicio puede también incluir datos. En particular, es el propósito de un servicio centrada en los datos.

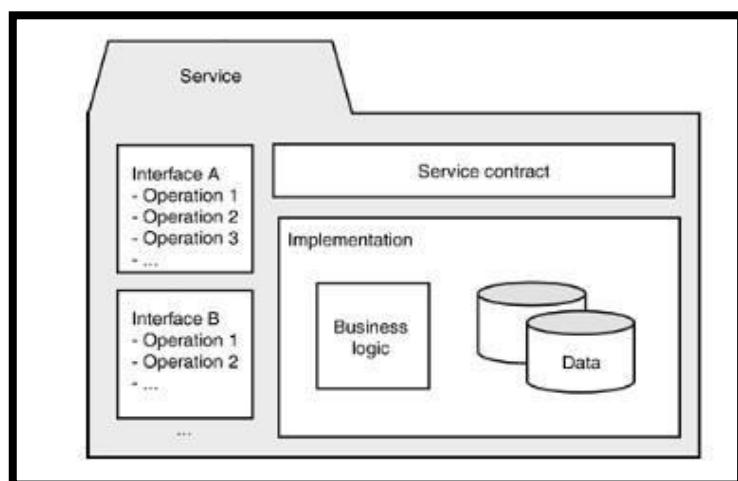


Figura 58. Componentes de un servicio en SOA(Krafzig, Banke, & Slama, 2004)

3.5.2.2.3 Repitorio de servicios

Un repositorio de servicios provee facilidades para descubrir servicios y adquirir toda la información para usar los servicios, particularmente si esos servicios deben ser descubiertos fuera del alcance funcional y temporal del proyecto que los creo. Aunque mucha de la información requerida es ya parte del contrato del servicio, el repositorio de servicio puede proveer información adicional, como una locación física, información sobre el proveedor, personas de contacto, las tarifas de uso, restricciones técnicas, problemas de seguridad y niveles de servicio disponibles.(Krafzig, Banke, & Slama, 2004, págs. 58-65)

Un repositorio de servicios puede ser arbitrariamente simple, en un extremo, la tecnología no puede ser requerida. Un lote de impreso de contratos de servicio localizado en una oficina y accesible por todos los proyectos es ya un repositorio de servicios valido. Sin embargo, existen mejores maneras de proporcionar esta información al tiempo que conserva la simplicidad del repositorio.(Krafzig, Banke, & Slama, 2004, págs. 58-65)

En algunos casos, las compañías han desarrollado sus propias herramientas que automáticamente generan la descripción de los servicios de las definiciones formales de los servicios. Esta es una particularidad muy útil si la definición formal del servicio esta anotado con información adicional sobre el servicio. Notar que esta información es típicamente muy diferente de la metainformación proveída por APIs de bajo nivel como clases Java. Esto es debido a los diferentes roles que las definiciones de los servicios juegan en SOA. Los servicios están típicamente no relacionados con códigos de librerías, pero están ligados a la ejecución. (Krafzig, Banke, & Slama, 2004, págs. 58-65)

Es importante distinguir entre tiempo de desarrollo y tiempo de ejecución de servicios “vinculados”. Vincular se refiere a la manera en que las definiciones del servicio y las instancias del servicio son localizadas, incorporadas dentro de la aplicación cliente, y finalmente conectados al nivel de la red.(Krafzig, Banke, & Slama, 2004, págs. 58-65)

- **Tiempo de desarrollo vinculación:** Si los servicios son descubiertos y vinculados en tiempo de desarrollo, las firmas de las operaciones del servicio son conocidas de antemano, tanto como el protocolo de servicio y la locación física del servicio (o al menos el exacto nombre del servicio en un directorio de servicio). Sin embargo, el tiempo de desarrollo de vinculación es un simple modelo, es suficiente para más propuestas. Permite a los proyectos identificar funcionalidades que han sido creadas por antiguos proyectos y reusar esos servicios.
- **Tiempo de ejecución vinculación:** Tiempo de ejecución es aún más complejo que el tiempo de desarrollo de vinculación. El tiempo de ejecución de vinculación va más allá de la complejidad de la búsqueda dinámica del servicio por propiedades con interfaces de servicios predefinidos es muy raro y es limitado para muy pocos dominios de aplicaciones. Un raro ejemplo para un dominio que realmente requiera servicios dinámica y altamente vinculados está en el mundo inalámbrico

3.5.2.2.4 Bus de servicios

Un bus de servicio conecta a todos los servicios SOA y aplicaciones frontend. Si dos participantes necesitan comunicarse, por ejemplo, si una aplicación frontend necesita invocar alguna funcionalidad de un servicio básico, el bus de servicio hace que suceda. El bus de servicio es similar al concepto de un bus de software tal como se define en el contexto de CORBA. Sin embargo, las diferencias significativas existen entre estos conceptos. La más importante, el bus de servicios no están necesariamente

compuesto de una simple tecnología, sino que más bien comprende una variedad de productos y conceptos.(Krafzig, Banke, & Slama, 2004, págs. 58-65)

De acuerdo con(Krafzig, Banke, & Slama, 2004, págs. 58-65), se mencionan las características más resaltantes del bus de servicio:

- **Conectividad:** El propósito principal del bus de servicio es interconectar los participantes de SOA. Provee facilidades que permiten a los participantes de una frontend aplicación SOA y servicios para invocar la funcionalidad de los servicios.
- **Heterogeneidad de la tecnología:** El bus de servicios debe abarcar una variedad de tecnologías diferentes. La realidad de las empresas es caracterizada por tecnologías heterogéneas. Consecuentemente, el bus de servicios debe ser capaz para conectar participantes que estén basadas en lenguajes de programas diferentes, sistemas operativos, entornos de ejecución. Además, usualmente encontrar una multitud de productos middleware y protocolos de comunicación en la empresa y todo esto debe ser soportado por el bus de servicio.
- **Heterogeneidad de los conceptos de comunicación:** Similarmente a la heterogeneidad de las tecnologías, el bus de servicio debe también abarcar una variedad de conceptos de comunicaciones diferentes. Debido a los requisitos divergentes de las diferentes aplicaciones, el bus de servicios debe permitir modos de comunicaciones diferentes. Obviamente, se debe al menos tener facilidades para comunicación de sincronización y desincronización.
- **Servicios técnicos:** Aunque el propósito del bus de servicios es principalmente comunicación, debe también proveer servicios técnicos, tales como logging, auditoria, seguridad, transformación de mensaje o transacciones.

3523 Principios de diseño de SOA

3.5.2.3.1 Estandarización de contratos de servicio

(Erl, 2019) menciona lo siguiente “*Servicios dentro del mismo inventario de servicios están en conformidad con las mismas normas de diseño del contrato*”.

Cuando un servicio es implementado como un servicio web, el contrato del servicio puede estar compuesto de un WSDL y múltiples esquemas XML y definiciones de políticas, así como documentos complementarios, por ejemplo, un SLA (véase Figura 59).(Erl, 2019)

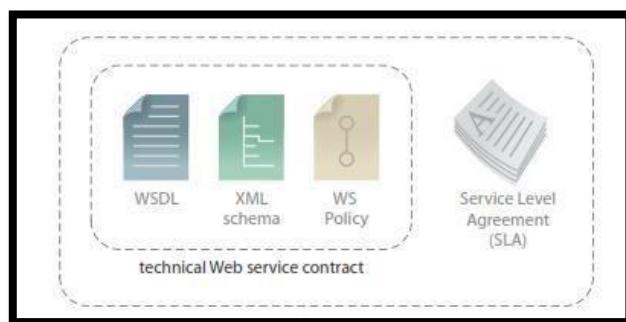


Figura 59.Estándar de contratación de servicios en SOA(Erl, 2019).

Este principio predica el enfoque de “Primer contrato” en la entrega del servicio, por el que se desarrollan a medida contratos (antes de la implementación de estos) de acuerdo con el diseño de las normas que se aplican a todos los servicios dentro de un determinado inventario de servicios.(Erl, 2019)

Las políticas y esquemas normalizadas pueden ser centralizadas por lo que una definición representa un conjunto oficial de afirmaciones políticas o tipos complejos que pueden ser referenciados por múltiples definiciones WSDL.(Erl, 2019)

Las normas de diseño de contrato pueden afectar y dar forma a muchas definiciones de elementos y la estructura general del WSDL, XML Schema, y políticas de definición de documentos.(Erl, 2019)

3.5.2.3.2 Acoplamiento de servicios

El acoplamiento refiere a la relación entre dos cosas, es comparable con el nivel de dependencia, se recomienda establecer una relación entre los límites de negocio del servicio tanto interior como exterior que reduzca o afloje las dependencias entre el contrato del servicio, su implementación y sus consumidores. El principio “Servicio bajo acoplamiento” menciona que el diseño debe ser independiente en el diseño y evolución de la lógica del servicio para lo cual marca una línea base para garantizar una confiable interoperabilidad.(Erl, 2019)

(Erl, 2019)menciona lo siguiente “*Los contratos de servicio imponen bajo consumo de acoplamiento de requisitos y ellos mismos desacoplados en su ambiente*”.

Un contrato de servicio que es derivado de su ambiente subyacente puede terminar formando tipos negativos de acoplamiento.(Erl, 2019)

El acoplamiento de la lógica-contrato es considerado una forma positiva de acoplamiento, dado a que representa la creación independiente de un contrato que es desacoplado del entorno del servicio.(Erl, 2019)

Los programas consumidores de servicios son requeridos para acoplarlos a un contrato de servicio. Como resultado, ellos heredan las formas negativas o acoplamientos positivos que residen dentro del contrato del servicio.(Erl, 2019)

Los diseñadores de consumidores de servicios no pueden estar conscientes del hecho de que el contrato de su programa este formando alguna dependencia con el acoplamiento negativo. Esto puede dirigir a varias formas de “indirecta” malintencionado acoplamiento. La proliferación del acoplamiento negativo es indeseable debido a que esto dirige a un frágil y recordado inflexible inventario de servicios de arquitecturas de integración pasadas.(Erl, 2019)

Este principio se relaciona con el patrón contrato centralizado que dicta que el contrato de servicios es el único medio de acceso de servicios, lógica y recursos.(Erl, 2019)

La proliferación del acoplamiento positivo es deseable ya que permite a implementaciones de servicios evolucionar si consumidores de servicios.(Erl, 2019)

3.5.2.3.3 Abstracción de servicios

(Erl, 2019) menciona lo siguiente: "Los contratos de servicios solo contienen información esencial y su información sobre servicios es limitada por lo que es publicado en contratos de servicios".

Cuando se determina que información de un servicio debe ser abstraído, es muy útil categorizar la meta data de los servicios en distintas categorías. La aplicación de este principio puede afectar la abstracción de cada uno de la metainformación de tipos diferentes.(Erl, 2019)

Este principio avoca el ocultamiento deliberado de la meta data del servicio para que una mínima cantidad de información de un servicio sea accesible al mundo exterior. Los procedimientos de control de acceso pueden llegar a ser un requisito necesario para ser abordado en un nivel de organización a través de la introducción de procesos nuevos o modificados (véase Figura 60).(Erl, 2019)

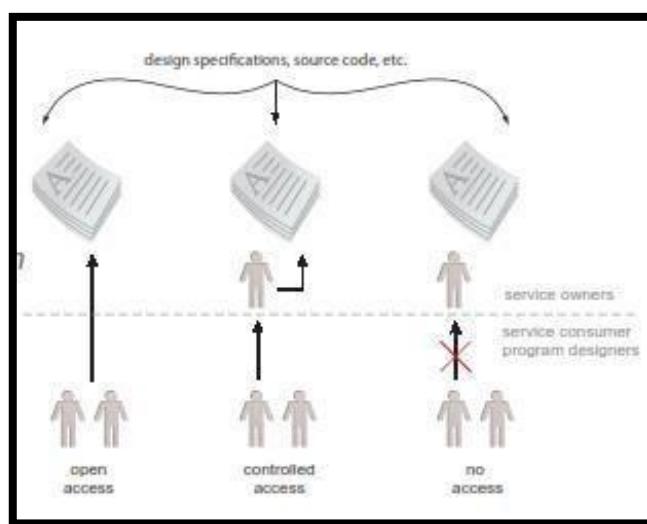


Figura 60. Abstracción de servicios en SOA.(Erl, 2019)

La aplicación de este principio puede convertirse eficazmente en un servicio dentro de una "caja negra" donde la única información sobre el servicio es lo que se publica en su contrato (que puede abarcar lo que también se publicó en un registro de servicios).(Erl, 2019)

Por lo tanto, el contenido del propio contrato del servicio es un punto focal primario para el que existen diferentes niveles de abstracción (detallados, concisos y optimizados).(Erl, 2019)

Como resultado de este principio, los diseñadores de consumidores de servicios pueden no estar conscientes de que un servicio está componiendo a otros.(Erl, 2019)

Esto pone un gran énfasis en la fiabilidad y la previsibilidad de un servicio, independientemente de lo que puede ser encapsulado (también se plantea cuestiones como que debe ser publicado dentro de los SLA).(Erl, 2019)

3.5.2.3.4 Reutilización de servicios

Los servicios contienen y expresan lógica desconocida y pueden ser posicionadas y reusadas como recursos empresariales. Dentro de una arquitectura orientada a servicios, la reutilización representa una característica de diseño objetivo que está ligado a alcanzar repetidos ROI para servicios agnósticos. Este principio combina técnicas de diseño de productos comerciales tradicionales con entrega de proyectos empresariales tradicionales. La colocación de servicios como recursos empresariales reutilizables se relacionan con el patrón de diseño de lógica centralizada que dicta que cada servicio reutilizable sea el punto de acceso único para el cuerpo de la lógica que representa. Cuando se combinan, la lógica y la centralización del contrato resulta ser muy estandarizado y el inventario de servicios normalizados ayudan mucho a maximizar la potencialidad de reusabilidad y la flexibilidad del consumo acoplamiento.(Erl, 2019)

Los procesos de entrega de proyectos por lo general tienen que ser cambiados como resultado de la incorporación constante de este principio a fin de asegurar que la lógica centralizada sea siempre respetada y que la reutilización potencial de los servicios agnósticos sea maximizada.(Erl, 2019)

El más grande obstáculo para realizar este principio es usualmente asociarlo con la superación de la resistencia cultural a estos cambios.(Erl, 2019)

3.5.2.3.5 Autonomía de servicios

(Erl, 2019) menciona lo siguiente: “*Los servicios ejercen un alto nivel de control sobre su subyacente tiempo de ejecución de entorno*”.

El control de un servicio que tiene sobre su aplicación subyacente en tiempo de ejecución, más predecible su comportamiento en tiempo de ejecución será. Reducir el acceso compartido de los recursos de los servicios y aumentar el aislamiento físico puede incrementar la capacidad del funcionamiento de un servicio de manera autónoma.(Erl, 2019)

La autonomía de servicios individuales es especialmente importante para la eficacia de las composiciones de servicios. Debido a la composición de un servicio otra automáticamente pierde su autonomía, el nivel de autonomía de un controlador compuesto puede limitarse a alcanzar a menudo los niveles de autonomía de sus miembros compuestos (véase Figura 61).(Erl, 2019)

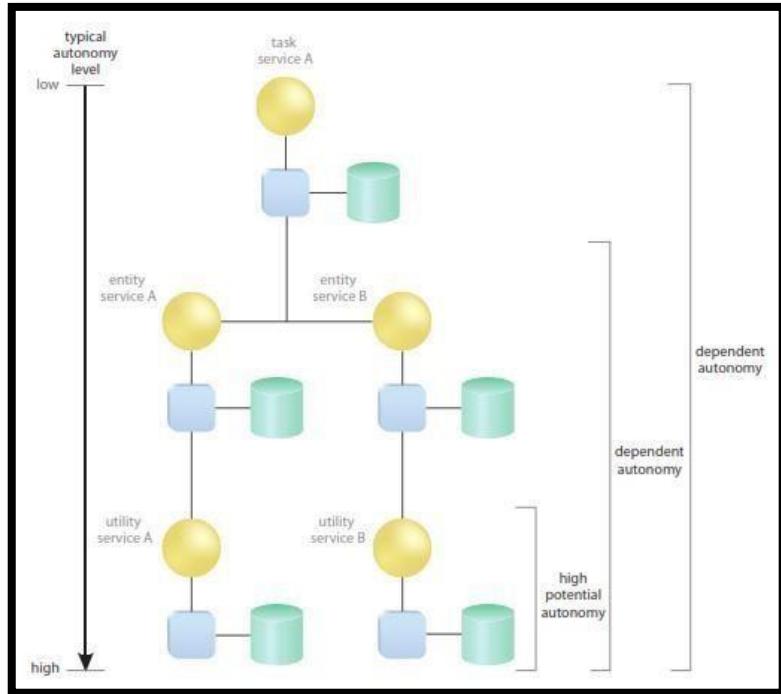


Figura 61. Autonomía de servicios en SOA(Erl, 2019)

3.5.2.3.6 Servicios sin estado

(Erl, 2019) menciona lo siguiente: “*Los servicios minimizan el consumo de recursos haciendo referencia a la gestión de la información de estado cuando sea necesario*”.

Dependiendo de la naturaleza de su lógica y su rol dentro de una composición, un servicio puede necesitar transitar a través de diferentes estados (véase Figura 62) y puede necesitar administrar diferentes tipos y cantidades de estados de datos.(Erl, 2019)

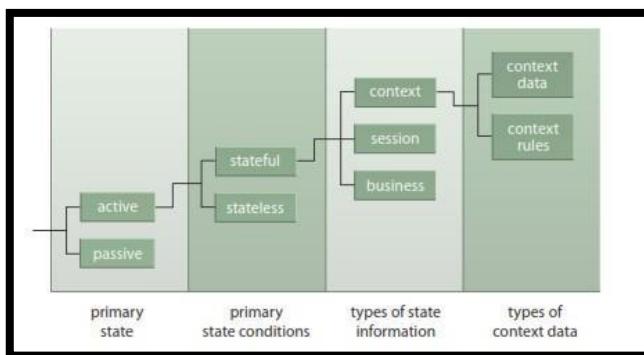


Figura 62. Tipos de estados de datos en SOA(Erl, 2019)

La gestión de datos de estado consume recursos del sistema y puede dar a lugar una carga significativa de recursos cuando varias instancias de servicios se invocan al mismo tiempo, sobre todo con los servicios agnósticos que están involucradas en la automatización de múltiples procesos de negocio.(Erl, 2019)

Por lo tanto, la delegación temporal y el aplazamiento de la administración del estado pueden aumentar la escalabilidad del servicio y soportar un amplio rango de reusabilidad y descomposición con el tiempo.(Erl, 2019)

Los estados de los datos son comúnmente diferentes en tiempo de ejecución permitiendo que un servicio permanezca activo y sin estado, mientras que se produce otro procesamiento.(Erl, 2019)

3.5.2.3.7 Descubrimiento de servicios

(Erl, 2019) menciona que: “*Los servicios se complementan con los metadatos de comunicación por el cual pueden ser descubiertos e interpretados de manera efectiva*”.

De los cuatro tipos de servicios de información meta funcionales (Funcional, Tecnológica, Programable y Calidad de Servicio) y de calidad de los datos son más relevantes cuando se centra en la calidad de las comunicaciones de un servicio para fines de detectabilidad e interpretación.(Erl, 2019)

La aplicación de este principio es compatible con un proceso estandarizado de descubrimiento de servicios e interpretación dentro de una organización mediante el uso de un registro de servicios como repositorio central de datos de servicio de metadatos.(Erl, 2019)

Un objetivo clave de este principio es permitir la amplia gama de miembros del equipo de proyecto para llevar a cabo eficazmente el proceso de descubrimiento y no limitarla a aquellos con experiencia técnica.(Erl, 2019)

Tanto los contratos de servicio y registros dentro de un registro de servicios contienen metainformación con capacidad de descubrimiento y características de interpretabilidad.

Gran parte de esta información se relaciona y se origina en el documento de perfil de servicio que puede haber sido creado y mantenido desde que el servicio fue conceptualizado durante la fase del modelado de servicios.(Erl, 2019)

3.5.2.3.8 Composición de servicios

(Erl, 2019) menciona que: “*Los servicios son participantes efectivos de composición, independientemente del tamaño y la complejidad de la composición*”.

La orientación de servicios es un paradigma de diseño con un acercamiento distinto para llevar a cabo una separación de intereses.(Erl, 2019)

Son capaces de abordar las preocupaciones agnósticas en interrelacionados que pueden ser reutilizados para resolver múltiples problemas.(Erl, 2019)

Esto requiere un medio eficaz de descomposición de lógica resolutiva y descomposición iterativa para poder resolver nuevos problemas.(Erl, 2019)

Este principio se basa principalmente en la capacidad de un servicio de actuar como un miembro de composición eficaz para que pueda apoyar la realización de los nuevos requisitos de negocios que puedan ser cumplidas por el conjunto de composiciones de servicios.(Erl, 2019)

El potencial de composición de un servicio aumenta y se vuelve cada vez importante a medida que más servicios estén disponibles dentro de un inventario de servicio dado.(Erl, 2019)

Un aspecto clave de este principio y de composiciones de servicios en general es que las preocupaciones individuales son, de hecho, resuelto por las capacidades de servicio, ya que es la capacidad que se componen dentro de una composición de servicios.(Erl, 2019)

El éxito y diseño de la composición de servicios se basa en el potencial de la composición colectiva de cada miembro.(Erl, 2019)

3.5.3 Microservicios

3.5.3.1 Definición

Según (Perez-Herrera, 2015), este tipo de arquitectura tiene como idea principal dividir el software en servicios que sean lo más independientes posibles entre ellos distribuidos en diferentes lugares lo cual lo convierte en un sistema distribuido.

Según (Fowler & Lewis, s.f.), los principales aportantes en este tipo de arquitectura, los microservicios se definen como un estilo arquitectural en el que múltiples servicios funcionan individualmente desplegados de una manera automatizada, además se comunican con mecanismos ligeros generalmente usando un recurso API basado en HTTP.

De acuerdo con(Newman, 2015, págs. 2-3), los microservicios son servicios autónomos que trabajan juntos. Detallando a más profundidad analiza los sistemas monolíticos los cuales menciona que usan un código fuente en muchos casos muy extensos, lo cual hace que hacer nuevos cambios o corregir errores sea más dificultoso, en contraparte lo que se desea con microservicios es que sea modular para esto necesita ser cohesivo ósea agrupar código con la misma razón y separar por otro lado código que cumpla diferentes razones. Es por eso por lo que microservicios se enfocan en hacer servicios en base a los límites del negocio de esta forma se evita que estos sean demasiado grandes. Sobre esto menciona que los microservicios deben ser lo suficientemente pequeños y no muy pequeños, mientras más se reduzca el tamaño más crecen las ventajas, pero a la vez las desventajas de la implementación de esta arquitectura; también depende del tamaño del equipo dado que se asignan pocas personas.

Cada microservicio debe ser desplegado de manera aislado en una plataforma como servicio (PAAS), es por eso por lo que se trata de evitar que múltiples servicios estén en un solo lugar. Las comunicaciones entre estos son mediante llamadas en red.(Newman, 2015)

3.5.3.2 Principios

Menciona que se puede escoger que principios adoptar en base a la organización, siempre conociendo los riesgos que ello aqueja.(Newman, 2015, págs. 45-50)

De acuerdo con (Newman, 2015, págs. 45-50), se mencionan los siguientes principios que se resaltan más:

Modelar entorno a conceptos de negocios

Se menciona que en base a la experiencia de interfaces basadas en límites del contexto de negocio son más estables. Al modelar en dominios se asegura que se reflejen los cambios de los procesos de negocio más fácilmente.

Adoptar la cultura de automatización

Dado que los microservicios aumentan la complejidad, es bueno contar con fortalecer una cultura de automatización con herramientas que soporten los microservicios. El testeo automatizado es esencial, es un proceso más complejo que los sistemas monolíticos, también se menciona que se debe adoptar la integración continua para una rápida retroalimentación en la calidad de la producción.

Considerar definir ambientes que especifiquen las diferencias de uno y otro pero que puedan desplegarse de una manera uniforme.

Esconder detalles internos de implementación

Para maximizar la habilidad de permitir la evolución de los microservicios independientes es vital que se esconda los detalles de su implementación, definir bien los límites de negocios es un buen soporte.

Los servicios deben esconder su base de datos para evitar el acoplamiento que si tienen las tradicionales arquitecturas orientadas a servicios y para la generación de reportes hacer un movimiento masivo de datos que puede darse por eventos.

Los servicios deben proveer APIs con tecnología agnóstica (que son servicios que no se preocupan del contexto en el que son llamados) que permite que se usen diferentes tecnologías. Considerar usar REST que formaliza la separación de la implementación interna y externa.

Descentralizar todas las cosas:

Para maximizar la autonomía de los microservicios se necesita buscar constantemente los cambios para delegar el control de los servicios a los equipos.

Hay que asegurar que cada equipo tiene sus servicios es un paso importante, ellos son responsables de los cambios, de cuando lanzan una nueva versión, cada equipo debe ser experto en el dominio del negocio en el cual están creando. Intentar adoptar un modelo de gobierno compartido en los que cada equipo colectivamente comparten responsabilidades de la visión técnica del sistema.

Se puede este principio a la arquitectura, evitar enfoques como los buses de servicios empresariales o sistemas de orquestación los cuales centralizan la lógica de negocios y sus servicios son tontos. En vez preferir coreografía sobre orquestación y usar un middleware tonto con endpoint inteligentes que aseguren mantener la lógica asociada y

su información dentro de los límites de los servicios ayudando a tener las cosas cohesivas.

Despliegue independiente:

Se debe tratar de asegurar que los microservicios puedan ser desplegados por ellos mismos. Aun cuando hay cambios deberían coexistir versiones diferentes que permitan a los clientes cambiar sobre el tiempo. Esto permite optimizar la velocidad de nuevas características e incrementar la autonomía de los equipos ya que no necesitan orquestar sus despliegues. Se puede adoptar por el modelo un servicio por host que reduce el impacto de afectar a otro servicio no relacionado. En general se recomienda no que el despliegue de un servicio no bloquee a otro y que los clientes decidan cuando se actualizan ellos mismos.

Aislar las fallas:

Una arquitectura de microservicios puede ser más resistente que un sistema monolítico, pero solo si se cuenta con un plan de fallas dentro del sistema. Si no se cuenta con el hecho de que puede y va a fallar, el sistema va a comenzar a fallar y puede hacer al sistema más frágil.

Cuando se usan llamadas no se debe tratar las llamadas remotas como las llamadas locales en las que hay una especie de fallo.

Comprender que y como usar los circuitbreakers (interruptores automáticos) para limitar las consecuencias de un componente defectuoso. Comprender el impacto de que pasaría si una parte del sistema se comporta erróneamente, a veces se debe sacrificar la disponibilidad o consistencia.

Altamente observables:

No se puede confiar en observar el comportamiento de una sola interfaz para ver si el sistema funciona correctamente, En cambio se debe usar una vista de todo unido para ver lo que está pasando. Usar monitoreo con transacciones de prueba que simulen un comportamiento real. Es importante agregar logs, estadísticas y usar IDS correlacionados que permitan el rastreo de las llamadas del sistema.

3533 Ventajas

Según(Newman, 2015, págs. 4-7), las ventajas que podemos encontrar en el uso de la arquitectura microservicios son las siguientes:

Tecnología heterogénea:

Dado que los microservicios pueden estar en diferentes sitios independientes unos de otros se pueden usar diversidades de tecnología, esto nos ayuda a encontrar la herramienta apropiada para el tipo de trabajo que se realice en vez de conformarse a una sola tecnología estandarizada.

En la siguiente imagen (véase Figura 63) se muestra que en fines de buscar un mejor rendimiento se pueden realizar distintas bases de datos. El ejemplo trata de una red

social que guarda las relaciones de usuarios en una base de datos de grafos, las imágenes en un almacenamiento blob y el uso de tecnología java, y las publicaciones (posts) mediante tecnología Ruby usando un almacenamiento de documentos

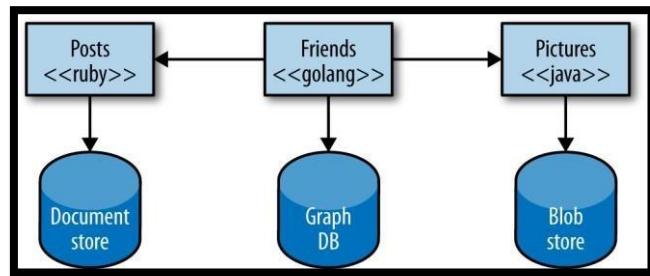


Figura 63. Ejemplo práctico de microservicios: red social(Newman, 2015, pág. 4)

Con los microservicios se logra que nuevas tecnologías puedan ser integradas rápidamente, esto debido mayormente a que los riesgos son mucho menores que una sola aplicación monolítica.

Resistencia: La arquitectura de microservicios debe ser resistente a fallos, si algo falla y no están en cascada se puede aislar el problema y el aplicativo continúa trabajando; por otro lado, en la arquitectura monolítica si el servicio falla todo el aplicativo debe pararse, una forma de mitigar el error se corría en varias máquinas el aplicativo. Sin embargo, el autor menciona que se debe ser cuidadoso para asegurar que los microservicios puedan mejorar la resistencia a fallos. La red puede y va a fallar para eso la arquitectura debe estar preparada.

Escalabilidad: Cuando se tiene un servicio monolítico todo debe ser escalado junto, dado que está relacionado y restringido por lo grande que es el aplicativo. Con más pequeños servicios se puede escalar solo los servicios que se desean

En la siguiente imagen (véase Figura 64), se muestra el escalado solo en la parte que se requiere pudiendo poner menos hardware a las que no lo requieren.

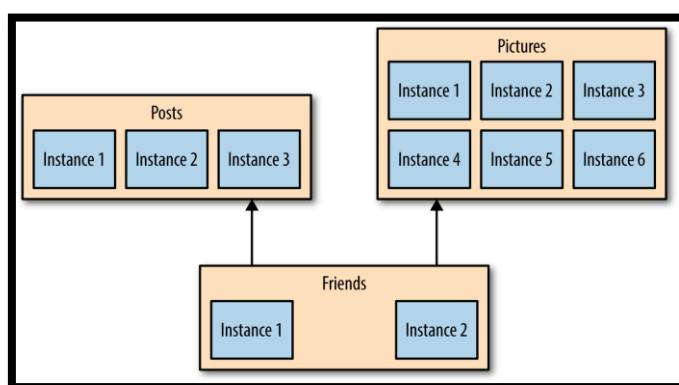


Figura 64. Escalabilidad de los microservicios(Newman, 2015, pág. 6)

Facilidad de desplegar: En una aplicación monolítica la tarea de desplegar lleva muchos riesgos, lanzar una versión demanda mucho tiempo, ya que se toma un tiempo

para solucionar todos los posibles errores entre versiones. Con microservicios se pueden realizar cambios y desplegar solo dicho servicio, esto permite nuestro código desplegar rápidamente, lo cual permite que las nuevas funcionalidades se puedan ver más rápido

Alineamiento con la organización: Existen problemas asociados con largos equipos y largos códigos fuentes. Es realidad que los equipos más pequeños con códigos fuentes más pequeños tienden a ser más productivos. Microservicios ayuda a alinear la arquitectura a la organización, ayudando a reducir el número de personas trabajando sobre un código fuente.

Componibilidad: Una de las claves de los microservicios es la reusabilidad de funcionalidades en otras palabras componentes. Los microservicios pueden ser usados de diferentes formas en diferentes lugares. Esto es importante ahora que todo está conectado a diferencia de antes que se limitaba a una sola aplicación en escritorio, web o celular. Es por eso por lo que la arquitectura debe ser adaptable a nuevos tipos de comunicación.

Optimizado para ser reemplazable: A menudo en grandes organizaciones hay aplicativos antiguos que no se remueven o cambian porque son muy grandes y ponen en riesgo al trabajo. Con servicios individuales el costo de reemplazarlos por otras implementaciones o incluso eliminarlas son mucho más fáciles de manejar.

3.5.4 Desventajas

Según (Fowler & Lewis, s.f.), las desventajas que podemos encontrar en el uso de la arquitectura microservicios son las siguientes:

Distribución: Los microservicios que usan sistemas distribuidos promueven la modularidad, pero el hecho que sea distribuido puede ser su principal desventaja. Las complejidades de estas son:

- 1) Rendimiento: El rendimiento de realizar llamadas a funciones remotas son lentas a comparación de llamadas a funciones en el proceso. Esto se puede mitigar mediante:
 - a. Incrementar la granularidad: De esta manera se haría menos llamadas de funciones remotas, lo cual por otro lado complicaría la programación del modelo de la aplicación, se tiene que saber cómo se interconectarán los servicios y llamar a un servicio colaborativo al menos una vez.
 - b. Asíncrono: Si se realizan tareas asíncronas en paralelo se tiene la ventaja de que puede durar a lo mucho lo que la más lenta de todas las llamadas remotas demoraría.
- 2) Confiabilidad: Las llamadas a funciones en el proceso se esperan que funcionen siempre, pero una llamada remota podría fallar en cualquier momento. Si hay más cantidad de microservicios la probabilidad de que ocurra es más alta. Esto se puede mitigar mediante la colaboración asíncrona ajustando el manejo de fallos y pueda resultar en una mejor recuperación, aun así, se debe agregar complejidad adicional averiguando las consecuencias de cada llamada remota.

Consistencia eventual: Cuando se realiza una operación en el sistema puede ser que este se quede colgado hasta que finalice su operación, esto es un problema de usabilidad ya que el usuario incluso desconoce si ha ocurrido un error o no.

Esto se puede complicar en la medida que la lógica de negocio tome decisiones sobre información inconsistente cuando esto sucede diagnosticar el problema se puede volver extremadamente difícil.

Con los microservicios la cantidad de consistencias eventuales va a crecer debido a que la base de datos esta descentralizada, mientras que en una aplicación monolítica todo forma parte de una transacción. Para esto los desarrolladores deben prever estos problemas de la consistencia y saber cómo detectarlos antes de que el código ocasione estos problemas.

Complejidad operacional: Al ser cada unidad independiente para ser desplegada se hace muy conveniente al momento de desplegar, pero esto añade otro tipo de complejidades como tener un número mucho mayor de microservicios que de aplicaciones. Muchas organizaciones tendrán la dificultad de lidiar con tal grande cantidad de servicios

Esto refuerza el rol de la entrega continua, y en la arquitectura de microservicios es esencial. Además, la complejidad operacional también se ve incrementada debido a la creciente demanda de manejo de servicios y la monitorización de estos.

Manejar esta complejidad requerirá nuevas habilidades y herramientas; aun así, seguirá siendo más compleja que una arquitectura monolítica. Además, se necesita introducir la cultura “devops” que consiste en una excelente colaboración entre desarrolladores, operaciones y todo aquel involucrado en la entrega del software

3.6 Casos de éxito

A continuación, se detallará los sistemas que han demostrado su éxito en diferentes áreas, con fines científicos o experimentales.

3.6.1 Sistema web de gestión académica en el centro de idiomas de la ESPAM MFL

(Ganchoso & Vera, 2015)

3.6.1.1 Objetivos

Los objetivos son los siguientes:

- Mejorar la gestión de procesos académicos del centro de idiomas ESPAM MFL donde se encuentran procesos de preinscripción, admisión, matrícula y gestión de expedientes.
- Disminuir trámites a nivel personal y el uso de papeles.
- Reducir la carga de trabajo del personal debido a la mayor demanda.

3.6.1.2 Solución:

La solución consiste en diseñar e implementar una solución que permita mejorar la propuesta pedagógica de calidad de formación del estudiante para esto brinda los procesos de preinscripción, selección, admisión, planes de estudio, modelos pedagógicos, ofertas de programas e información académica. Además, se mejora el manejo de información, disponibilidad, menor tiempo y la reducción del uso de papel

3.6.1.3 Resultados:

El sistema fue utilizado por 2 ciclos académicos permitiendo la mejora de los procesos académicos y en la reducción de tiempo, y mejora de la atención brindada. (véase Tabla 2)

Tabla 2

Mejora de procesos de gestión académica ESPAM-MFL

Proceso de gestión académica	Tiempo en segundos		
	Sin sistema	Con sistema	% de mejora
Matrícula (por alumno)	50	35	30%
Elaboración de listados (por curso)	93	15	84%
Búsqueda de reporte completo de notas de un módulo (por curso)	120	20	83%
Búsqueda de historial académico (por alumno)	600	15	98%
Elaboración de certificados (por alumno)	900	50	94%

Fuente: (Ganchoso & Vera, 2015)

3.6.2 Implementación de un sistema web para la mejora del proceso académico de la institución educativa wari-vilca-huayucachi

(Acevedo, 2018)

3.6.2.1 Objetivos

A continuación, se muestran los objetivos:

- Disminuir tiempo en cambios del plan curricular.
- Disminuir tiempo y errores de registro de alumnos y matrículas.
- Disminuir tiempo de entrega de registros auxiliares.
- Disminuir errores en asignación de notas de alumnos
- Disminuir tiempo en correcciones de notas.
- Entrega de boleta de notas más rápido

3.6.2.2 Solución:

Mejorar el proceso administrativo académico en la Institución Educativa “Wari-Vilca”- Huayucachi, 2018 mediante la implementación de sistema web.

Se empleo la metodología RUP y su arquitectura (véase Figura 65) consta de 3 capas: capa de presentación, capa de procesos y capa de datos

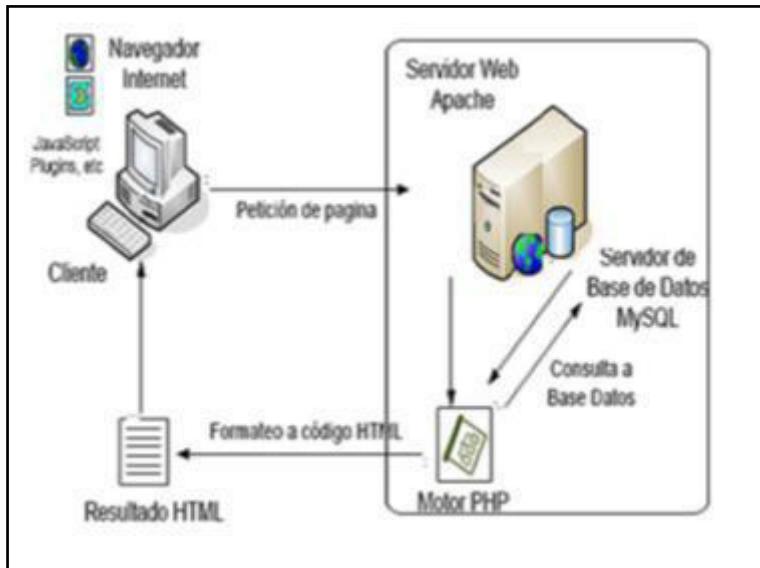


Figura 65.Arquitectura de la solución para el Instituto Educativo Wari-Vilca(Acevedo, 2018)

3.6.2.3 Resultados:

Los resultados obtenidos con la implementación del sistema web comprueban lo siguiente:

- Se logró mejorar el proceso administrativo académico.
- El sistema web influye positivamente en un 34.4% en el proceso de entrega de boleta de notas mejorando el proceso administrativo académico.
- El sistema web influye positivamente en un 25% en el proceso de consultas y reportes mejorando el proceso administrativo académico.
- Los padres de familia y/o apoderados pueden tener acceso rápido, detallado y confiable a la información solicitada referente a lo académico en menor tiempo.
- Se facilitó el control de las notas por parte del personal administrativo teniendo ahora toda la información en el sistema web.
- Se redujo el tiempo de consulta por parte de los padres de familia y/o apoderados.

3.6.3 Implementación de un sistema web para optimizar la gestión académica en la I.E. “Villa corazón de Jesús” del distrito de San Juan de Lurigancho. (Berrospi & Pilar, 2013)

3.6.3.1 Objetivos

- Reducir el tiempo de consulta del estado del nivel académico del estudiante de la I.E.P. Villa Corazón de Jesús
- Facilitar el llenado de notas por parte de los docentes a cargo de un grado y curso específico
- Reducir el tiempo de consulta de las asistencias de todo el personal de trabajo y alumnos de la I.E.P. Villa Corazón de Jesús

3.6.3.2 Solución

La solución consiste en la implementación de un sistema web que permita optimizar la gestión académica de la I.E.P. Villa Corazón de Jesús de San Juan de Lurigancho. Entre sus funciones tendrá el registro de notas, registros académicos, brinda control académico a los padres y permitirá la gestión de asistencias.

3.6.3.3 Resultados

A continuación, se muestran beneficios tangibles (véase Tabla 3) después de la implementación del sistema web con relación al porcentaje de cómo fue anteriormente.

Tabla 3

Beneficios tangibles del sistema web para I.E. Villa Corazón de Jesús

Nombre de beneficio	Detalle
Evitar la pérdida de la información.	Reducción al 100%.
Registro y consulta de datos de los alumnos.	Incremento en un 40%
Ahorrar el material de trabajo manual.	Incremento en un 50%
Verificación y registro de asistencias de alumnos y empleados	Incremento en un 50%
Consulta de las notas de los alumnos.	Incremento en un 30%

Fuente: (Berrospi & Pilar, 2013)

A continuación, se muestran los beneficios intangibles (véase Tabla 4) luego de la implementación del sistema web en relación de cómo fue anteriormente.

Tabla 4

Beneficios intangibles del sistema web para I.E. Villa Corazón de Jesús

Beneficios y mejoras	Sin el uso de un sistema web			Con el uso del sistema web			Total de beneficio
	Tiempo	RR. HH	Costo	Tiempo	RR. HH	Costo	
a.							
Satisfacción del padre o apoderado en consulta de datos del alumno.	10 min.	1	S/. 2,000.00	1 min.	1	S/. 1,200.00	800.00
b. Ahorro de tiempo en el proceso de registro de asistencia de los empleados.	1.5 min.	1	S/. 2,000.00	10 seg.	1	S/. 1,200.00	800.00
c. Ahorro de	30	1	S/.	1 min.	1	S/.	S/.

tiempo en el proceso de consulta de asistencia de los empleados.	min.	2,000.00	1,200.00	800.00
Total			S/. 2400.00	

Fuente: (Berrospi & Pilar, 2013)

Y finalmente se obtuvo un retorno a la inversión en un año de 6.7 el cual fue positivo, y la tasa de interna de retorno en 17% mayor a la tasa de descuento 10% el cual indica que es un proyecto que no genera pérdidas y un valor actual de neto de S/. 5.350.76 soles que indica que es un proyecto viable.

3.7 Benchmarking

Se seleccionaron tres casos de estudio cuyos aportes obtuvieron mayor impacto con la resolución del problema (véase Tabla 5).

Tabla 5
Benchmarking de las soluciones

Características	Sistema Web de Gestión Académica en el centro de idiomas de la ESPAM	Diseño e implementación de una app sobre desarrollo sostenible con backend de arquitectura basada en microservicios y de una reactnativefrontend app	Aplicación móvil de control académico utilizando la arquitectura de microservicios bajo la metodología de desarrollo Scrum
Metodología de desarrollo	Scrum	Scrum	Scrum
Tecnología Front-end	Aplicativo Web	Aplicación híbrida (ReactNative)	Aplicación híbrida (ReactNative)
Tecnología Back-end	Arquitectura de 3 capas	Arquitectura de microservicios	Arquitectura de microservicios
Permite la automatización y optimización de los procesos manuales en general	Si	Si	Si
Multiplataforma (Android, IOS y Web)	No	Si	Si
Uso de tecnología Open Source	Sí	Sí	Sí
Infraestructura de	On-Premise	Cloud Computing	Cloud Computing

despliegue	(Amazon Web Services)		
Nivel de complejidad (cantidad de funcionalidades que ofrece el sistema)	Regular	Regular	Alto

(Elaboración Propia)

Adicionalmente a las características mencionadas, nuestro aporte “Aplicación móvil de control académico utilizando la arquitectura de microservicios bajo la metodología de desarrollo Scrum” propicio las siguientes ventajas competitivas:

- Solución personalizada, al captar los requerimientos (a través de historias de usuario) por parte de la directora de escuela de la Escuela Profesional de Obstetricia (UNMSM).
- Respaldo de la Escuela Profesional de Obstetricia (UNMSM), dado que se cuenta con la constancia de validación del producto por parte de la directora de la EPO – UNMSM(véase Anexo 2).
- Económicamente viable, debido al uso masivo de los teléfonos inteligentes; estudiantes, docentes y otros administrativos pueden realizar sus actividades académicas accediendo al sistema de control académico a través de sus celulares.

4 RESOLUCIÓN DEL PROBLEMA

4.1 Cuadros comparativos

4.1.1 Comparación metodología de desarrollo

Para seleccionar la metodología de desarrollo se analizó mediante dos cuadros comparativos siendo la primera comparativa entre metodologías tradicionales y ágiles (véase Tabla 6), y la segunda comparativa es un cuadro de resumen cubrimiento porcentual por área de proceso de las metodologías: XP, Scrum e Iconix (véase Tabla 7).

A continuación, se muestra la primera comparativa entre metodología ágil (Scrum) y metodología tradicional (RUP) (véase Tabla 6).

Tabla 6
Comparativa entre metodologías tradicionales y ágiles

	Tradicional	Ágil
Prioridad de negocio	Cumplimiento	Valor
Tiempo del proyecto	Largo	Corto
Tamaño del equipo	Grande	Pequeño
Interacción con el cliente	Mínima	Máxima
Ambiente del desarrollo	Controlado	Incertidumbre
Dirección	Organizativa	Colaborativa
Rigidez del producto	Cerrado	Ampliable
Requisitos	Claros	Ambiguos
Riesgo de fallos	Bajo	Alto
Enfoque de desarrollo	Procesos	Personas
Necesidad de documentación	Alta	Baja
Probabilidad de cambios en el equipo	Alta	Baja
Roles intercambiables	No flexible	Flexible

Fuente: (García, 2018)

Se elige a la metodología Scrum ya que:

- Está más enfocada en el desarrollo de software debido al tiempo que es más reducido.
- Las entregas son muchas más rápidas que RUP (los sprints duran de 1 a 4 semanas mientras que en RUP las iteraciones están basadas en más tiempo).
- Exige solo la documentación necesaria.
- Está más orientado a una constante interacción con las personas.
- Es más abierto a los cambios.

A continuación, se mostrará el segundo comparativo entre SCRUM y otras metodologías ágiles (véase Tabla 7). Se muestran los niveles de CMMI, las áreas de proceso de cada nivel y las metodologías ágiles Scrum, XP e Iconix. Aquí se evalúan la cantidad de subprácticas de cada nivel según CMMI que las metodologías ágiles pueden cumplir de forma de porcentaje que permite realizar un comparativo entre éstas.

Tabla 7

Resumen porcentual por área de proceso de metodologías ágiles (Scrum, XP, Iconix)

Nivel	Área de proceso	SCRUM	XP	ICONIX
2	1.REQM	80%	80%	90%
2	2.PP	68%	43%	14%
2	3.PMC	88%	55%	0%
2	4.SAM	0%	0%	0%
2	5.MA	38%	38%	0%
2	6.PPQA	13%	50%	13%
2	7.CM	0%	100%	0%
3	8.REQD	93%	93%	83%
3	9.TS	0%	31%	44%
3	10.PI	0%	61%	53%
3	11.VER	0%	72%	69%
3	12.VAL	80%	90%	55%
3	13.OPF	0%	0%	0%
3	14.OPD	0%	0%	0%
3	15.OT	0%	14%	0%
3	16.IPM	58%	33%	0%
3	17.RSKM	7%	36%	0%
3	18.DAR	0%	0%	0%
4	19.OPP	0%	0%	0%
4	20.QPM	0%	0%	0%
5	21.OPM	0%	0%	0%
5	22.CAR	0%	0%	0%

Fuente:(Brito J. A., 2016)

Scrum se destaca más en gestión que la metodología XP o Iconix, debido a que las áreas que destacan son PP (Planeación de proyectos), PMC (Monitoreo y Control de

Proyectos) e IPM (Administración de Proyectos Integrado), mientras que XP o Iconix están más enfocados en la construcción y desarrollo del producto.

Debido a que nuestro proyecto se inicia de cero, necesitamos realizar una buena gestión sobre las funcionalidades que tendrá el sistema, además debe estar enfocado en un avance ágil para dar entregables a los clientes en cortos periodos de tiempo, es por eso por lo que optamos por la metodología SCRUM para la planificación y el desarrollo del producto.

4.1.2 Comparación frontend (lado cliente)

Para seleccionar la tecnología frontend de desarrollo se analizó mediante dos cuadros comparativos siendo la primera comparativa entre frameworks multiplataformas Xamarin, ReactNative, Flutter (véase Tabla 8), y la segunda comparativa es un análisis comparativo de criterios con escala de likert entre soluciones como desarrollo nativo, ReactNative, Native Script e Ionic (véase Tabla 9).

A continuación, se analizará el comparativo (véase Tabla 8) entre frameworks multiplataforma Xamarin, ReactNative y Flutter.

Tabla 8

Comparación de frameworks multiplataforma Xamarin, ReactNative, Flutter

	Xamarin	ReactNative	Flutter
Lenguaje	C#	Javascript	Dart
Curva de aprendizaje	C# es el 7mo lenguaje de programación más popular del 2019 y no requiere de aprendizaje de desarrolladores C#.	JavaScript es el lenguaje más popular del 2019 y no se requiere de aprendizaje para desarrolladores de JavaScript and React.	Uso de un lenguaje casi desconocido pero fácil de aprender para desarrolladores OOP
Experiencia del desarrollador	-	Cambio en caliente y puede ser usado en cualquier IDE preferido.	Cambio en caliente con estados y puede ser usado en cualquier IDE preferido.
Mercado, comunidad y soporte	1.4 millones de desarrolladores	Dos veces más grande y popular que Xamarin y el 6to framework más popular del 2019	Comunidad pequeña, pero de rápido crecimiento.
Documentación	Bien escritor y regularmente	Algunas veces los documentos están	Bien escritor y regularmente

	mantenidos	desactualizados	mantenidos
Componentes y librerías listas para usar	Pocos y no son de código abierto	Larga lista de librerías de terceros.	Pocos, pero son de código abierto.
Código compartido	El desarrollo de la interfaz es dependiente de la plataforma a menos que sea desarrollado por XamarinForms. El código específico de la plataforma se necesita a menudo.	El código específico de la plataforma se necesita con mayor frecuencia. Se espera un promedio del 85%	Un único código base.
Opciones de ambiente de desarrollo	Microsoft Visual Studio y Xamarin Studio	Casi todos los editores de texto e IDEs	Casi todos los editores de texto e IDEs
Aspecto y comportamiento nativo	Si se desarrolla con Xamarin.Forms se puede desarrollar una aplicación cercana a la nativa, pero en sí es una no nativa.	Native indistinguible	Utiliza sus propios widgets con elementos de UI de diseño material.
Licencias y costos	Licencia es requerida para un aplicativo empresarial.	Libre y código abierto	Libre y código abierto
Tiempo de compilación y velocidad de desarrollo	Compilación es más lenta que la solución nativa.	Compilación es más lenta que la solución nativa, pero ofrece recarga en caliente.	Compilación es más lenta que la solución nativa, pero ofrece recarga en caliente.
Despliegue de código y rendimiento del aplicativo	Compilación AOT y cerca del rendimiento nativo	Rendimiento menor debido al bridge(puente) y la compilación JIT	Rendimiento excepcional gracias a los Widgets y Skia.
Pruebas	Cualquier herramienta de .Net puede ser usada. Hay depuración en Visual Studio. Tomar ventaja de Xamarin Test Cloud	Jest es la herramienta de prueba más sugerida.	Las pruebas unitarias, widgets e integración son oficialmente soportadas.

Fuente: (Satei, 2019)

De este comparativo (véase Tabla 8) se elige a ReactNative debido a que:

- Su lenguaje JavaScript es el más usado.
- Se admiten cambios en caliente.
- Se permite el uso de cualquier IDE preferido a diferencia de Xamarin que es usado en un IDE específico.
- Es de código libre al igual que Flutter y diferente con Xamarin.
- Tienen una gran cantidad de librerías de terceros si bien posee documentación algo desactualizada tiene una gran comunidad mayor a Xamarin y a Flutter.
- La apariencia y la forma de los aplicativos en ReactNative se ven como componentes nativos a diferencia de Xamarin o Flutter.

También contamos con una tabla comparativa bajo la escala de Likert de las diferentes soluciones como: desarrollo nativo, reactnative, native script e ionicque (véase

Tabla 9) que se muestra a continuación:

Tabla 9

Resultado de análisis comparativo de criterios con escala de Likert: desarrollo nativo, ReactNative, Native Script e Ionic

Criterios	Soluciones			
	Nativas	ReactNative	Native Script	Ionic
Aprendizaje y calidad de documentación	2	5	3	4
Costo de desarrollo	2	4	4	4
Emuladores y depuración	4	5	4	4
Tiempo de respuesta y velocidad	5	5	4	1
Reconocimiento comercial	2	5	1	2
Reutilización de código y trabajo en equipo	2	5	5	4
Mantenimiento y actualización	2	3	4	4
Total	22	32	25	23

Fuente:(Brito, Gómez, Santos, & Bernardino, 2018)

De este comparativo (véase

Tabla 9) se aprecia que reactnative tiene mejores resultados en términos generales debido a que:

- El aprendizaje es más fácil.
- Se cuenta con documentación accesible.
- El coste de desarrollo es menor en comparación del desarrollo nativo.

- En la emulación y depuración todos poseen posibilidad de emular y depurar, pero en ReactNative se pueden reflejar los cambios inmediatamente.
- El tiempo de respuesta en reactnative es mejor que ionic y native script y casi parecido a la aplicación nativa.
- En la respuesta comercial, si bien reactnative es relativamente nuevo éste ha ido creciendo de forma exponencial.
- En la reutilización y trabajo en equipo la ventaja de reactnative al igual que ionic o native script es que es código reutilizable para plataformas móviles en cambio en desarrollo nativo se requiere un código distinto para cada plataforma
- En el tema de mantenimiento y actualización reactnative tiene mejor ventaja que el desarrollo nativo en el sentido que sólo se modifica un proyecto para los cambios, aunque ionic y native script lleven un poco de ventaja dadas sus versiones estables.

4.1.3 Comparación backend (lado servidor)

Para seleccionar la arquitectura del desarrollo backend se analizó mediante dos cuadros comparativos: la primera comparativa entre la arquitectura monolítica comparada con la arquitectura de microservicios (véase Tabla 10), y la segunda comparativa entre SOA (arquitectura orientada a servicios) con la arquitectura de microservicios (véase Tabla 11).

A continuación, se muestra la primera comparativa (véase Tabla 10), entre la arquitectura monolítica y la arquitectura de microservicios.

Tabla 10

Comparativa entre arquitectura monolítica y arquitectura de microservicios

Categoría	Arquitectura monolítica	Arquitectura de microservicios
Tiempo al mercado	Rápido en el comienzo, lento después cuando el código fuente crece.	Lento en el comienzo por los desafíos tecnológicos que tienen los microservicios. Rápido después.
Refactorización	Difícil de hacer, dado que los cambios afectan muchos lugares.	Fácil y seguro dado que los cambios están contenidos dentro de un microservicio
Despliegue	El monolito entero ha sido desplegado siempre.	Puede ser desplegado en partes pequeñas, un servicio a la vez.
Lenguaje del código	Difícil de cambiar. Tanto como el código fuente sea largo. Requiere gran reescritura	Lenguaje y herramientas pueden ser seleccionadas por servicio. Servicios son pequeños entonces el cambiar es fácil.
Escalamiento	Escalamiento significa desplegar todo el monolito.	Escalamiento puede darse por servicio.
Habilidades	No requiere mucho, dado que	Múltiples diferentes

DevOps	el número de tecnologías es limitado	tecnologías de DevOps son requeridas.
Entendible	Difícil de entender dado que su complejidad es alta. Muchas partes móviles.	Fácil de entender como código fuente es estrictamente modular y los servicios usan SRP (Principio de responsabilidad única)
Rendimiento	No sobrecarga de comunicaciones. Componentes tecnológicos podrían no soportar rendimiento.	Comunicación agrega sobrecarga. Posible aumento en rendimiento dado las alternativas de tecnologías

Fuente: (Kalske, Mäkitalo, & Mikkonen, 2018)

Se puede resaltar del comparativo (véase Tabla 11) que elegimos la arquitectura de microservicios en vez de la arquitectura monolítica, a continuación, se detallan las consideraciones que se tomaron:

- Si bien el tiempo al mercado demora más al comienzo el aplicativo será más robusto y realizar los cambios no será tan tedioso en el futuro, dado que creemos que el software podría crecer y adaptarse a las complejidades del control académico no solo de la EP de Obstetricia sino también de expandirse a más escuelas.
- El aplicativo se plantea usar como un proyecto mínimo viable y adaptarse a la metodología ágil por lo que realizar cambios es muy importante, entonces éste debe ser fácil de realizar cambios en su código fuente.
- Se prefiere despliegues individuales para no afectar a otros servicios que si funcionan correctamente.
- Es bueno contar con diversidad tecnológica de lenguajes de programación, debido a que si en cierto momento se considera mejor dados temas de rendimiento será mucho más fácil de cambiar.
- El escalamiento es preferible que sea por servicio a por monolito(todo) dado que se enfoca en las funcionalidades que necesitan escalarse, así mismo se aprovechan mejor los recursos.
- Consideramos importantes las habilidades devops para reducir el tiempo en que se demora la verificación, despliegue de forma automática de los servicios y así reducir las actividades no autónomas.
- Es bueno que sea entendible dado que el tamaño de un microservicio es mucho menor al de un monolito.
- El rendimiento puede mejorarse con el uso de variedad tecnológica y no estar apegado a pocas tecnologías conforme lo requiera el proyecto.

A continuación, se muestra la segunda comparativa(véase Tabla 11), entre la SOA y la arquitectura de microservicios.

Tabla 11
Comparativa entre SOA y arquitectura de microservicios

ASUNTO	SOA	Microservicios
Despliegue	Despliegue monolítico todos a la vez	Despliegue individual del servicio
Equipos	Servicios, integración y la interfaz de usuario son manejados por equipos individuales.	Cada microservicio es manejado por equipos individuales
Interfaz de usuario	Portal de todos los servicios	Parte de los microservicios
Alcance de la arquitectura	Toda la compañía/empresa	Un proyecto
Flexibilidad	Ajustes de procesos de negocios en el tope de los servicios	Despliegue rápido e independiente
Mecanismo de integración	Integración inteligente y compleja	Integración simple y primitiva
Nube nativa	No	Si
Administración / Gobierno	Centralizado	Distribuido
Almacenamiento de datos	Compartida	Por unidad
Escalabilidad	Limitado comparado a microservicios, cuello de botella en integrar unidades o sobrecarga de uso de análisis de mensajes. Elasticidad limitada	Mejor escalabilidad horizontal. Elástico
Unidad	Base de datos compartida, enlaces de unidades para servir a procesos de negocios. Bajo acoplamiento	Autónomo, desacoplado, propio contenedor, escalable independientemente.
Comunicación principal	Orquestación	Coreografía
Encaja	Infraestructuras de tamaño largo	Infraestructuras de tamaño mediano
Tamaño de servicio	Granularidad fina o gruesa	Fina granularidad, pequeño
Versionado	Mantenimiento de múltiples servicios de diferentes versiones	Debería ser parte de la arquitectura, más abierta a cambios
Nivel de administración	Centralizado	Anarquía
Localización de	Componente de integración	Servicio particular

las reglas de negocio

Fuente:(Cerny, Donahoo, & Pechanec, 2017)

Se puede resaltar del comparativo(véaseTabla 11) que elegimos la arquitectura de microservicios en vez de la arquitectura SOA, a continuación, se detallan las consideraciones que se tomaron:

- Se prefiere un despliegue individual de un microservicio de esta forma no afecta a otros.
- La interfaz de usuario toma partes de los microservicios para componerlas dentro de su vista.
- El alcance de microservicios se asemeja a nivel de proyecto (“Control académico para EP Obstetricia”).
- La arquitectura de microservicios es flexible ya que cuenta con despliegue rápido y además independiente (no afecta en el despliegue de otros servicios)
- Se prefiere la integración entre los microservicios ya que es de forma sencilla en vez de compleja.
- El despliegue por defecto de microservicios está adaptado para varios entornos especialmente en nube donde los costos muchas veces son menores y mejores recursos a diferencia de contar con servidores físicos que son más costosos y recursos limitados.
- El gobierno es descentralizado ya que cada microservicio contiene la lógica de negocio y éste mismo puede invocar otros servicios.
- La base de datos es independiente en los microservicios y no compartida para mantener el bajo acoplamiento y alta cohesión de los microservicios.
- La arquitectura de microservicios permite realizar escalamiento horizontal que es más potente que el escalamiento vertical.
- El tipo de software encaja en una infraestructura de tamaño intermedia debido a que el alcance del proyecto no es demasiado grande.
- Con respecto a las comunicaciones entre microservicios es simpley no requiere de integraciones complejas como buses.
- Tiene componentes con bajo acoplamiento debido a que son independientes.
- Los microservicios tienen granularidad fina dado que son de tamaño pequeña a diferencia los servicios de SOA que tienen granularidad gruesa y tamaño más grande.
- La administración de los microservicios es anárquica es decir no tiene un “administrador” ya que cada una funciona por si sola.

4.2 Descripción de la solución tecnológica

4.2.1 Inicio

En esta fase se definirá la visión del producto, definir el equipo de trabajo, historias de backlog a alto nivel y definición del plan de lanzamiento.

4.2.1.1 Visión del proyecto

La visión del proyecto es sobre la gestión de la información de las clases como asistencias, notas, encuestas y reportes de manera online que apoye en la mejora de la calidad educativa en la Escuela Profesional de Obstetricia de UNMSM.

La dueña del producto (productowner) será la directora de escuela Clara Diaz Tinoco encargada de brindar todos los requerimientos del negocio, así como asegurar que se dé mayor valor al proyecto.

4.2.1.2 Identificación de scrum master y stakeholders

El scrum master encargado de velar por que se siga las prácticas de scrum es Maycol Henrry Espinoza Ramírez. Los stakeholders son los alumnos, profesores y la directora de escuela de la Escuela Profesional de Obstetricia.

4.2.1.3 Equipo SCRUM

El equipo scrum está conformado por Maycol Henrry Espinoza Ramírez(Desarrollador Frontend) y Omar Cuba Estrella(Desarrollador Backend).

4.2.1.4 Creación del backlog priorizado del producto

Se crea el backlog(véase Tabla 12) con las épicas del negocio que luego se dividirán en historias de usuario.

Tabla 12 *Creación del backlog priorizado del producto*

BACKLOG PRIORIZADO DEL PRODUCTO

Permitir el inicio y cierre de sesión de los usuarios.

Gestionar asistencias de alumnos

Gestionar asistencias de profesores

Gestionar notas de alumnos

Permitir al alumno evaluar la clase impartida por el profesor(encuestas).

Contar con un área de anuncios sobre el curso del profesor.

Contar con un área de preguntas y respuestas sobre el curso entre profesores y alumnos.

Contar con un área de notificaciones

Permitir modificar las fórmulas de las notas (personalización por el profesor de un grupo)

Contar con estadísticas en base a las asistencias de las clases de alumnos y profesores.

Contar con estadísticas en base a las notas.

Contar con estadísticas en base a las encuestas

(Elaboración propia)

4.2.1.5 Realizar la planificación del lanzamiento

Cronograma de lanzamiento: El inicio del proyecto será el 1 de marzo de 2019 y la fecha de fin será el 31 de agosto de 2019, es decir tiene una duración aproximada de 6 meses.(De la Cruz, Espinoza, & Cuba, 2019)

Duración del Sprint: Existirá unsprint 0 para refinar los requerimientos y definir la arquitectura y los demás sprints están enumerados del 1 al 5. La duración de cada sprint debe ser de aproximadamente 1 mes (4 semanas), es posible que se pueda reducir el tiempo de la duración del sprint de acuerdo con la velocidad del equipo. (De la Cruz, Espinoza, & Cuba, 2019)

4.2.2 Análisis y planificación

4.2.2.1 Módulos

De acuerdo con el backlog priorizado se identificaron los siguientes módulos presentes en la siguiente imagen(véase Figura 66).



Figura 66. Módulos del sistema (Elaboración propia)

4.2.2.2 Product backlog del sistema

En esta fase se crea el product backlog (véaseTabla 13) por el dueño del producto es decir se realiza la creación de historias de usuario cada historia de usuario menciona a que módulo del sistema pertenece, cuenta con un id definido de la forma (“HU” +número correlativo de historia de usuario), posee peso de importancia definida por el producto owner, y estimación definida por el equipo SCRUM mediante el uso de la herramienta planningpoker.

También se consideraron incluir “spikes”los cuales se consideran como tareas adicionales que se realizan para realizar tareas de investigación, análisis o diseño las cuales se pueden incluir dentro de un sprint, cuentan con id definido de la forma (“SPIKE” + número correlativo de spike).

A diferencia de (De la Cruz, Espinoza, & Cuba, 2019), se aumentaron las historias de usuario para trabajarlas de una mejor forma y acomodarlas mejor de acuerdo con su tipo, lo cual se verá en detalle en el desarrollo de los sprints.

Tabla 13
Product backlog del sistema: pesos y estimaciones

MÓDULO	ID HU	Nombre HU	PE SO	ESTIMACIÓN
AUTENTICACIÓN	HU01	Iniciar sesión	3	5
	HU02	Escoger acceso	3	3
	HU03	Cerrar sesión	2	2
	HU04	Visualizar menú de opciones	3	3
PERÍODO ACADÉMICO	HU05	Ver cursos	3	3
	HU06	Ver grupos	3	4
CLASE	HU07	Ver clases de hoy	4	3
	HU08	Consultar listado de clases	4	4
	HU09	Consultar clase	4	4
		Guardar asistencias de	5	4
	HU10	Alumno de la clase		
	HU11	Finalizar clase	4	3
NOTAS	HU12	Iniciar clase	4	3
		Consultar notas como profesor	5	3
	HU13	Consultar notas como alumno	5	3
	HU14	Registrar notas	5	5
	HU15	Consultar promedio final de alumnos	3	3
	HU16			
COPARTICIPACIÓN	HU17	Consultar anuncios	3	3
	HU18	Registrar anuncio	3	3
	HU19	Consultar preguntas	3	3
	HU20	Consultar respuestas	3	3
	HU21	Registrar pregunta	4	2
	HU22	Registrar respuesta	3	5
NOTAS	HU23	Consultar fórmula	3	5
	HU24	Personalizar fórmula	2	3
	HU25	Volver a fórmula no personalizada	2	3
ENCUESTA	HU26	Encuestar clase	5	3
	HU27	Consultar promedio de encuesta por clase	2	2
NOTIFICACIÓN				
	HU28	Recibir cantidad de notificaciones	3	5
	HU29	Consultar notificaciones	4	2
REPORTES	HU30	Actualizar notificaciones	2	2
	HU31	Generar reporte asistencias	5	5
	HU32	Generar reporte notas	5	5
SPIKES	HU33	Generar reporte encuestas	5	5
	SPIKE01	Preparación de interfaces de	4	4

		usuario		
SPIKE02	Diseño de la arquitectura Backend	5	5	
SPIKE03	Diseño de la arquitectura Frontend	5	5	
SPIKE04	Preparar autenticación por OAuth mediante Token	5	5	
SPIKE05	Implementación de Websockets en ReactNative	4	4	

(Elaboración propia)

A continuación, se mostrará los enunciados de las historias de usuario y sus correspondientes criterios de aceptación para validarla. (véase Tabla 14)

Tabla 14

Product backlog: Descripción de historias de usuario (HU) y lista de criterios de aceptación (CA)

ID HU	Nombr e HU	Enunciado HU	#	Título CA	Criterio de aceptación (CA)
HU0 1	Iniciar sesión	Como usuario deseo iniciar sesión en el sistema para poder hacer uso de las funciones de éste.	1	Contraseña enmascarada	Dado que el usuario está en el ingreso de datos para autenticación cuando el usuario digite la contraseña entonces ésta debe de enmascararse con puntos(.)
			2	Autenticación inválida	Dado que el usuario ingresó sus datos de autenticación y que los datos son inválidos cuando el usuario indique iniciar sesión se debe mostrar un mensaje de credenciales inválidas.
			3	Autenticación válida	Dado que el usuario ingresó sus datos de autenticación y que los datos son válidos cuando el usuario indique iniciar sesión se debe mostrar los accesos disponibles para éste, si solo tuviera uno se dirigiría a la interfaz según su perfil con funciones como: notificaciones, menú de opciones y por defecto las clases de hoy (sólo alumno o profesor).
HU0 2	Escoger acceso	Como usuario deseo escoger el acceso deseado para poder acceder a las	1	Escoger acceso	Dado que el usuario se autenticó y que tuvo más de un acceso cuando el usuario seleccione un acceso se debe mostrar la interfaz correspondiente a su acceso con funciones como:

		funciones habilitadas del acceso.		notificaciones, menú de opciones y por defecto las clases de hoy (sólo alumno o profesor).
HU0 3	Cerrar sesión	Como usuario deseo cerrar sesión en el sistema para poder salir del aplicativo y tener restringido mis accesos cuando lo desee.	1 Indicar cierre de sesión 2 Confirmar cierre de sesión	Dado que el usuario esta autenticado y abra el menú de opciones cuando el usuario indique la opción cerrar sesión el sistema debe mostrar un mensaje de confirmación Dado que el usuario esta autenticado e indico la opción cerrar sesión cuando el usuario indique la opción cerrar sesión se debe cerrar la sesión y volver al formulario de autenticación
HU0 4	Visualizar menú de opciones	Como usuario deseo ver las opciones principales según el perfil que tenga para poder elegir a cuál de ellos ir.	1 Visualizar opciones	Dado que el usuario esta autenticado cuando indica ver opciones se debe listar las opciones disponibles según acceso que tuviese e incluir la opción "cerrar sesión".
HU0 5	Ver cursos	Como profesor/alumno deseo ver el listado de cursos en los que me encuentro inscrito con el fin de poder elegir que curso deseo consultar.	1 Ver cursos asignados (Profesor) 2 Ver cursos asignados (Alumno)	Dado que el usuario sea profesor cuando indique ver la opción ver cursos se debe mostrar los cursos que enseña ya sea un profesor responsable o designado Dado que el usuario sea alumno cuando indique ver la opción ver grupos se debe mostrar los cursos en los que está matriculado
HU0 6	Ver grupos	Como profesor/alumno deseo ver el listado de grupos de un curso en los que me encuentro inscrito con el fin de poder elegir que grupo deseo consultar.	1 Ver grupos asignados (Profesor) 2 Ver grupos asignados (Alumno)	Dado que el usuario sea profesor cuando indique ver la opción ver grupos del curso se debe mostrar los grupos que enseña ya sea un profesor responsable o designado Dado que el usuario sea alumno cuando indique ver la opción ver cursos se debe mostrar los grupos en los que está inscrito

HU07	Ver clases de hoy	Como profesor/alumno deseo ver las clases de hoy para poder ver que clases tengo pendientes de una forma ordenada	1 Ver clases de hoy	Dado que el usuario bien sea alumno o profesor por primera vez cuando éste haya entrado con su acceso se le mostrará las clases disponibles (nombre del tema, grupo, hora, curso) del día de hoy de forma ordenada de menor a mayor.
HU08	Consultar listado de clases	Como profesor/alumno deseo ver el listado de clases de un grupo para poder ver las clases que ya realice como las que faltan	1 Consultar listado de Clases	Dado que el usuario como profesor/alumno haya seleccionado un grupo cuando indique consultar listado de clases se debe mostrar todas las clases diferencias por estado (iniciada, finalizada, no iniciada) e información de la fecha y hora de inicio.
HU09	Consultar clase	Como profesor/alumno deseo consultar una clase para ver su información detallada de ésta.	1 Consultar clase (Profesor) 2 Filtrar alumnos 3 Consultar clase (Alumno)	<p>Dado que el usuario sea profesor y se le hayan listado las clases disponibles cuando éste consulte la clase se le debe mostrar información detallada como:</p> <p>La fecha de la clase, nombre del tema, profesor asignado, estado de la clase, hora de inicio, hora de fin, y la lista de asistencias de los alumnos(código, nombre y estado de asistencia), de acuerdo al estado de la clase se habilitaran las opciones: Iniciar clase(estado de clase no iniciada), Guardar asistencias(estado de clase iniciada), y Finalizar clase(estado de clase iniciada).</p> <p>Dado que el usuario sea profesor y se haya consultado la clase cuando éste requiera filtrar se le debe permitir filtrar los alumnos por su código o nombre de alumno.</p> <p>Dado que el usuario sea profesor y se le hayan listado las clases disponibles cuando éste consulte la clase se le debe mostrar información detallada como:</p> <p>La fecha de la clase, nombre del tema, profesor asignado, estado de la clase, estado de asistencia del alumno.</p>

HU1	0	Iniciar clase	Como profesor deseo iniciar la clase con el fin de marcar mi asistencia y la de mis alumnos.	1	Iniciar clase	Dado el usuario como profesor y se encuentra asignado a una clase y haya consultado una clase en estado no iniciada cuando éste indique iniciar clase se debe iniciar la clase con la fecha y hora actual, así como inicializar las asistencias de los alumnos por defecto a inasistencia.
HU1	1	Guardar asistencias de alumno de la clase	Como profesor deseo guardar asistencias de alumno de la clase con el fin de poder contar con esta información almacenada para luego consultarla.	1	Guardar asistencias	Dado el usuario como profesor y se encuentra asignado a una clase y haya consultado una clase en estado iniciada cuando éste indique guardar asistencias de alumno se deben guardar las asistencias de los alumnos modificada en la clase con su tipo de asistencia actualizada.
HU1	2	Finalizar clase	Como profesor deseo finalizar clase con el fin de dar por culminado la clase y poder ver otras clases que tuviera.	1	Finalizar clase	Dado el usuario como profesor y se encuentra asignado a una clase y haya consultado una clase en estado iniciada cuando éste indique finalizar clase se debe finalizar la clase con la fecha y hora actual.
HU1	3	Consultar notas como profesor	Como profesor deseo ver las notas de los alumnos con el fin de llevar un control adecuado de cada evaluación de los alumnos.	1	Listar fórmulas	Dado el usuario como profesor y ha elegido un grupo cuando desea consultar las notas del grupo se debe mostrar las fórmulas específicas, subespecíficas y la nota de la primera fórmula subespecífica
				2	Listar notas de alumnos	Dado el usuario como profesor y tiene fórmulas específicas y subespecíficas por elegir cuando éste elige fórmula específica y subespecífica se debe mostrar la relación de alumnos con todas sus notas si no tuviera por defecto debe ser 0.
				3	Filtrar alumnos	Dado que el usuario sea profesor y se han listado de notas cuando éste requiera filtrar se le debe permitir filtrar los alumnos por su código o nombre de alumno.

HU1	4	Consultar notas como Alumno	Como alumno deseo ver las notas de los alumnos con el fin de llevar un control de mis evaluaciones como el promedio de éstos.	1 Ver notas y promedio de alumno	Dado el usuario como alumno y se ha elegido un grupo cuando desea consultar sus notas y promedio se debe mostrar cada una de sus notas y su promedio de notas.
HU1	5	Registrar notas	Como profesor deseo registrar las notas con el fin de tener actualizado las notas y llevar un control adecuado.	1 Registrar nota	Dado el usuario como profesor y se ha consultado las notas cuando éste ingresa las notas e indica la opción guardar se debe registrar las notas cambiadas de la evaluación.
HU1	6	Consultar promedio final de alumnos	Como profesor deseo consultar el promedio ponderado final de los alumnos con el fin de evaluar el desempeño general en el curso de éstos	1 Listar alumnos 2 Consultar promedio final de alumno	Dado el usuario como profesor cuando éste consulta el promedio final de los alumnos se debe mostrar primero la relación de los alumnos Dado el usuario como profesor y se ha mostrado la relación de alumnos cuando éste el promedio final de un alumno se debe mostrar las notas del promedio final de cada grupo en el que está inscrito.
HU1	7	Consultar anuncios	Como profesor/anuncio deseo consultar los anuncios que eh publicado con el fin de revisarlos	1 Consultar anuncios	Dado el usuario como profesor/alumno y haya seleccionado un grupo cuando desee ver los anuncios se debe mostrar la lista de anuncios relacionadas al grupo elegida, ordenadas de mayor a menor fecha.
HU1	8	Registrar anuncio	Como profesor deseo registrar un anuncio para avisar a los alumnos sobre un acontecimiento.	1 Mostrar formulario de registro de anuncio 2 Registrar anuncio	Dado el usuario como profesor y haya seleccionado un grupo cuando desee registrar un anuncio se debe mostrar un formulario del registro de anuncio (título, descripción del anuncio). Dado el usuario como profesor y mostrado el formulario de registro de anuncio cuando haya ingresado el formulario y guardar lo se debe mostrar un mensaje de registro exitoso.

HU1	9	Consultar preguntas	Como profesor/alumno deseo consultar las preguntas que he publicado con el fin de revisarlas	1	Consultar preguntas	Dado el usuario como profesor/alumno y haya seleccionado un grupo cuando deseé ver las preguntas se debe mostrar la lista de preguntas relacionadas al grupo elegida ordenadas de mayor a menor fecha.
HU2	0	Consultar respuestas	Como profesor/alumno deseo consultar respuestas para obtener información de los demás sobre la pregunta.	1	Consultar respuestas	Dado el usuario como profesor/alumno y haya seleccionado un grupo cuando deseé ver las respuestas se debe mostrar la lista de respuestas relacionadas al grupo elegida ordenadas de mayor a menor fecha.
HU2	1	Registrar pregunta	Como profesor/alumno deseo registrar una pregunta para resolver una duda que tuviera.	1	Mostrar formulario de registro de pregunta	Dado el usuario como profesor y haya seleccionado una clase cuando deseé registrar una pregunta se debe mostrar un formulario del registro de pregunta (descripción de la pregunta).
				2	Registrar pregunta	Dado el usuario como profesor y mostrado el formulario de registro de pregunta cuando haya ingresado el formulario y guardarlo se debe mostrar un mensaje de registro exitoso.
HU2	2	Registrar respuesta	Como profesor/alumno deseo registrar una respuesta con el fin de contestar una pregunta y tratar de solucionar la duda.	1	Mostrar formulario de registro de respuesta	Dado el usuario como profesor y haya seleccionado una pregunta cuando deseé registrar una respuesta se debe mostrar un formulario del registro de respuesta (descripción de la respuesta).
				2	Registrar respuesta	Dado el usuario como profesor y mostrado el formulario de registro de respuesta cuando haya ingresado el formulario y guardarlo se debe mostrar un mensaje de registro exitoso.
HU2	3	Consultar fórmula	Como profesor/alumno deseo consultar las fórmulas de los grupos que estoy inscrito con el fin de	1	Consultar fórmulas específicas	Dado el usuario como el profesor/alumno y haya elegido un grupo cuando este consultando las fórmulas del grupo se debe mostrar las fórmulas específicas del grupo y el cálculo del promedio de las notas del grupo.

		saber cómo se calculan los promedios de notas	2 Consultar fórmulas subespecíficas	Dado el usuario como el profesor/alumno y haya consultado las fórmulas específicas cuando este consultando una fórmula específica en particular se debe mostrar las fórmulas subespecíficas del grupo.
HU2	4 Personas lizar fórmula	Como profesor deseo personalizar la fórmula porque deseo cambiar la fórmula general y registrar mi propia forma de evaluación	1 Dar mantenimiento a fórmulas personalizadas 2 Cambiar a fórmula personalizada 3 No ha realizado cambios	Dado el usuario como profesor y la fórmula sea de tipo personalizada cuando desee registrar datos se debe mostrar la información sobre la fórmula (peso, sigla y nombre) ya sea de tipo específica o subespecífica. Puede haber registro/modificación o eliminación de fórmulas personalizadas específicas o subespecíficas, se debe desvincular con la fórmula personalizada anterior y eliminar sus respectivas notas. Dado el usuario como profesor y la fórmula del grupo sea de tipo no personalizada cuando cambie la fórmula a fórmulas personalizada se debe permitir cambiar de fórmula no personalizada a personalizada tanto de las fórmulas específicas como las subespecíficas. Dado el usuario como profesor y haya consultado las fórmulas cuando indique cambiar fórmula y no haya realizado cambios se debe emitir un mensaje informativo indicando que no se hicieron cambios
HU2	5 Volver a fórmula no personaliza lizada	Como profesor deseo personalizar la fórmula porque deseo cambiar la fórmula general y registrar mi propia forma de evaluación	1 Volver a fórmula no personalizada	Dado el usuario como profesor y la fórmula sea de tipo personalizada cuando indique cambiar la fórmula personalizada a personalizada se debe volver a la fórmula anterior y eliminar la fórmula personalizada anterior con sus respectivas notas.
HU2	6 Encuest ar clase	Como alumno deseo encuestar clase con el fin de poder evaluar la enseñanza y	1 Ver preguntas de encuesta de la clase	Dado el usuario como alumno y haya consultado una clase en estado iniciada o finalizada cuando éste haya consultado la clase se le debe mostrar una encuesta de la clase.

		el aprendizaje en la clase.	2	Encuestar clase	Dado el usuario como alumno y haya visto la encuesta de la clase y la ha llenado cuando éste indique registrar se debe registrar la encuesta y emitir un mensaje informativo.
HU2	Consultar promedio de encuesta por clase	Como profesor, alumno deseo consultar el promedio de encuesta por clase con el fin de poder ver cómo fue de buena o mala la clase según los alumnos.	1	Consultar promedio en el listado de clases	Dado el usuario como profesor/alumno cuando éste haya entrado a ver la lista de clases se debe ver la información de las clases y el promedio de cada una de éstas.
			2	Consultar promedio en el detalle de la clase	Dado el usuario como profesor/alumno cuando éste haya consultado una clase se debe ver la información de la clase y el promedio de la clase.
HU2	Recibir cantidad de notificaciones	Como usuario deseo recibir la cantidad actual de notificaciones no vistas que tengo esto con la finalidad de enterarme cuantos nuevos acontecimientos que me son de mi interés han ocurrido.	1	Registrar notificaciones de asistencias	Dado el usuario como profesor cuando registre una asistencia de la clase se debe llegar una notificación a los alumnos.
			2	Registrar notificaciones de notas	Dado el usuario como profesor cuando registre las notas de los alumnos le debe llegar a los alumnos interesados
			3	Consultar número de notificaciones	Dado el usuario que haya sido autenticado en el sistema cuando éste ingrese o se encuentre en la plataforma y tenga nuevas notificaciones se debe mostrar la cantidad de nuevas notificaciones que no ha visto.
HU2	Consultar notificaciones	Como usuario deseo recibir la cantidad actual de notificaciones no vistas que tengo esto con la finalidad de enterarme cuales son los nuevos acontecimientos que me son de mi interés han ocurrido.	1	Consultar últimas notificaciones	Dado el usuario que haya sido autenticado en el sistema cuando indique ver notificaciones se debe mostrar las últimas 8 notificaciones ordenados de fecha de mayor a menor.
			2	Consultar todas las notificaciones	Dado el usuario que haya sido autenticado en el sistema cuando indique ver todas las notificaciones se debe mostrar todas las notificaciones ordenados de fecha de mayor a menor y con un estado (no vistas, vistas).
HU3	Actualizar las notifica	Como usuario deseo actualizar las	1	Actualizar las notificaciones	Dado el usuario que haya sido autenticado al sistema y haya consultado las notificaciones cuando

		ciones	notificaciones que consulte como ya vistas con el fin de que pueda diferenciar las que no consulte de las que consulte.	es como vistas	éste las revise se debe actualizar las notificaciones como ya vistas.
HU3	1	Generar reporte asistencias	Como director de escuela deseo ver el reporte de asistencias con el fin de realizar el control de asistencias de los cursos.	1 Filtrar búsqueda 2 Mostrar estadísticas	Como director de escuela cuando haya ingresado a la sección de reportes de asistencias se debe poder filtrar por curso: (todos los cursos o un curso en particular), por grupo (todos los cursos o uno en particular), por alumno (todos los alumnos o uno en particular) o si es por profesor (todos los profesores o uno en particular) Como director de escuela y haya realizado los filtros de búsqueda deseado cuando desee ver estadísticas se debe mostrar un cuadro estadístico de las asistencias (a tiempo, tardanzas, faltas y la cantidad restante) y un gráfico de barras que indique cuantas asistencias faltan concretar ya sea por alumno o por profesor.
HU3	2	Generar reporte notas	Como director de escuela deseo ver el reporte de asistencias con el fin de realizar el control de notas de los cursos.	1 Filtrar búsqueda 2 Mostrar estadísticas	Como director de escuela cuando haya ingresado a la sección de reportes de notas se debe poder filtrar por curso: (todos los cursos o un curso en particular), por grupo (todos los cursos o uno en particular), por alumno (todos los alumnos o uno en particular) Como director de escuela y haya realizado los filtros de búsqueda deseado cuando desee ver estadísticas se debe mostrar un cuadro estadístico de las notas (del 0 al 5, 6 al 10, 11 al 15 y 16 al 20) y un gráfico de barras que indique cuantas notas faltan registrar por los profesores.

HU3	Generar reporte encuest 3	Como director de escuela deseo ver el reporte de asistencias con el fin de realizar el control en base a encuestas de clases de los cursos.	1 Filtrar búsqueda 2 Mostrar estadísticas	Como director de escuela cuando haya ingresado a la sección de reportes de encuestas se debe poder filtrar por curso: (todos los cursos o un curso en particular), por grupo (todos los cursos o uno en particular), por profesor (todos los profesores o uno en particular) Como director de escuela y haya realizado los filtros de búsqueda deseado cuando desee ver estadísticas se debe mostrar el promedio de todas las encuestas en estrellas, y la relación de preguntas cada una con el promedio de estrellas de encuestas.
------------	---------------------------	---	--	---

(Elaboración propia)

4.2.2.3 Sprint0

Fecha de inicio: 1 de marzo del 2019

Fecha de fin: 31 de marzo del 2019

Planificación:

Este sprint(véase Tabla 15) es una fase previa a los demás sprintsse diferencia en que no realizará entregables de software con valor al cliente, sino que servirá para tener más claro el análisis de las historias de usuario, la arquitectura frontend, backend y ambientes que se usarán para los siguientes sprints. El 1 de marzo del 2019 se realizó la planificación de este sprint con duración alrededor de 4 horas. La duración del sprint se estimó en 1 mes aproximadamente. Así mismo se realizó la estimación de las tareas de los spikescon el equipo SCRUM mediante la técnica planningpoker. El avance fue de lunes a viernes de 9 a 5 pm y las reuniones diarias para revisar los avances del sprint duraba alrededor de 15 minutos por Skype.

Tabla 15
Sprint 0 del sistema

ID HU	NOMBRE HU	#	TAREA	COMPRO METIDO A:	EST IMA CIÓ N
SPIKE01	Preparació n deinterface s de usuario	1 2 3	Creación de prototipos para el acceso del alumno Creación de prototipos para el acceso del profesor Creación de prototipos para el acceso de la directora de escuela	OMAR CUBA OMAR CUBA OMAR CUBA	4 4 3

SPIKE02	Diseño de la arquitectura Backend	1	Establecer lineamientos, tales como patrones de diseño, buenas prácticas, principios SOLID, entre otros en la arquitectura de microservicios.	OMAR CUBA	5
		2	Identificación y definición de los componentes la arquitectura de microservicios.	OMAR CUBA	5
		3	Diseño del esquema de bases de datos para cada uno de los microservicios	OMAR CUBA	5
		4	Implementación de las infraestructuras de desarrollo y pruebas en repositorio con GIT.	OMAR CUBA	2
		5	Generación de script (archivo bat) que inicialice por consola los microservicios de forma automatizada.	MAYCOL ESPINOZA	2
SPIKE03	Diseño de la arquitectura Frontend	1	Creación y diseño del proyecto Frontend basado en ReactNative con el IDE WebStormJetBrains	MAYCOL ESPINOZA	5
		2	Implementación e integración de la infraestructura de desarrollo y pruebas.	MAYCOL ESPINOZA	2
		3	Integración con el simulador o dispositivo móvil a través de la herramienta Expo App.	MAYCOL ESPINOZA	3
SPIKE04	Preparar autenticación por Oauth mediante tokens	1	Implementación de Spring Oauth 2.0 en el un servicio que a su vez sea transversal a todos los microservicios (AuthorizationServer)	MAYCOL ESPINOZA	5
		2	Creación del endpointOauth/token	MAYCOL ESPINOZA	4
		3	Implementación del endpointOauthrefresh token	MAYCOL ESPINOZA	4
		4	Implementación de los receptores de autenticación para los restantes microservicios donde se encuentran los recursos (ResourceServer)	MAYCOL ESPINOZA	4
			(Elaboración propia)		

4.2.2.4 Sprint 1

Fecha de inicio: 1 de abril del 2019

Fecha de fin: 30 de abril del 2019

Planificación:

Este es el primer sprint que entrega valor al cliente (véase Tabla 16). El 1 de abril del 2019 se realizó la planificación de este sprint con duración alrededor de 4 horas. La duración del sprint se estimó en 1 mes aproximadamente. Así mismo se realizó la estimación de las tareas con el equipo SCRUM mediante la técnica planningpoker y se comprometió al equipo scrum para su desarrollo. Abarca todas las historias de usuario del módulo de autenticación y una historia de usuario del módulo de asistencia. El avance se planeó de lunes a viernes de 9 a 5 pm y las reuniones diarias para revisar los avances del sprint contarían con una duración aproximada de 15 minutos online por Hangouts.

Tabla 16
Sprint 1 del sistema

ID HU	NOMBR E HU	#	TAREA	COMPROMETID O A:	ESTI -MA- CIÓ N
HU01	Iniciar sesión	1	Creación del esquema, tablas, constraints, queries correspondientes a la base de datos autenticación. (Backend)	MAYCOL ESPINOZA	3
		2	Llenado inicial de datos de prueba de la tabla usuarios. (Backend)	OMAR CUBA	2
		3	Creación de microservicio de autenticación: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	MAYCOL ESPINOZA	4
		4	Desarrollo de la API "Iniciar sesión" en el microservicio autenticación. (Backend)	MAYCOL ESPINOZA	5
		5	Integración de Oauth 2 en el microservicio autenticación. (Backend)	MAYCOL ESPINOZA	4
		6	Desarrollo de la GUI "Iniciar sesión" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	5
		7	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		8	Realización de pruebas en base a los criterios de aceptación (Frontend)	MAYCOL ESPINOZA	2

HU02	Escoger acceso	1	Llenado inicial de datos de prueba de las tablas roles y perfiles. (Backend)	OMAR CUBA	2
		2	Desarrollo de la GUI "Escoger acceso" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	3
		3	Realización de pruebas en base a los criterios de aceptación. (Frontend)	OMAR CUBA	2
HU03	Cerrar sesión	1	Desarrollo del API "Cerrar sesión" en el microservicio autenticación. (Backend)	MAYCOL ESPINOZA	4
		2	Desarrollo de la funcionalidad "Cerrar sesión" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	3
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	OMAR CUBA	2
HU04	Visualizar menú de opciones	1	Acoplar el listado de opciones (roles) con las que cuenta cada usuario (perfil) dentro del servicio encargado de generar los tokens. (Frontend)	MAYCOL ESPINOZA	4
		2	Desarrollo del menú lateral izquierdo, el cual permite visualizar las opciones disponibles para el usuario autenticado. Por defecto oculta, y que al pasar el dedo sobre el margen izquierdo se muestre nuevamente. (Frontend)	MAYCOL ESPINOZA	5
		3	Definición del esquema de navegabilidad para el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	5
		4	Llenado inicial de datos de los roles correspondientes a cada usuario según su perfil en la base de datos autenticación. (Backend)	OMAR CUBA	2
		5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	OMAR CUBA	2
HU05	Ver cursos	1	Creación del esquema, tablas, constraints, queries correspondientes a la base de datos periodo-académico.	OMAR CUBA	4

		(Backend)		
	2	Llenado inicial de datos de prueba de tablas como cursos, persona (alumnos y profesores), Matriculado. (Backend)	OMAR CUBA	3
	3	Creación de microservicio de periodo-académico: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	OMAR CUBA	3
	4	Desarrollo de la API "Ver cursos" en el microservicio periodo-académico. (Backend)	OMAR CUBA	4
	5	Integración de Oauth 2 en el microservicio periodo-académico. (Backend)	OMAR CUBA	2
	6	Desarrollo de la API "Ver cursos" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	4
	7	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
	8	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	3
HU06	Ver grupos	1 Llenado inicial de datos de tablas como AgrupacionCurricular, DetalleGrupo del microservicio periodo-académico. (Backend)	OMAR CUBA	5
		2 Desarrollo de la API "Ver grupos" en el microservicio periodo-académico. (Backend)	OMAR CUBA	5
		3 Desarrollo de la GUI "Ver grupos" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	4
		4 Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		5 Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	3
		(Elaboración propia)		

Revisión: En la fecha 30 de abril de 2019 se dió la revisión con la dueña del producto Clara Díaz Tinoco la cual consistió en la validación de los criterios de aceptación de las

historias de usuario definidas en el sprint. La duración de la reunión fue de 3 horas aproximadamente.

Retrospectiva: En la fecha 30 de abril de 2019 se dio la retrospectiva entre los integrantes del equipo scrum. La duración de la reunión fue de 3 horas. Se analizaron los siguientes puntos:

- Ambos avanzamos la arquitectura de microservicios inicialmente, pero luego decidimos que sería mejor separar el backend y el frontend así veríamos más rápidos los avances.
- Al iniciar el desarrollo tanto en el lado backend como frontend del proyecto tuvimos complicaciones con el manejo de las tecnologías debido a que necesitábamos capacitarnos.
- Hubo complicaciones en el diseño de la arquitectura ya que se necesitaba tener una buena abstracción para ver cómo se separaban los microservicios de acuerdo con la lógica de negocio.
- Algunas interfaces de usuario debieron ser reajustados para cumplir los criterios aceptación.
- La coordinación es un factor muy importante para que se trabaje en armonía es importante compartir y/o notificar sobre cambios, recursos o conocimientos que otro integrante del equipo scrum desconoce.
- El microservicio periodo-académico contiene información útil para otros microservicios esto es importante ya que puede ser invocado por otros microservicios para realizar validaciones.

4.2.2.5 Sprint 2

Fecha de inicio: 1 de mayo del 2019

Fecha de fin: 31 de mayo del 2019

Planificación:

El 1 de mayo del 2019 se realizó la planificación de este sprint(véase Tabla 17) con duración alrededor de 4 horas. Así mismo se realizó la estimación de las tareas con el equipo SCRUM mediante la técnica planningpoker y se comprometió al equipo scrum para su desarrollo. Abarca todas las historias de usuario del módulo de asistencias e historias de usuario del módulo de encuestas. El avance se planeó de lunes a viernes de 9 a 5 pm y las reuniones diarias para revisar los avances del sprint contaría una duración aproximada de 15 minutos presencial.

Tabla 17

Sprint 2 del sistema

ID HU	NOMBRE HU	#	TAREA	COMPROMETIDO A:	ESTIMA- CIÓN
HU07	Ver clases de hoy	1	Creación del esquema, tablas, constraints, queries correspondientes a la base	OMAR CUBA	3

		de datos de asistencia. (Backend)		
	2	Llenado inicial de datos de prueba de la tabla clase del microservicio periodo-académico y de la tabla asistencia_clase	OMAR CUBA	2
	3	Creación de microservicio de asistencia: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	OMAR CUBA	3
	4	Desarrollo de la API "Ver clase de hoy" en el microservicio asistencia. (Backend)	OMAR CUBA	4
	5	Creación del servicio Composición que permite combinar los otros servicios. Desarrollo de API Ver clases de hoy en el servicio composición, además de la información de las clases de hoy se debe ver estado de clase (microservicio periodo académico y microservicio asistencias). (Backend)	OMAR CUBA	4
	6	Desarrollo de la GUI "Ver clase de hoy" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	4
	7	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
	8	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU08	Consultar listado de clases	1 Desarrollo de la API "Ver lista de clases por grupo" en el microservicio asistencia. (Backend) 2 Desarrollo de API Listado de clases en la consulta de la lista de clases de un grupo en el servicio	OMAR CUBA	4 3

		Composición. Se debe listar las clases y agregar estado de clase (microservicio periodo académico y microservicio asistencias). (Backend)		
	3	Desarrollo de la GUI "Listar clases por grupo" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	5
	4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
	5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU09	Consultar clase	1 Llenado de datos de prueba de la tabla asistencia_clase y asistencia_alumno. (Backend)	OMAR CUBA	2
	2	Modificar la API "Ver clases de hoy" del servicio Composición que usa a su vez los microservicios periodo-académico y asistencia con el fin de que traiga más información de cada clase y las asistencias del alumno para que se pueda realizar la funcionalidad "Consultar clase". (Backend)	OMAR CUBA	4
	3	Desarrollo de la GUI "Consulta de clase" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	5
	4	Asegurar el comportamiento responsivo para ambas partes del acordeón (Detalle de clase y listado de asistencias) (Frontend)	MAYCOL ESPINOZA	5
	5	Implementar las "go backs" (regresar atrás Android) en todas las vistas realizadas hasta el momento. (Frontend)	MAYCOL ESPINOZA	5

		6	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		7	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU10	Iniciar clase	1	Desarrollo de la funcionalidad "Iniciar clase" en el microservicio asistencia. (Backend)	OMAR CUBA	5
		2	Desarrollo de la funcionalidad "Iniciar clase" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	1
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU11	Guardar asistencias de alumno de la clase	1	Desarrollo de la API "Guardar asistencias de alumno" en el microservicio asistencia. (Backend)	OMAR CUBA	5
		2	Desarrollo de la GUI "Guardar asistencias de alumno" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	1
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU12	Finalizar clase	1	Desarrollo de la funcionalidad "Finalizarclase" en el microservicio asistencia. (Backend)	OMAR CUBA	5
		2	Desarrollo de la funcionalidad "Finalizarclase" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	1
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2

		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU26 Encuestar clase	1	Creación del esquema, tablas, constraints, queries correspondientes a la base de datos de encuesta (Backend)	OMAR CUBA	5	
	2	Creación de microservicio de encuesta: alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	OMAR CUBA	4	
	3	Desarrollo de la API "Realizar encuesta" en el microservicio asistencia. (Backend)	OMAR CUBA	5	
	4	Desarrollo de la GUI "Realizar encuesta" en el aplicativo móvil. (Frontend)	MAYCOL ESPINOZA	4	
	5	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2	
	6	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2	
HU27 Consultar promedio de encuesta por clase	1	Desarrollo del API "Consultar promedio de encuesta por clase" en el microservicio encuesta. (Backend)	OMAR CUBA	5	
	2	Usar el servicio composición para combinar las clases (microservicio periodo-académico) con el promedio de encuestas (microservicio encuesta) (Backend)	OMAR CUBA	4	
	3	Agregar promedio de encuesta a la clase. (Frontend)	MAYCOL ESPINOZA	1	
	4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2	

5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
(Elaboración propia)			

Revisión: En la fecha 31 de mayo de 2019 se dio la revisión con la dueña del producto Clara Díaz Tinoco la cual consistió en la validación de los criterios de aceptación de las historias de usuario definidas en el sprint. La duración de la reunión fue de 3 horas aproximadamente.

Retrospectiva: En la fecha 31 de mayo de 2019 se dio la retrospectiva entre los integrantes del equipo scrum. La duración de la reunión fue de 3 horas. Se analizaron los siguientes puntos:

- Se redujo la complejidad en poder usar las tecnologías ya que adquirimos más práctica, nos costó menos esfuerzo en comparación con el primer sprint.
- Se culminó con la parte de acceso al sistema para los alumnos, profesores y directora de escuela.
- Acordamos que debemos mantener la coordinación ya que es importante en este tipo de proyectos en lo que respecta a base de datos servicios y la parte visual cuáles son sus usos de cada uno.
- El servicio composición fue útil para combinar la información de microservicios asistencia y periodo-académico y así permitir que cada uno siga aislado es decir trabajo de forma independiente.
- Los datos de prueba ficticios como la fecha actual y adaptación del servicio a consulta de fecha fueron muy útiles.

4.2.2.6 Sprint 3

Fecha de inicio: 1 de junio del 2019

Fecha de fin: 30 de junio del 2019

Planificación:

El 1 de junio del 2019 se realizó la planificación de este sprint (véase Tabla 18) con duración alrededor de 4 horas. Así mismo se realizó la estimación de las tareas con el equipo SCRUM mediante la técnica planningpoker y se comprometió al equipo scrum para su desarrollo. Abarca todas las historias de usuario del módulo de notas e historias de usuario del módulo de notificaciones. El avance se planeó de lunes a viernes de 9 a 5 pm y las reuniones diarias para revisar los avances del sprint contaría con una duración aproximada de 15 minutos de forma presencial.

Tabla 18

Sprint 3 del sistema

ID	HU	NOMBRE HU	#	TAREA	COMPROMETIDO A:	ESTIMACIÓN
----	----	-----------	---	-------	-----------------	------------

HU13	Consultar notas como profesor	1	Creación del esquema, tablas, constraints, queries correspondientes a la base de datos del microservicio nota (Backend)	OMAR CUBA	3
		2	Llenado inicial de datos de prueba de las tablas fórmula, distribucion_fórmula, nota_alumno del microservicio nota. (Backend)	OMAR CUBA	2
		3	Creación de OMAR CUBA microservicio nota: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	OMAR CUBA	2
		4	Desarrollo del API OMAR CUBA Consultar fórmulas específicas en el microservicio nota. (Backend)	OMAR CUBA	4
		5	Desarrollo del API OMAR CUBA Consultar notas en el microservicio nota. (Backend)	OMAR CUBA	3
		6	Desarrollo de la MAYCOL GUI listar fórmulas ESPINOZA específicas y subfórmulas específicas según el grupo seleccionado. (Frontend)	MAYCOL ESPINOSA	4
		7	Desarrollo de API OMAR CUBA Consultar notas por alumnos en el servicio composición. (Backend)	OMAR CUBA	3
		8	Desarrollo de la GUI sobre listar notas de alumnos según la fórmula	MAYCOL ESPINOSA	5

			seleccionada. (Frontend)		
		9	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		10	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU14	Consultar notas como alumno	1	Desarrollo del API buscar fórmulas y notas por alumno en el microservicio nota. (Backend)	OMAR CUBA	4
		2	Desarrollo de la GUI para mostrar las fórmulas y notas del alumno por grupo seleccionado. (Frontend)	MAYCOL ESPINOZA	3
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU15	Registrar notas	1	Desarrollo del API dar mantenimiento a notas en el microservicio nota. (Backend)	OMAR CUBA	5
		2	Desarrollo de la GUI para guardar notas. (Frontend)	MAYCOL ESPINOZA	2
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU16	Consultar promedio final de alumnos	1	Desarrollo del API "Buscar notas generales" en el microservicio nota, el cual se encarga de calcular las notas	OMAR CUBA	5

			promedio por alumno. (Backend)		
		2	Desarrollar la GUI para mostrar la relación de alumnos al consultar los promedios. (Frontend)	MAYCOL ESPINOZA	4
		3	Desarrollar la GUI para mostrar sus notas promedias por cada grupo inscrito de un alumno. (Frontend)	MAYCOL ESPINOZA	4
		4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU28	Recibir cantidad de notificaciones	1	Creación del esquema, tablas, constraints, queries correspondientes a la base de datos del microservicio notificación. (Backend)	OMAR CUBA	3
		2	Llenado inicial de datos de prueba de la tabla notificación del microservicio notificación. (Backend)	OMAR CUBA	2
		3	Creación de OMAR CUBA microservicio notificación: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	OMAR CUBA	2
		4	Desarrollo del websocket para guardar la cantidad de notificaciones por usuario autenticado en	OMAR CUBA	5

			el microservicio notificación. (Backend)		
		5	Permitir el registro de notificaciones en el microservicio notificaciones. (Llamada asíncrona por colas usando RabbitMQ) (Backend)	OMAR CUBA	3
		6	Desarrollo de invocaciones para registrar las notificaciones en los microservicios asistencia, nota. (Llamada asíncrona por colas usando RabbitMQ) (Backend)	OMAR CUBA	4
		7	Desarrollo de opción para consultar número de notificaciones, como la integración con websockets. (Frontend)	MAYCOL ESPINOZA	5
		8	Realización de pruebas con programa de ejemplo. (Backend)	OMAR CUBA	3
		9	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU29	Consultar notificaciones	1	Desarrollo del API listar notificaciones. (Backend)	OMAR CUBA	3
		2	Desarrollo del API listar últimas notificaciones. (Backend)	OMAR CUBA	1
		3	Desarrollo de la GUI para listar las ultimas notificaciones. (Frontend)	MAYCOL ESPINOZA	4
		4	Desarrollo de interfaz funcional para mostrar todas las notificaciones.	MAYCOL ESPINOZA	3

			(Frontend)	
		5	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA 2
		6	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA 2
HU30	Actualizar notificaciones	1	Desarrollo del API para actualizar notificaciones en el microservicio notificación. (Backend)	OMAR CUBA 3
		2	Desarrollo de funcionalidad para actualizar notificaciones al consultarlas. (Frontend)	MAYCOL ESPINOZA 4
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA 2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA 2
SPIKE 005	Averiguar websockets	1	Averiguar sobre implementación de websockets en servicios rest. (Backend)	OMAR CUBA 5
		2	Realizar integraciones de ejemplo. (Backend)	OMAR CUBA 4
		3	Integración de api gateway con websocket. (Backend)	OMAR CUBA 3
		4	Averiguar sobre implementación de websockets en ReactNative. (Frontend)	MAYCOL ESPINOZA 5
		5	Realizar integraciones de ejemplo. (Frontend)	MAYCOL ESPINOZA 4

(Elaboración propia)

Revisión: En la fecha 30 de junio de 2019 se dio la revisión con la dueña del producto Clara Díaz Tinoco la cual consistió en la validación de los criterios de aceptación de las historias de usuario definidas en el sprint. La duración de la reunión fue de 3 horas aproximadamente.

Retrospectiva: En la fecha 30 de junio de 2019 se dio la retrospectiva entre los integrantes del equipo scrum. La duración de la reunión fue de 3 horas. Se analizaron los siguientes puntos:

- Ya no fue complejo el desarrollo usando las tecnologías, ya que ya sabíamos los posibles errores y cómo abordarlos.
- Se avanzó con mayor rapidez.
- Se logró culminar con el alcance previsto.
- Se hizo uso de integración con colas usando RabbitMQ lo cual permitía las llamadas asíncronas entre servicios.
- Se tuvo que incluir un spike de investigación de websocketslo cual ayudo al desarrollo de las historias de usuario relacionadas a notificaciones.

4.2.2.7 Sprint 4

Fecha de inicio: 1 de julio del 2019

Fecha de fin: 21 de julio del 2019

Planificación:

El 1 de julio del 2019 se realizó la planificación de este sprint (véase Tabla 19) con duración alrededor de 4 horas. La duración del sprint se estimó en 3 semanas aproximadamente. Así mismo se realizó la estimación de las tareas con el equipo SCRUM mediante la técnica planningpoker y se comprometió al equipo scrum para su desarrollo. Abarca todas las historias de usuario del módulo de reportes y algunas historias de usuario del módulo de coparticipación. El avance se planeó de lunes a viernes de 9 a 5 pm y las reuniones diarias para revisar los avances del sprint contaría con una duración aproximada de 15 minutos en línea por Hangouts.

Tabla 19

Sprint 4 del sistema

ID HU	NOMBRE HU	#	TAREA	COMPROMETIDO A:	ESTI- MA- CIÓN
HU31	Generar reporte asistencias	1	Creación del esquema, tablas, constraints, queries correspondientes del microservicio reporte. (Backend)	OMAR CUBA	2
		2	Llenado inicial de la base de datos del microservicio reporte la cual proviene de las	OMAR CUBA	2

			demás bases de datos. (Backend)		
			Creación de OMAR CUBA microservicio reporte: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	3	
			Crear método para OMAR CUBA actualizar las tablas de asistencia en la base de datos del microservicio reporte mediante eventos usando RabbitMQ. (Backend)	3	
			Crear método en el OMAR CUBA microservicio asistencia que permita actualizar las asistencias en los reportes mediante eventos usando RabbitMQ. (Backend)	3	
			Desarrollo del API MAYCOL "Generar reportes de ESPINOZA asistencias" para alumnos y profesores, por todos los grupos o un grupo determinado. (Backend)	5	
			Desarrollo de la GUI para MAYCOL mostrar el reporte de ESPINOZA asistencia (Frontend)	5	
			Realización de pruebas OMAR CUBA con programa Postman. (Backend)	2	
			Realización de pruebas MAYCOL en base a los criterios de ESPINOZA aceptación. (Frontend)	2	
HU32	Generar reporte notas	1	Crear método para OMAR CUBA actualizar las tablas de notas en la base de datos del microservicio reporte mediante eventos usando RabbitMQ. (Backend)	3	
		2	Crear método en el OMAR CUBA microservicio nota que permite actualizar las asistencias en los reportes mediante eventos usando	3	

RabbitMQ. (Backend)

		3	Desarrollo del API "Generar reportes de notas" para alumnos y profesores, por todos los grupos o un grupo determinado. (Backend)	MAYCOL ESPINOZA	5
		4	Desarrollar de la GUI para mostrar el reporte de notas. (Frontend)	MAYCOL ESPINOZA	5
		5	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		6	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU33	Generar reporte encuestas	1	Crear método para actualizar las tablas de encuesta en la base de datos del microservicio reporte mediante eventos usando RabbitMQ. (Backend)	OMAR CUBA	3
		2	Crear método en el microservicio encuesta que permita actualizar las encuestas en los reportes mediante eventos usando RabbitMQ. (Backend)	OMAR CUBA	3
		3	Desarrollo del API "Generar reportes de encuestas" de acuerdo con las preguntas realizadas por todos los grupos o un grupo determinado. (Backend)	MAYCOL ESPINOZA	5
		4	Desarrollo de la GUI para mostrar el reporte de notas. (Frontend)	MAYCOL ESPINOZA	5
		5	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		6	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2

HU17	Consultar anuncios	1	Creación del esquema, tablas, constraints, queries correspondientes del microservicio coparticipación. (Backend)	OMAR CUBA	2
		2	Llenado inicial de la base de datos del microservicio coparticipación. (Backend)	OMAR CUBA	2
		3	Creación de microservicio coparticipación: Alojamiento de código (Git), conexión base de datos, mapeo de tablas, configuración de microservicio. (Backend)	OMAR CUBA	2
		4	Desarrollo del API "Consultar anuncios" en el microservicio coparticipación. (Backend)	OMAR CUBA	2
		5	Desarrollo de API "Consultar anuncios", que combina consulta de anuncios con las personas(periodo-académico) en el servicio composición. (Backend)	OMAR CUBA	2
		6	Desarrollo de la GUI para consultar anuncios. (Frontend)	MAYCOL ESPINOZA	4
		7	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		8	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
		1	Desarrollo del API "Registrar anuncio" en el microservicio coparticipación. (Backend)	OMAR CUBA	2
HU18	Registrar anuncio				

2	Desarrollo de método que combina consulta de anuncios (microservicio coparticipación) con datos del curso y los profesores (microservicio periodo-académico) en el servicio composición. (Backend)	OMAR CUBA	2
3	Desarrollo de la GUI para consultar anuncios. (Frontend)	MAYCOL ESPINOZA	4
4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2

(Elaboración propia)

Revisión: En la fecha 21 de julio de 2019 se dio la revisión con la dueña del producto Clara Díaz Tinoco la cual consistió en la validación de los criterios de aceptación de las historias de usuario definidas en el sprint. La duración de la reunión fue de 3 horas aproximadamente.

Retrospectiva: En la fecha 21 de julio de 2019 se dio la retrospectiva entre los integrantes del equipo scrum. La duración de la reunión fue de 3 horas. Se analizaron los siguientes puntos:

- La duración del sprint disminuyó y fue alrededor de 3 semanas menor a sus antecesoras que fueron 1 mes.
- Se comprobó que los reportes fueron importantes para el valor agregado como lo era esperado.
- Hubo que replicar mediante eventos la información de los otros microservicios debido a se necesitaba rapidez en las consultas de los reportes.
- Se uso gráficas estadísticas y barras como filtros en los reportes ya que simplificaban y mejoraban la interpretación de la información.

4.2.2.8 Sprint 5

Fecha de inicio: 22 de julio del 2019

Fecha de fin: 13 de agosto del 2019

Planificación:

El 22 de julio del 2019 se realizó la planificación de este sprint (véase Tabla 20) con duración alrededor de 4 horas. La duración del sprint se estimó en 3 semanas aproximadamente. Así mismo se realizó la estimación de las tareas con el equipo SCRUM mediante la técnica planningpoker y se comprometió al equipo scrum para su

desarrollo. Abarca las historias de usuario restantes del módulo de coparticipación (preguntas y respuestas) e historias de usuario del módulo de notas (fórmulas). El avance se planeó de lunes a viernes de 9 a 5 pm y las reuniones diarias para revisar los avances del sprint contarían con una duración aproximada de 15 minutos en línea por Hangouts.

Tabla 20
Sprint 5 del sistema

ID HU	NOMBRE HU	#	TAREA	COMPROMETIDO A:	ESTIMACIÓN
HU19	Consultar preguntas	1	Desarrollo del API “Consultar preguntas” en el microservicio coparticipación. (Backend)	OMAR CUBA	2
		2	Desarrollo de método que combina consulta de preguntas (microservicio coparticipación) con las personas, grupo y curso(periodo-académico) en el servicio Composición. (Backend)	OMAR CUBA	2
		3	Desarrollo de la GUI para consultar preguntas. (Frontend)	MAYCOL ESPINOZA	4
		4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU20	Consultar respuestas	1	Desarrollo del API “Consultar respuestas” en el microservicio coparticipación. (Backend)	OMAR CUBA	2
		2	Desarrollo de método que combina consulta de preguntas (microservicio coparticipación) con las personas, grupo y curso(periodo-académico) en el servicio Composición.	OMAR CUBA	2

(Backend)

		3	Desarrollo de la GUI para consultar preguntas. (Frontend)	MAYCOL ESPINOZA	4
		4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU21	Registrar pregunta	1	Desarrollo del API “Registrar pregunta” en el microservicio coparticipación. (Backend)	OMAR CUBA	2
		2	Invocación de notificaciones con eventos mediante uso de RabbitMQ. (Backend)	OMAR CUBA	2
		3	Desarrollar interfaz de usuario para registrar pregunta. (Frontend)	MAYCOL ESPINOZA	4
		4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU22	Registrar respuesta	1	Desarrollo del API “Registrar respuesta” en el microservicio coparticipación. (Backend)	OMAR CUBA	2
		2	Invocación de notificaciones con eventos mediante uso de RabbitMQ(Backend)	OMAR CUBA	2
		3	Desarrollo de la GUI para registrar pregunta. (Frontend)	MAYCOL ESPINOZA	4
		4	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		5	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2

HU23	Consultar fórmula	1	Desarrollar de la GUI para consultar fórmula específicas y subespecíficas por grupo seleccionado. (Frontend)	MAYCOL ESPINOZA	5
		2	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU24	Personalizar fórmula	1	Desarrollo del API “Dar mantenimiento a fórmula” en el microservicio nota que permita la personalización de la fórmula como sus modificaciones. (Backend)	OMAR CUBA	5
		2	Desarrollo de la GUI para cambiar la fórmula “no personalizada” a “fórmula personalizada”. (Backend)	MAYCOL ESPINOZA	5
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2
HU25	Volver a fórmula no personalizada	1	Desarrollo de la funcionalidad volver a fórmula no personalizada en el API “Dar mantenimiento a fórmula” (Backend)	OMAR CUBA	5
		2	Desarrollo de la GUI para volver de la fórmula “personalizada” a “fórmula no personalizada”. (Frontend)	MAYCOL ESPINOZA	5
		3	Realización de pruebas con programa Postman. (Backend)	OMAR CUBA	2
		4	Realización de pruebas en base a los criterios de aceptación. (Frontend)	MAYCOL ESPINOZA	2

(Elaboración propia)

Revisión: En la fecha 13 de agosto de 2019 se dio la revisión con la dueña del producto Clara Díaz Tinoco la cual consistió en la validación de los criterios de aceptación de las historias de usuario definidas en el sprint. La duración fue de 3 horas aproximadamente.

Retrospectiva: En la fecha 13 de agosto de 2019 se dio la retrospectiva entre los integrantes del equipo scrum. La duración fue de 3 horas. Se analizaron los siguientes puntos:

- La duración fue alrededor de 3 semanas se logró mantener el ritmo de avance como el anterior sprint.
 - Hubo algunas complicaciones en el desarrollo de las fórmulas backend y frontend que gracias a la coordinación se pudo superar.

4.2.3 Diseño de la solución

4.2.3.1 Diseño de la arquitectura

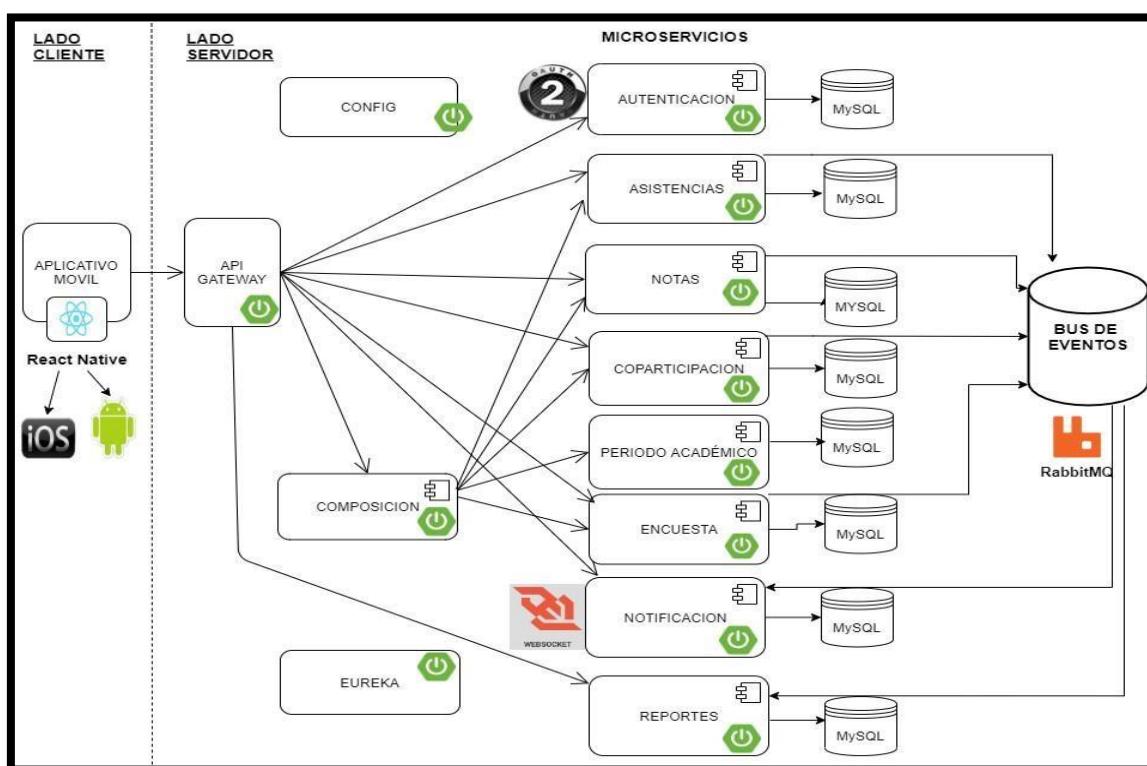


Figura 67. Diagrama del diseño de la arquitectura del sistema.(De la Cruz, Espinoza, & Cuba, 2019)

La presente arquitectura del sistema(véase Figura 67), está compuesto por el lado cliente(frontend) en el cual se usará el framework reactnative que permite el despliegue

en plataformas móviles como Android e iOS, por el lado del servidor(backend) se cuentan con los componentes no funcionales que apoyan a nuestra arquitectura de microservicios(confit, eureka, gateway y composición) , los microservicios(authenticación, asistencias, notas, coparticipación, encuesta, notificación, reportes) cada uno será desarrollado con springboot y cuenta con acceso privado a su base de datos, para agregar seguridad de acceso a servicios se hará uso de Oauth2 y por último para gestionar los eventos(mediante mensajería) se usará RabbitMQ.(De la Cruz, Espinoza, & Cuba, 2019)

A continuación, se detallarán cada uno de los componentes que escogimos para la arquitectura.

4.2.3.1.1 Lado cliente

En el lado cliente se encuentra la parte del software que interactúa directamente con los usuarios en este caso es un aplicativo móvil desarrollado con el framework reactnative. Este aplicativo interactúa con el lado servidor mediante el consumo de los servicios web mediante REST y también hace uso de websockets.

4.2.3.1.1.1 ReactNative

ReactNative es un framework JavaScript que permite crear aplicaciones las cuales serán renderizadas en Android y/o IOS. Está basado en React, el cual es una librería de Javascript de Facebook para desarrollar interfaces de usuario. Similar a React para web, reactnative usa una mixtura de Javascript y etiquetas XML(JSX), por debajo se invoca APIs que invocan el renderizado nativo de Java (para Android) y de Objective-c (para IOS). (Eisenman, 2016, págs. 1-4)

De acuerdo con(Eisenman, 2016, págs. 1-4), entre sus ventajas están:

- Provee de un mejor rendimiento que otros frameworks híbridas como cordova o ionic ya que usa el renderizado de forma nativa en vez de webviews.
- Para el desarrollador, al realizar los cambios no es necesario realizar un rebuild(reconstruir) el proyecto, si no que este solo necesita refrescar ya que usa Javascript.
- Si ya se sabe programar en react para web es más fácil adaptarse.
- Permite compartir e iterar más rápidamente y compartir el conocimiento más efectivamente.

De acuerdo con(Eisenman, 2016, págs. 1-4), entre sus desventajas están:

- Se introduce una nueva capa al proyecto que hace un poco más difícil la depuración del código fuente.

4.2.3.1.2 Lado servidor

En el lado servidor se encuentra la lógica de negocio y acceso a los datos, la cual no es visible para el cliente.

Aquí se encuentra la arquitectura de microservicios la que se puede clasificar en componentes en netamente no funcionales y funcionales(incorporan la lógica de negocio). Para los no funcionales se muestran los componentes: Eureka-Server, Config-Server, Api Gateway y Composición. Para los funcionales se tienen los siguientes componentes: Autenticación, Asistencia, Coparticipación, Notas, Encuestas, Periodo-Académico y Notificaciones. A continuación, se realiza la descripción detallada de los componentes anteriormente mencionados:

4.2.3.1.2.1 Componentes no funcionales

No contienen lógica de negocio, pero aportan en el funcionamiento de la arquitectura de microservicios se hace uso del framework springcloud y también consideramos aquí al servicio composición el cual requiere el uso de la combinación de varios microservicios para las consultas del lado cliente(frontend).

4.2.3.1.2.1.1 Eureka-Server

Este servicio usa el componente springcloud eureka que permite realizar el auto-registro, descubrimiento dinámico y balanceo de carga.(Rajesh, 2016, págs. 232-235)

El auto-registro de los microservicios es un mecanismo que mantiene un registro de los estados de los microservicios, de manera similar si un microservicio se da de baja, el registro se mantiene actualizado con los sucesos que hayan ocurrido en la red de servicios. El servidor eureka lo registra en el siguiente ping que realiza por defecto cada 30 segundos a los servicios si no responde lo quita de la lista de servicios.(Rajesh, 2016, págs. 232-235)

El descubrimiento automático de los microservicios es otra de sus funciones, los clientes son capaces de obtener el estado actual de cada microservicio. Este enfoque se opone a la configuración estática del registro de IPS de los servicios que obligaría a que se deba reiniciar los servicios o propagar cambios usando sólo springcloudconfig mediante springcloud bus manualmente. Se hace una consulta(ping) a los microservicios registrados por defecto cada 30 segundos.(Rajesh, 2016, págs. 232-235)

El balanceo de carga se hace mediante uso de ribbon (el cual es parte springcloud que realiza balanceo de carga y se hace uso del algoritmo round-robin para escoger al más servicio disponible más cercano)(Rajesh, 2016, págs. 232-235)

En el siguiente gráfico(véase Figura 68) se muestra su funcionamiento.

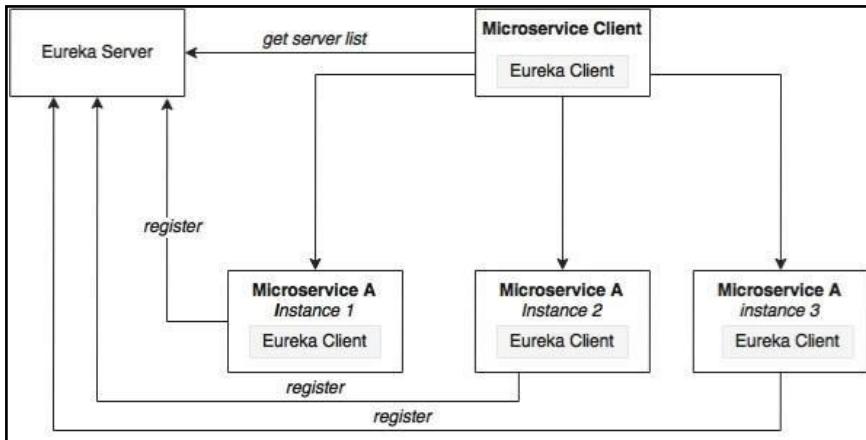


Figura 68. Diagrama del funcionamiento del servidorespringcloud eureka(Rajesh, 2016, págs. 232-235)

En la imagen (véase Figura 68), se puede apreciar el registro de las diferentes instancias de los microservicios A: instancia 1, instancia 2, instancia 3; y se puede apreciar que otro microservicio requiere la lista de servicios disponibles al servidor eureka. Además, se puede apreciar que todos son clientes de eureka server.

423.1212 Config-Server

Este servicio usa el componente springcloudconfig server el cual permite la configuración externa en el cual los aplicativos y servicios pueden depositar, acceder y administrar todas las propiedades de configuración en tiempo real. Se encarga de la propagación de los cambios de configuración a todos los microservicios suscritos. Si se desea propagar los cambios se puede hacer uso de springcloud bus(que provee de un mecanismo para refrescar las configuraciones de múltiples instancias, lo cual se logra al conectar las instancias de los servicios a través de un messagebroker) se puede usar RabbitMQ para estos fines. Spring Config-Server almacena las propiedades en un repositorio de control de versiones, tales como GIT o SVN, el repositorio puede ser local o remoto.(Rajesh, 2016, págs. 211-223)

Como se muestra en la imagen (véase Figura 69), cada microservicio contiene un cliente de configuración embebido el cual hace una búsqueda a un servidor central de configuración usando un simple mecanismo declarativo, para que luego éstos los puedan almacenar en su propio entorno de configuración.

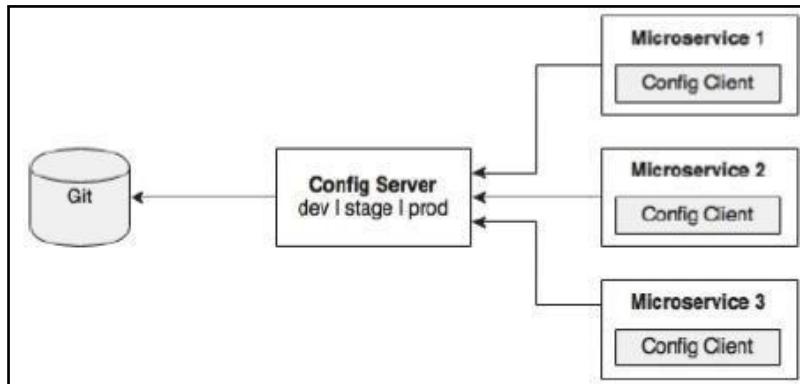


Figura 69. Diagrama del funcionamiento del servicio springcloudconfig(**Rajesh, 2016, pág. 213**)

4.2.3.1.2.1.3 API Gateway

Este servicio se sitúa entre el cliente y el servidor, transforma la data según lo requerido por el cliente (servicios UI). Actúa como un proxy para el backend, exponiendo un conjunto de APIs específicas al cliente. Además, es un intermediario que compone y transforma APIs de múltiples microservicios.(Rajesh, 2016, págs. 152-154)

Si se usa el api Gateway como un proxy reverso se puede usar como una plataforma compartida(B), si en cambio se requiere tener un control específico del tráfico y transformaciones complejas se requiere un api Gateway para cada microservicio serio(A) más útil (véase Figura 70).(Rajesh, 2016, págs. 152-154)

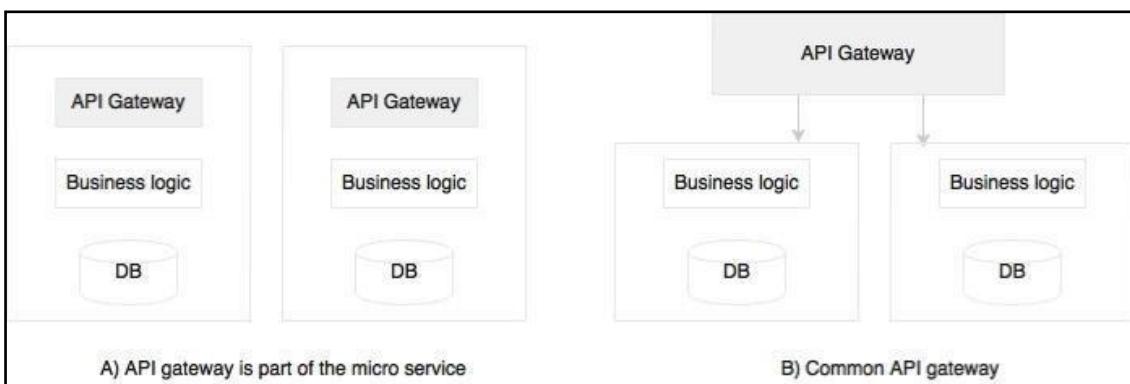


Figura 70. Tipos de API Gateway(**Rajesh, 2016, pág. 153**)

En la presente tesis se usó la opción B, un API Gateway común debido a que no necesitábamos tener control específico del tráfico y además por contar con limitada disponibilidad de recursos.

De acuerdo con(**Rajesh, 2016, págs. 152-154**) en muchas implementaciones de microservicios los endpoints no son expuestos directamente permanecen como servicios privados. Las razones para hacerlo son:

- Solo un grupo de microservicios son requeridos por los clientes.

- Políticas específicas del cliente, son fáciles de manejar en un solo sitio en vez de varios lugares.
- Las transformaciones de que requiere el cliente son más difíciles ser implementadas en los endpoints de los servicios.
- Si hay agregación de data, para evitar múltiples llamadas de clientes que restrinjan el ancho de banda de la red, el gateway debe usarse en el medio.

En la presente tesis, hacemos uso de zuul el cual es un servicio gateway que proviene de los productos de microservicios de Netflix. También denominado servicio de proxy reverso, éste usa el servidor de eureka para el descubrimiento de servicios y ribbon para balancear la carga entre las instancias de los servicios. También es capaz de enrutamiento, monitoreo, administración, flexibilidad, seguridad entre otros; además puede cambiar el comportamiento de los servicios al sobrescribirlos en la capa de los APIS.

4231214 Composición

Este servicio usa el patrón API composition, este patrón implementa una consulta(query) al invocar a los servicios que tienen la data por medio de sus APIS y las combina los resultados en un resultado. (Richardson, 2018, págs. 238-245)

Api Composer tiene 2 tipos de participantes (véase Figura 71): el primero tipo es el API Composer que implementa la consulta al realizar las consultas a los servicios proveedores) y como segundo tipo vienen a ser los servicios proveedores (son aquellos servicios que proveen la información que requiere la consulta)

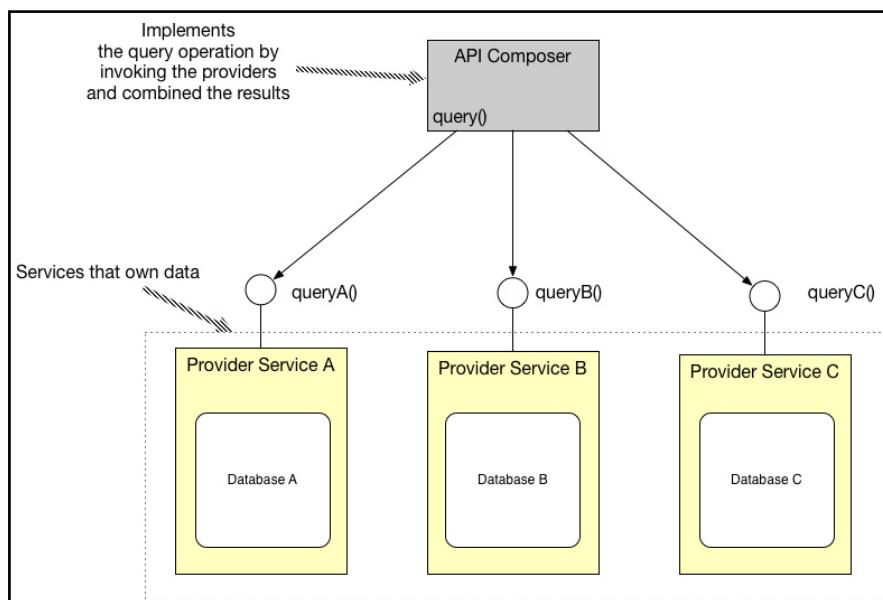


Figura 71. Diagrama de funcionamiento del API Composer(Richardson, 2018, pág. 239)

Cuando se desarrolla un sistema distribuido es recomendable minimizar los tiempos de respuestas para eso el API Composer debe llamar en paralelo para conseguir este fin, en

algunos casos si habrá dependencia y tendrá que ser secuencial, el objetivo es combinar operaciones paralelas y secuenciales para obtener un mejor rendimiento. Se recomienda usar el modelo de programación reactiva (permite la programación asíncrona y operaciones en paralelo), en java se cuenta con CompletableFutures, rxjava observables u otro equivalente, en la presente tesis se hizo uso CompletableFuture de java8 para la llamada asíncrona en paralelo de las APIS de los servicios. (Richardson, 2018, págs. 238-245)

De acuerdo con(Richardson, 2018, págs. 238-245), para implementar un API Composition hay 3 opciones las cuales se detallarán a continuación:

- 1. Implementar API Composition en el cliente:** En esta opción (véase Figura 72) el cliente se encarga de realizar las consultas a los servicios proveedores. Es eficiente siempre y cuando la conexión a estos servicios sea rápida, si están en una conexión lenta acceder a ellos y cuenta con firewall externo no es práctico.

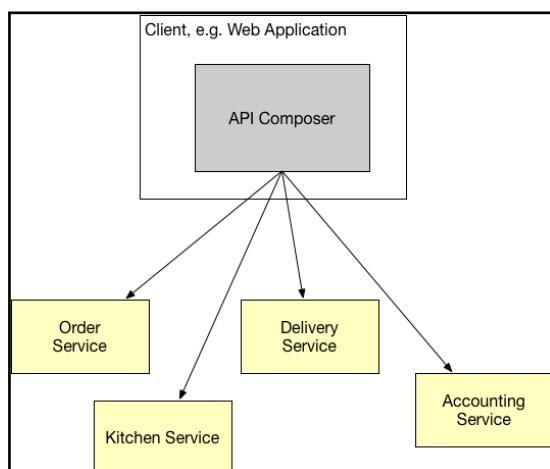


Figura 72. Diagrama de API Composer en el cliente. (Richardson, 2018, pág. 241)

- 2. Implementar API Composition en el API Gateway:** En esta opción (véase Figura 73) el API Gateway implementa la lógica del API Composition o también llamado API Composer, esto hace que el cliente haga solo uso de una única llamada y traer eficientemente la data de números servicios

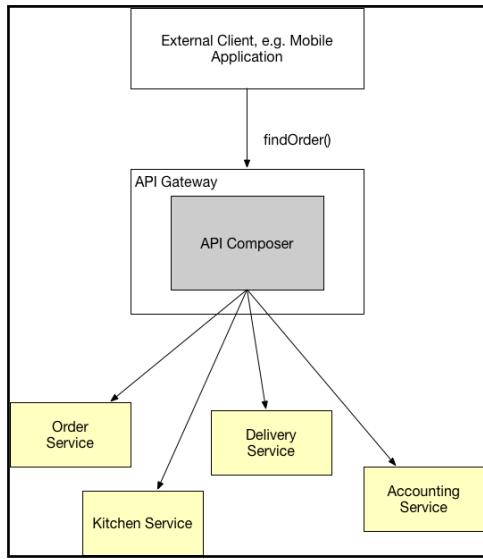


Figura 73. Diagrama de API Composer en el API Gateway(Ricardson, 2018, pág. 242)

3. **Implementar API composition en un servicio aislado:** Este enfoque (véase Figura 74) se puede usar cuando los servicios requieren usar por otros servicios y cuando la lógica sea compleja para ser parte del API Gateway.

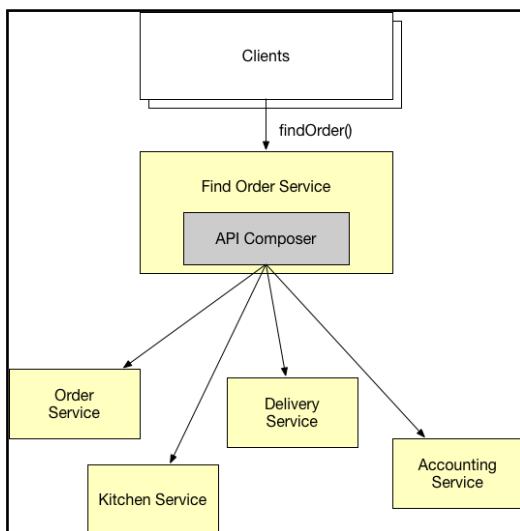


Figura 74. Diagrama de API Composer como servicio aislado(Ricardson, 2018, pág. 243)

En la presente tesis, se usó la tercera opción para que sea otro servicio el que maneje la lógica de combinar la información y no afecte al API Gateway ya que lo usamos compartido para los otros microservicios, además que eficientemente trae la información de los otros servicios en una sola llamada ya que se encuentran en una conexión aceptable, quedo descartada la opción del API Composition en el cliente debido a que la conexión no es confiable en la rapidez esta se usara en distintos lugares en los cuales la conexión puede ser lenta.

Por último (Richardson, 2018, págs. 238-245) menciona que las desventajas de usar el patrón API Composition son:

Incremento de la sobrecarga: Se necesita recursos de cómputo y red para poder combinar y realizar queries sobre los diferentes servicios, a diferencia de un aplicativo monolítico que tiene una sola base de datos en el que solo una llamada es suficiente.

Riesgo de disponibilidad reducida: Como se usan n servicios en vez de solo 1 servicio es más probable uno pueda no estar disponible, para esto se pueden usar estrategias de cache por si esto sucediera. La otra estrategia es que el API Composition puede omitir esa información que aún sea útil pero que no es en sí la información completa.

Riesgo de falta de consistencia transaccional: Puede ser que en el momento de realizar la consulta un servicio tenga información de un servicio este consistente y otra no, dado que en microservicios existe la consistencia eventual. A diferencia de un aplicativo monolítico que tiene transacciones ACID que asegura que la data es consistente.

4.2.3.1.2.2 Componentes funcionales

Aquí se encuentra la lógica de negocio, y es donde se encuentran los microservicios. Cada uno es independiente entre sí definidos así por límites de negocio, manejan su propia base de datos. Para la seguridad se hace uso de Oauth2 que permite acceso a los servicios cuando el usuario se autentica mediante tokens. Por último, se hace uso de RabbitMQ para que los microservicios al registrar cambios se comuniquen entre sí mediante eventos(mensajes) esto permite la programación asincrónica y el uso de colas en cambio la comunicación mediante HTTP por defecto no es secuencial es decir aumenta el tiempo de ejecución del servicio y no es recomendable a menos que se maneje de forma asincrónica.

4.2.3.1.2.2.1 Autenticación

Descripción:

Este microservicio contiene la información relacionada al usuario, roles y perfiles; permite a un usuario con perfiles como (profesor, alumno, directora de escuela) la autenticación en el sistema mediante el ingreso de su usuario y contraseña; también permite realizar el cierre de sesión de un usuario dado. Aquí se usa Oauth2 el cual brinda token al cliente que permitirá el acceso de los microservicios.

Base de datos:

A continuación, se muestra el diagrama de base de datos(véase Figura 75) perteneciente al microservicio autenticación.

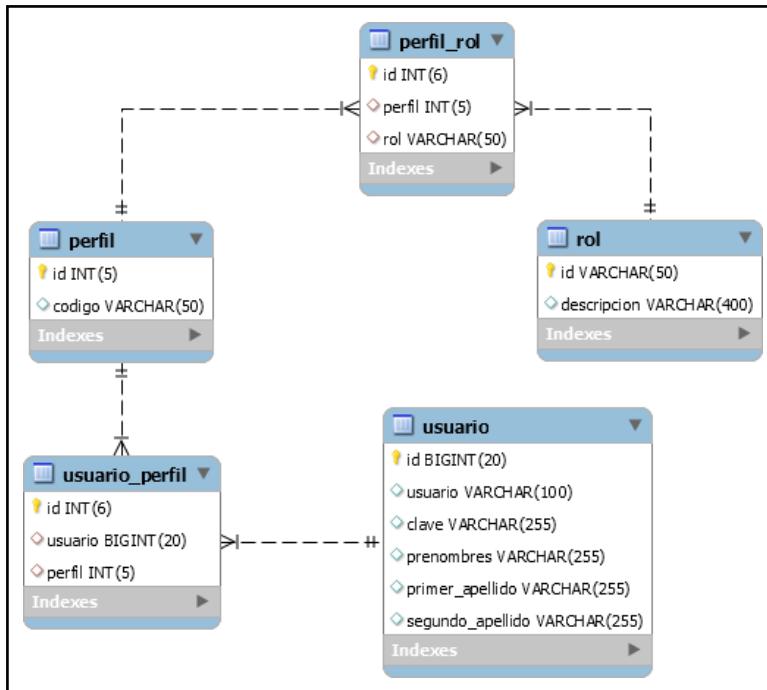


Figura 75.Diagrama de base de datos del servicio autenticación(Elaboración propia)

Servicios expuestos:

1. Iniciar sesión:

URL: {{URL_APIGATEWAY}}/aplicativos/autenticacion/oauth/token

Descripción: Permite la autenticación del usuario mediante el usuario y contraseña.

2. Cerrar sesión:

URL: {{URL_APIGATEWAY}}/aplicativos/autenticacion/logout

Descripción: Permite el cierre de sesión mediante el token.

3. Obtener información del usuario:

URL: {{URL_APIGATEWAY}}/aplicativos/autenticacion/user

Descripción: Permite obtener la información del usuario que ha iniciado sesión mediante el token.

4231222 Asistencia

Descripción:

Este microservicio permite registrar/actualizar y finalizar las asistencias de las clases, en cada clase se registran las asistencias de los alumnos.

- Al registrar la clase se inicializa por defecto a todos los alumnos como inasistencia.

- Al actualizar la clase se guardan las modificaciones de las asistencias de alumnos en la clase.
- Al finalizar la clase se culmina el proceso de asistencias de la clase y de los alumnos.
- Además, permite realizar consultas para obtener la información de las asistencias registradas.

Base de datos:

A continuación, se muestra el diagrama de base de datos(véaseFigura 76) perteneciente al microservicio asistencia.

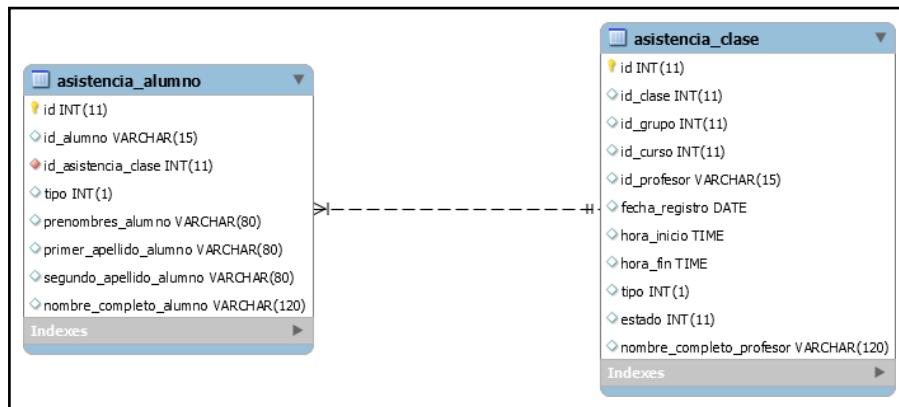


Figura 76.Diagrama de base de datos del servicio asistencia(Elaboración propia)

Servicios expuestos:

1. Iniciar asistencia:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/iniciarAsistenciaClase?idClase={{ID_CLASE}}

Descripción:Permite iniciar la asistencia mediante el id de la clase.

2. Actualizar asistencia:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/actualizarAsistenciaClase

Descripción:Permite la actualización de la clase se envía la información de la clase, y las asistencias de los alumnos.

3. Finalizar asistencia:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/finalizarAsistenciaClase?idClase={{ID_CLASE}}

Descripción:Permite finalizar la asistencia de la clase mediante el id de la clase.

4. Obtener asistencia de clase por id de clase:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/buscarPorIdClase?idClase={{ID_CLASE}}&traerAsistAlumnos={{TRAER_ASIST_ALUMNOS}}

Descripción: Permite realizar el listado de las asistencias de las clases por el id de clase y las asistencias de alumno(opcional)

5. Obtener asistencia de alumnos por id de grupo:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/buscarAsistenciaAlumnoPorIdGrupo?idGrupo={{ID_GRUPO}}

Descripción: Permite realizar el listado de las asistencias de los alumnos por el id de grupo

6. Obtener asistencia de clases por id de grupo:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/buscarAsistenciaClasePorIdGrup?idGrupo={{ID_GRUPO}}

Permite realizar el listado de las asistencias de las clases por el id de grupo

7. Obtener asistencias de alumnos por la lista de id de las clases:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/buscarAsistenciaAlumnoPorClases?idListaClases={{LISTA_ID_CLASES}}

Descripción: Permite realizar el listado de las asistencias de los alumnos por la lista de id de las clases.

8. Obtener asistencias de clases por la lista de id de las clases:

URL: {{URL_APIGATEWAY}}/aplicativos/asistencia-service/api/asistencia/buscarAsistenciaAlumnoPorClases?idListaClases={{LISTA_ID_CLASES}}

Descripción: Permite realizar el listado de las asistencias de las clases por la lista de id de las clases.

4231223 Encuesta

Descripción:

Este microservicio permite realizar las encuestas actualmente estas calificaciones serán de 1 a 5 estrellas. Permite obtener la última plantilla de encuesta en estado activo con la cual se realizarán las encuestas. Por último, permite calcular el promedio de encuestas realizadas en base a las clases.

Base de datos:

A continuación, se muestra el diagrama de base de datos(véaseFigura 77) perteneciente al microservicio encuesta.

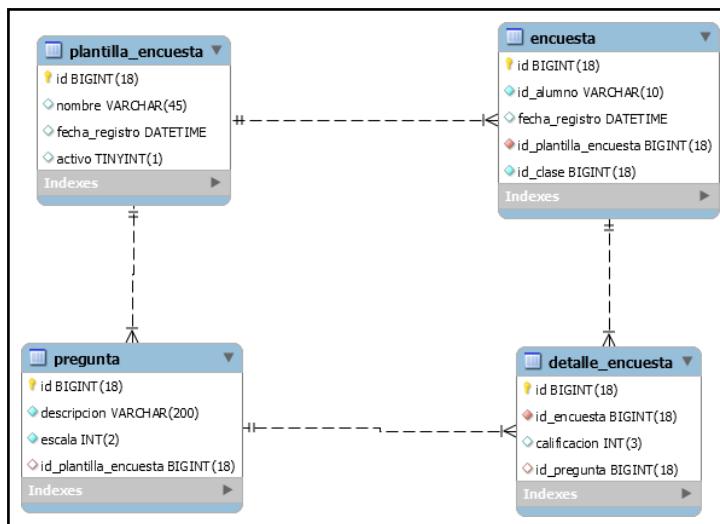


Figura 77.Diagrama de base de datos del servicio encuesta(Elaboración propia)

Servicios expuestos:

1. Obtener plantilla de encuesta actual:

URL: {{URL_APIGATEWAY}}/encuesta-service/api/encuesta/obtenerPlantillaEncuestaActual

Descripción: Permite obtener la actual plantilla de encuesta con la cual se realizarán las encuestas.

2. Registrar encuesta

URL: {{URL_APIGATEWAY}}/encuesta-service/api/encuesta/registrarEncuesta

Descripción: Permite registrar la encuesta de la clase realizada por el alumno se guardan los detalles de la encuesta: la calificación dada por el alumno y la pregunta vinculada a ésta.

3. Obtener promedio por lista de id de las clases:

URL: {{URL_APIGATEWAY}}/encuesta-service/api/encuesta/buscarPromedioPorClases?idListaClases={{LISTA_ID_CLASES}}

Descripción: Permite obtener la lista de encuestas de las clases con el promedio calculado de todas las encuestas realizadas por cada clase, se envía la lista de id de las clases.

4231224 Coparticipación

Descripción:

Entre las principales funciones de este microservicio son las de “registrar pregunta”, el cual permite a los estudiantes registrar preguntas referentes a un tema en específico o en general para que puedan ser respondidas por el docente o en todo caso por un compañero de clase. “responder respuesta” es el caso consecuente de la acción

“registrar pregunta”, puede ser realizado por profesores y/o alumnos con el fin de solventar la duda registrada en la pregunta. “Registrar anuncios (por parte de los profesores)”, esta funcionalidad permite al profesor hacer anuncios de algunos imprevistos o avisos que quiera comunicar.

Base de datos:

A continuación, se muestra el diagrama de base de datos(véaseFigura 78) perteneciente al microservicio coparticipación.



Figura 78. Diagrama de base de datos del servicio coparticipación(Elaboración propia)

Servicios expuestos:

1. Listar anuncios:

URL: {{URL_APIGATEWAY}}/coparticipacion-service/api/coparticipacion/buscarAnunciosPor?idCurso={{ID_CURSO}}&idGrupo={{ID_GRUPO}}

Descripción: Permite listar los anuncios dado el código del curso y el código de grupo(opcional)

2. Registrar anuncios:

URL: {{URL_APIGATEWAY}}/coparticipacion-service/api/coparticipacion/registrarAnuncio

Descripción: Permite registrar un anuncio del profesor que contiene un título y detalle además se indica a que curso corresponde y opcionalmente a que grupo irá dirigido.

3. Listar preguntas:

URL: {{URL_APIGATEWAY}}/coparticipacion-service/api/coparticipacion/buscarPreguntasPor?idCurso={{ID_CURSO}}&idGrupo={{ID_GRUPO}}&idClase={{ID_CLASE}}

Descripción: Permite listar las preguntas registradas por el alumno o profesor, se puede buscar por código de curso(obligatorio), código de grupo es opcional, código de clase es opcional.

4. Registrar pregunta:

URL: {{URL_APIGATEWAY}}/coparticipacion-service/api/coparticipacion/registrarPregunta

Descripción: Permite registrar una pregunta del profesor o del alumno que contiene un detalle además se indica a que curso corresponde, opcionalmente a que grupo y/o clase está dirigida.

5. Listar respuestas:

URL: {{URL_APIGATEWAY}}/coparticipacion-service/api/coparticipacion/buscarRespuestasPor?idPregunta={{ID_PREGUNTA}}

Descripción: Permite listar las respuestas dado el código de la pregunta.

6. Registrar respuestas:

URL: {{URL_APIGATEWAY}}/coparticipacion-service/api/coparticipacion/registrarRespuesta

Descripción: Permite registrar una respuesta de una pregunta ya registrada, opcionalmente se puede vincular esta respuesta al nombre de la persona que hizo otra respuesta.

4231225 Notas

Descripción:

Este microservicio permite a los docentes asignados a los grupos de un curso poder ingresar las notas de las diferentes evaluaciones de sus alumnos como por ejemplo examen parcial, prácticas, exposiciones entre otros. A su vez permite la consulta y el registro de fórmulas generales(fórmula promedio de todo el curso) y específicas(fórmula específica por cada fórmula general).

Base de datos:

A continuación, se muestra el diagrama de base de datos(véase Figura 79) perteneciente al microservicio notas.

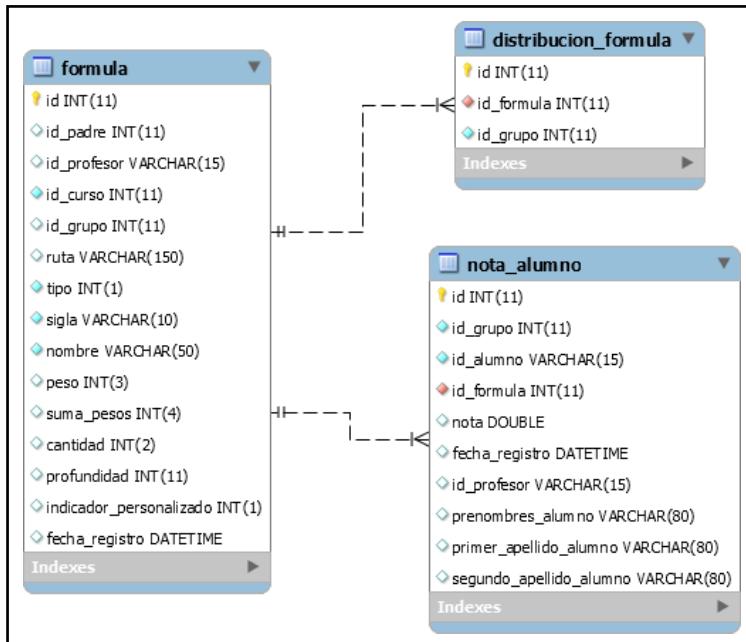


Figura 79. Diagrama de base de datos del servicio notas(Elaboración propia)

Servicios expuestos:

1. Listar fórmulas generales:

URL: {{URL_APIGATEWAY}}/nota-service/api/nota/buscarFormulasGenerales?idCurso={{ID_CURSO}}

Descripción: Permite listar las fórmulas generales del curso, estas representan la fórmula principal para calcular la nota final del curso. Por ejemplo: fórmula nota final es igual al promedio de teoría + promedio de prácticas clínicas y seminarios.

2. Listar fórmulas específicas:

URL: {{URL_APIGATEWAY}}/aplicativos/nota-service/api/nota/buscarFormulasEspecificas?idFormulaPadreGeneral={{ID_FORMULA_PADRE_GENERAL}}&idGrupo={{ID_GRUPO}}

Descripción: Permite listar las fórmulas específicas dado una fórmula general seleccionada. Por ejemplo: fórmula del promedio de teoría es igual al cálculo del promedio de exámenes parciales.

3. Listar fórmulas específicas hijas:

URL: {{URL_APIGATEWAY}}/aplicativos/nota-service/api/nota/buscarFormulasEspecificasHijas?idFormulaPadreEspecifica={{FORMULA_PADRE_ESPECIFICA}}

Descripción: Permite listar las fórmulas dependientes de la fórmula específica. Por ejemplo: una fórmula específica como examen parcial puede tener fórmulas hijas como 4 exámenes parciales con diferente porcentaje cada uno.

4. Listar notas:

URL: {{URL_APIGATEWAY}}//aplicativos/nota-service/api/nota/buscarNotasEspecificas?idGrupo={{ID_GRUPO}}&idFormulaEspecificaHija={{ID_FORMULA_ESPECIFICA_HIJA}}

Descripción: Permite listar las notas según el grupo y la fórmula específica hija. Se realiza una búsqueda de una fórmula por grupo si existe y si no se toma por defecto la fórmula por defecto para todos los grupos.

5. Listar fórmulas y notas por alumno:

URL: {{URL_APIGATEWAY}}//aplicativos/nota-service/api/nota/buscarFormulasYNotasPorAlumno?idFormulaPadreGeneral={{ID_FORMULA_PADRE_GENERAL}}&idGrupo={{ID_GRUPO}}

Descripción: Permite mostrar todas las notas del alumno dado una fórmula padre. Por ejemplo: si trajera la fórmula teoría me traería el detalle de las notas de todas las evaluaciones de teoría en la cual se incluyen los 4 exámenes parciales.

6. Dar mantenimiento a notas:

URL: {{URL_APIGATEWAY}}//aplicativos/nota-service/api/nota/darMantenimientoANotas

Descripción: Permite poder modificar las notas de los alumnos, se envía una relación de alumnos y sus notas respectivas.

4231226 Periodo académico:

Descripción:

Este microservicio contiene la información de los cursos en un semestre, tiene la información de los alumnos y profesores, los grupos definidos en el curso, la relación de alumnos inscritos en dichos grupos y también la relación de los horarios de las clases para cada grupo.

Base de datos:

A continuación, se muestra el diagrama de base de datos (véase Figura 80) perteneciente al microservicio periodo-académico.

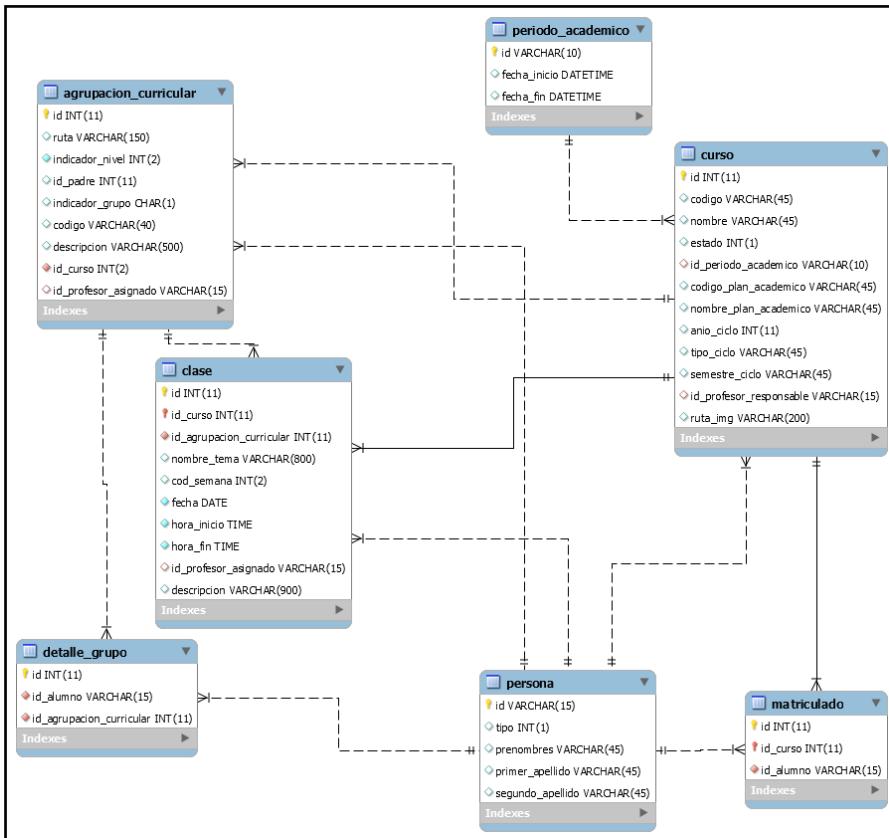


Figura 80. Diagrama de base de datos del servicio periodo-académico (Elaboración propia)

Servicios expuestos:

1. Listar cursos por código de profesor/alumno

URL: {{URL_APIGATEWAY}} /aplicativos/periodo-academico-service/api/curso/listar?tipoPersona={{TIPO_PERSONA}}

Descripción: Según el usuario que inicio de sesión y tipo de persona(alumno o profesor) se listan los cursos en los que se encuentran inscritos bien sean los alumnos o los profesores

2. Buscar curso por id

URL: {{URL_APIGATEWAY}} /aplicativos/periodo-academico-service/api/curso/buscarPorId?id={{ID_CURSO}}

Descripción: Según el id del curso dado se busca la información detallada del curso.

3. Listar clases de alumno o profesor

URL: {{URL_APIGATEWAY}} /aplicativos/periodo-academico-service/api/clase/buscarPor?idCurso={{ID_CURSO}}&tipoPersona={{TIPO_PERSONA}}

Descripción: Según el usuario que inicio de sesión y tipo de persona (alumno o profesor) y el id del curso se muestran todas las clases en los que se encuentran inscritos bien sean los alumnos o los profesores

4. Buscar clases de hoy

URL: {{URL_APIGATEWAY}}/aplicativos/periodo-academico-service/api/clase/buscarPorFechaHoy?tipoPersona={{TIPO_PERSONA}}

Descripción: Método que permite listar las clases del presente día.

5. Buscar por id de clase

URL: {{URL_APIGATEWAY}}/aplicativos/periodo-academico-service/api/clase/buscarPorId?id={{ID_CLASE}}

Descripción: Método que permite buscar la información de la clase por medio del id de la clase.

6. Listar alumnos por id de grupo

URL: {{URL_APIGATEWAY}}/aplicativos/periodo-academico-service/api/persona/buscarAlumnosPor?idGrupo={{ID_GRUPO}}

Descripción: Método que permite que según el id de grupo de un curso se pueden listar los alumnos inscritos en éste.

7. Buscar por id de clase y listar alumnos inscritos

URL: {{URL_APIGATEWAY}}/aplicativos/periodo-academico-service/api/clase/buscarPorIdYListarAlumnos?id={{ID_CURSO}}

Descripción: Método que permite mostrar la información de las clases junto a la relación de los alumnos dado el id del curso.

8. Listar grupos

URL: {{URL_APIGATEWAY}}/aplicativos/periodo-academico-service/api/agrupacion_curricular/buscarPor?idCurso={{ID_CURSO}}&tipoPersona={{TIPO_PERSONA}}&traerTodoProfPrincipal={{TRAER_TODO_PROFESORES}}

Descripción: Método que permite mostrar la relación de grupos según el curso, tipo de persona y si es profesor principal traer toda la información si no se busca por el que ha iniciado sesión sea alumno o profesor.

9. Listar clases por id de grupo

URL: {{URL_APIGATEWAY}}/aplicativos/periodo-academico-service/api/clase/buscarPorIdGrupo?idGrupo={{ID_GRUPO}}&tipoPersona={{TIPO_PERSONA}}&traerTodoProfPrincipal={{TRAER_TODO_PROFESORES}}

Descripción: Método que permite mostrar la relación de clases según el curso, tipo de persona y si es profesor principal traer toda la información si no se busca por el que ha iniciado sesión sea alumno o profesor.

423.1.227 Notificación

Descripción:

Este microservicio es útil cuando hay cambios o actualizaciones por parte de los alumnos y profesores que es de importancia para los demás, de lo cual se resaltan las notificaciones de las asistencias, notas, anuncios, preguntas y respuestas. Además de los servicios rest expuestos, se hace uso de websockets para la sincronización del número

de notificaciones de los usuarios y la invocación internamente del servicio firebase para dispositivos Android con el fin de enviar notificaciones cuando el aplicativo se encuentre en background(segundo plano) o cerrado.

Base de datos:

A continuación, se muestra el diagrama de base de datos(véaseFigura 81) perteneciente al microservicio notificación.

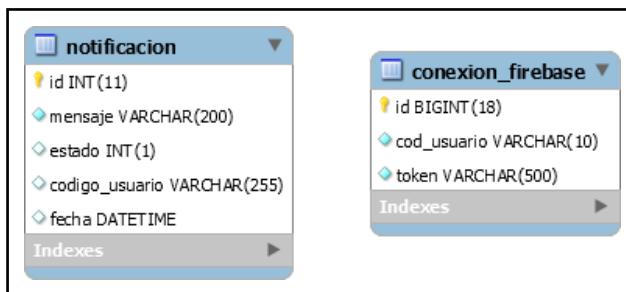


Figura 81. Diagrama de base de datos del servicio notificación(Elaboración propia)

Servicios expuestos

1. Listar notificaciones

URL: {{URL_APIGATEWAY}}/aplicativos/notificacion-service/notificacion/listarNotificaciones

Descripción:Método que permite realizar la consulta de todas las notificaciones por medio del código de usuario

2. Listar últimas 8 notificaciones

URL: {{URL_APIGATEWAY}}/aplicativos/notificacion-service/notificacion/listarUltimasNotificaciones

Descripción:Método que permite realizar la consulta de las últimas notificaciones por medio del código de usuario

3. Actualizar notificaciones del usuario

URL: {{URL_APIGATEWAY}}/aplicativos/notificacion-service/notificacion/actualizarNotificaciones

Descripción:Método que permite actualizar las notificaciones consultadas por el usuario, se envía la lista de los códigos de las notificaciones que ya se revisó.

4231228 Reportes

Descripción:

Este microservicio sirve para que se pueda generar reportes y/o estadísticas y que no necesite llamar a otros servicios es por eso por lo que cuenta con una base de datos que es alimentada por eventos por los microservicios de asistencias, notas, encuestas y periodo académico.

Servicios expuestos

1. Generar reporte de asistencia de estudiantes

URL: {{URL_APIGATEWAY}}/aplicativos/reporte-service/api/reporte/attendance/student

Descripción:Método que permite realizar el reporte de asistencias de estudiantes mediante el código del curso, código del grupo (se puede definir uno específico o TODOS) y código de alumno (se puede definir uno específico o TODOS)

2. Generar reporte de asistencia de profesores

URL: {{URL_APIGATEWAY}}/aplicativos/reporte-service/api/reporte/attendance/professor

Descripción:Método que permite realizar el reporte de asistencias de profesores mediante el código del curso, código del grupo (se puede definir uno específico o TODOS) y código de profesor (se puede definir uno específico o TODOS)

3. Generar reporte de calificaciones de estudiantes

URL: {{URL_APIGATEWAY}}/aplicativos/reporte-service/api/reporte/grades/student

Descripción:Método que permite realizar el reporte de notas de estudiantes mediante el código del curso, código del grupo (se puede definir uno específico o TODOS) y código de alumno (se puede definir uno específico o TODOS)

4. Generar reporte de asistencia de estudiantes

URL: {{URL_APIGATEWAY}}/aplicativos/reporte-service/api/reporte/quiz

Descripción:Método que permite realizar el reporte del promedio de encuestas de los profesores mediante el código del curso, código del grupo (se puede definir uno específico o TODOS) y código de profesor (se puede definir uno específico o TODOS)

4.2.3.2 Herramientas útiles

En el transcurso del desarrollo de la solución informática, se hizo uso de las siguientes herramientas informáticas:

Postman: nos permitió realizar pruebas de los servicios rest expuestos por los microservicios a través de formato JSON. En la siguiente imagen (véase Figura 82) se muestra el archivo postman que usamos que contienen los microservicios organizados por carpetas y contenido cada uno peticiones por cada funcionalidad de éstos.

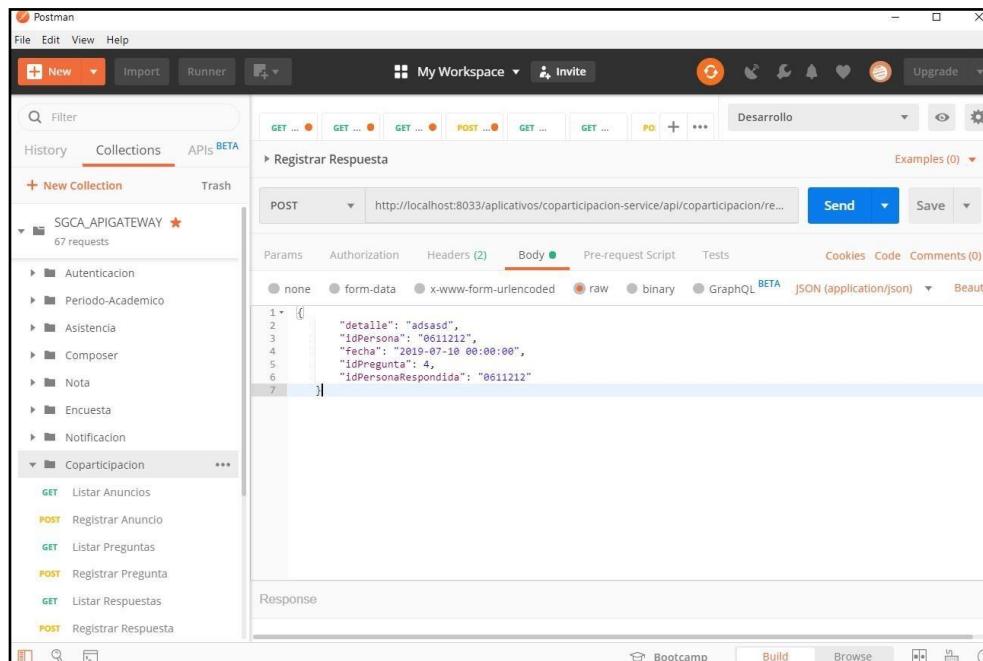


Figura 82. Uso de los servicios en la herramienta Postman(Elaboración propia).

Bitbucket: El uso de bitbucket nos permitió alojar de forma privada el repositorio con el controlador de versiones git(véase Figura 83 y Figura 84) en la nube. Este repositorio contiene el código fuente, backup de bases de datos (queries y backups) y el archivo postman que contiene las pruebas de los servicios rest, el uso del controlador de versiones fue bastante útil dado que se usó para comparar con versiones anteriores, para visualizar los nuevos cambios y además ayudo a solucionar algunos inconvenientes que se presentasen.

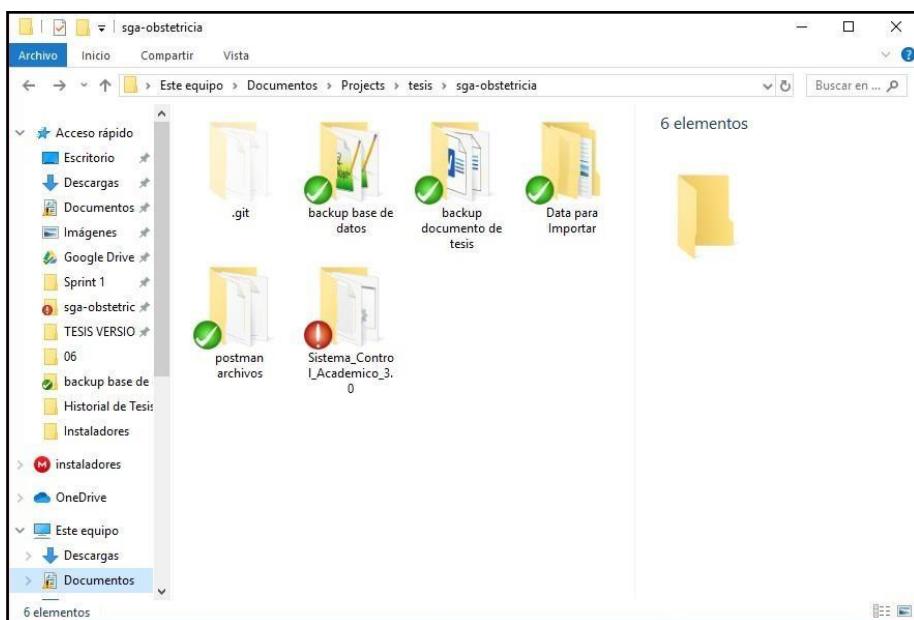


Figura 83. Uso de Bitbucket con repositorio GIT (Elaboración propia).

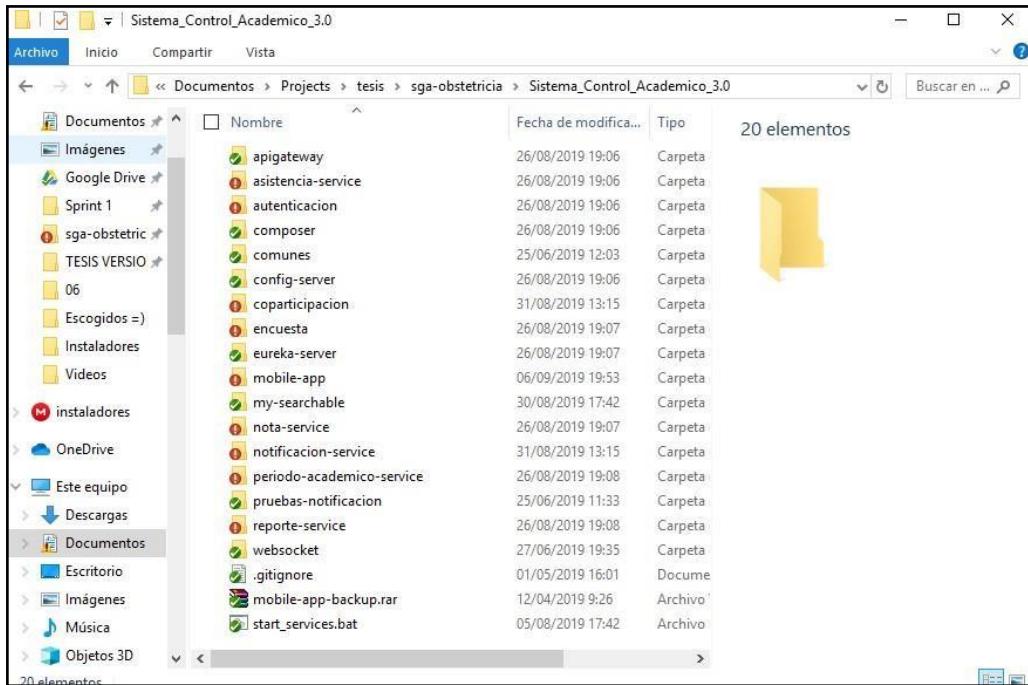


Figura 84. Código fuente del repositorio GIT en Bitbucket (Elaboración propia).

Onedrive: Nos permitió alojar toda la documentación de la tesis (véase Figura 85) como recursos compartidos, documentos, formatos, bibliografía.

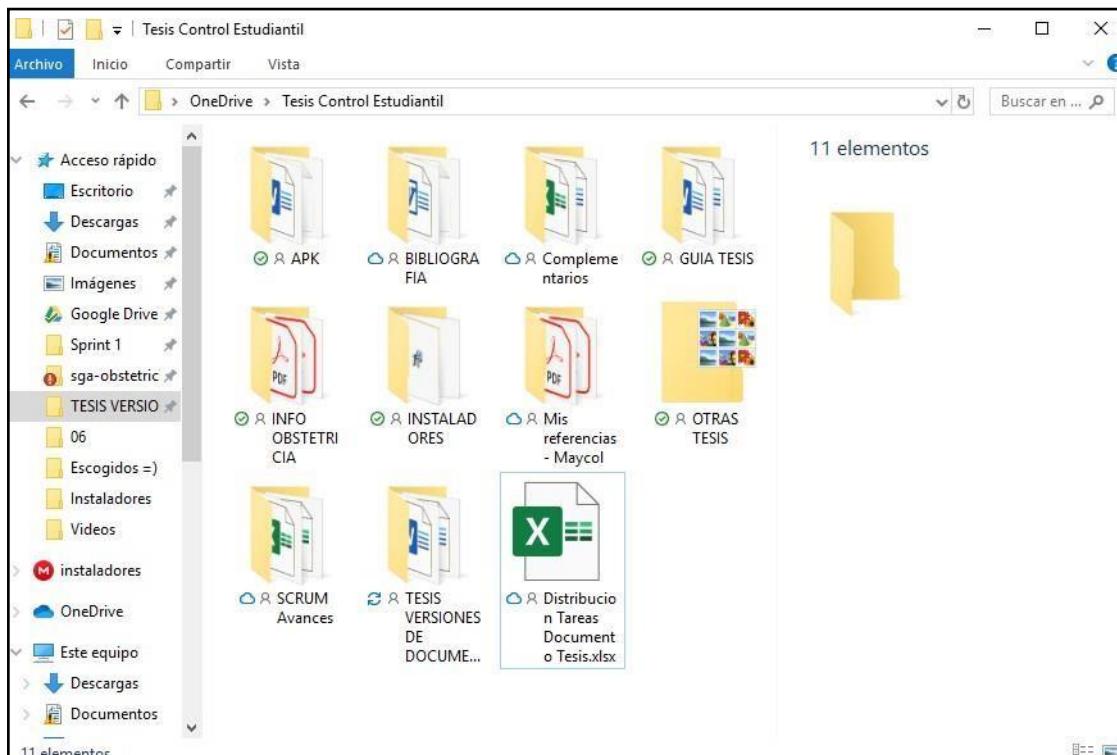


Figura 85. Uso de nube OneDrive para compartir archivos (Elaboración propia).

Hangouts, Skype: Nos permitieron trabajar de forma remotaen la cual realizábamos reuniones sobre el trabajo realizado y las tareas pendientes.

STS (Spring Tool Suite): Nos permitió el desarrollo, debug, pruebas de los microservicios.

Visual Studio Code, Atom: Nos permitieron el desarrollo, debug y pruebas del lado frontend(reactnative)

DB Beaver CommunityEdition: Nos permitió generar el esquema de la base de datos relacional.

ToadforMysql: Nos permitió realizar operaciones (creación, modificación, eliminación, consultas) en las bases de datos, así como importar información de archivos excel para realizar la carga académica como los cursos, grupos, clases, fórmulas, distribución de fórmulas, alumnos y profesores.

MysqlWorkbench: Nos permitió realizar operaciones (creación, modificación, eliminación, consultas) en las bases de datos, así como exportar las bases de datos en formato sql.

4.2.3.3 Manual de usuario

4.2.3.3.1 Inicio

Acceso al sistema

Aquí el usuario puede indicar sus parámetros de acceso (id de usuario y su clave) si es correcto dependiendo del acceso que tuviera se les mostrarán accesos a los menús como director de escuela, profesor o alumno; caso contrario indicará que el usuario y/o clave no son los correctos. (véaseFigura 86)



Figura 86. Interfaz: Acceso al sistema (Elaboración propia)

Seleccionar acceso

En el caso que tenga más de un acceso a los menús se mostrará la relación de accesos que tiene el usuario. Puede haber accesos de director de escuela, profesor y/o alumno si es que contase con estos. (véase Figura 87)



Figura 87. Interfaz: Elegir accesos (Elaboración propia)

4.2.3.3.2 Módulo del profesor

Menú: Profesor

Por defecto se listan las clases que corresponden al presente día, ordenadas a través del horario. Como se puede apreciar, cada clase (enmarcada dentro de un rectángulo) presenta su respectiva valoración de estrellas (0 no hay calificación, 1 es muy bajo y 5 es muy alto). (véase Figura 88)



Figura 88.Interfaz del profesor: Visualizar las clases de hoy (Elaboración propia)

Al deslizar de izquierda a derecha, se puede visualizar un menú lateral, donde se encuentran las opciones disponibles para el docente (Clases del presente día, Asistencias, Calificaciones, Preguntas y respuestas, Anuncios, Fórmulas, Cambio de perfil, Cerrar sesión). (véase Figura 89)

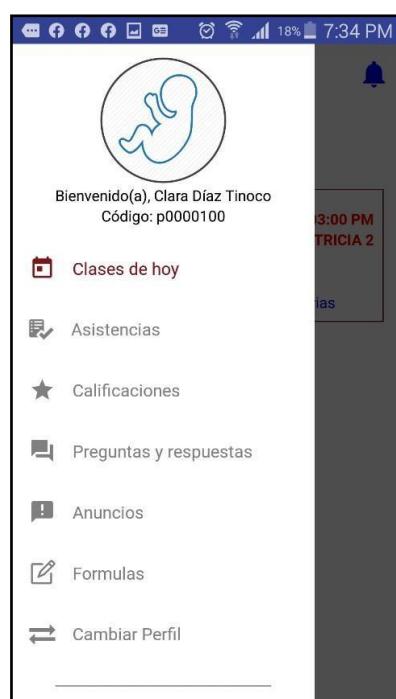


Figura 89.Interfaz del profesor: Visualizarmenúlateral(Elaboración propia)
Luego de seleccionar alguna de las clases correspondientes al presente día, el sistema presenta dos secciones; en una de ellas, se muestra el detalle de la clase, tales como,

nombre del docente, estado, horas de inicio y fin, valoración. En la otra, se muestra el listado de alumnos (código, nombres completos y estado de asistencia), el cual podrá ser filtrado por el código universitario o nombres. (véaseFigura 90)

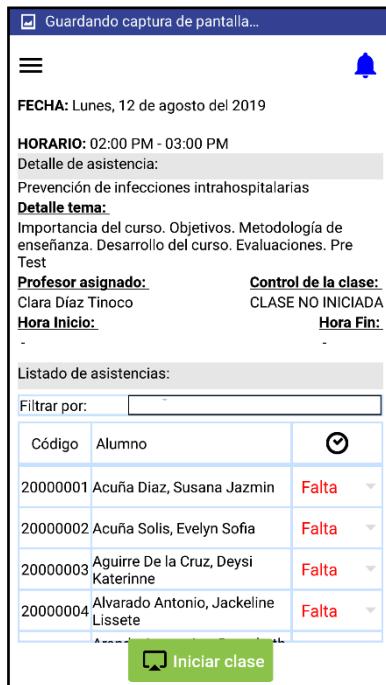


Figura 90.Interfaz del profesor: Visualizar clases de hoy (Elaboración propia)
Para el registro de asistencias, es indispensable que el docente inicie la clase indicando la opción “Iniciar clase”. (véaseFigura 91)

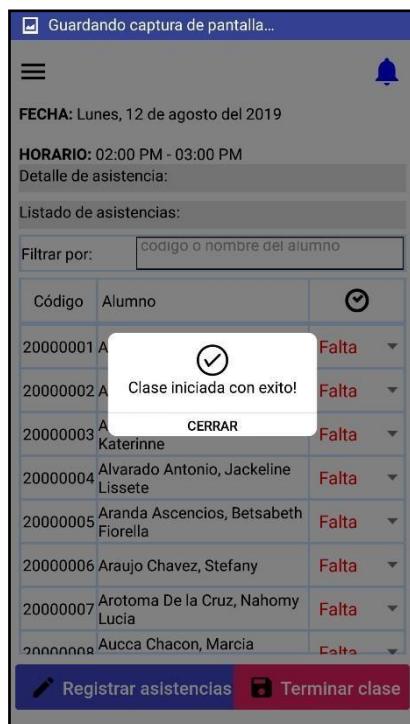


Figura 91.Interfaz del profesor:Visualizar clases de hoy – Iniciar clase (Elaboración propia)

Luego de haber registrado la hora de inicio de la clase, el docente puede registrar la asistencia de sus alumnos. El docente escoge alguno de los siguientes estados para la asistencia de un alumno: “Temprano”, “Tarde” y “Falta”. También se permite realizar filtrado de alumnos ya sea por código o nombre de alumno. (véase Figura 92)

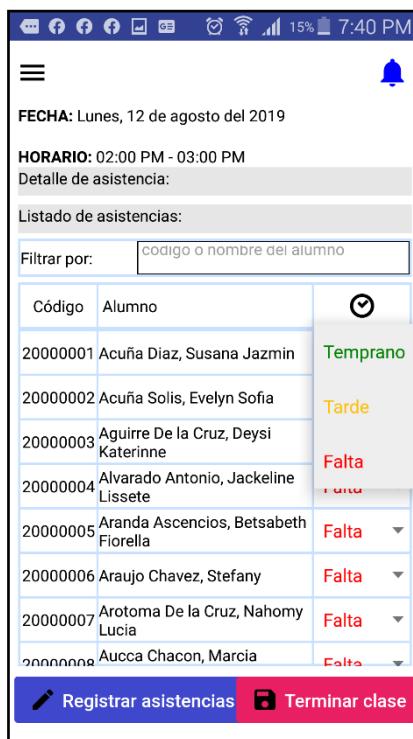


Figura 92.Interfaz del profesor:Visualizar clases de hoy – Modificar asistencias de alumnos (Elaboración propia)

Luego de haber realizado las modificaciones correspondientes cuando el profesor desee guardar los cambios debe indicar la opción Registrar asistencias. (véase Figura 93)

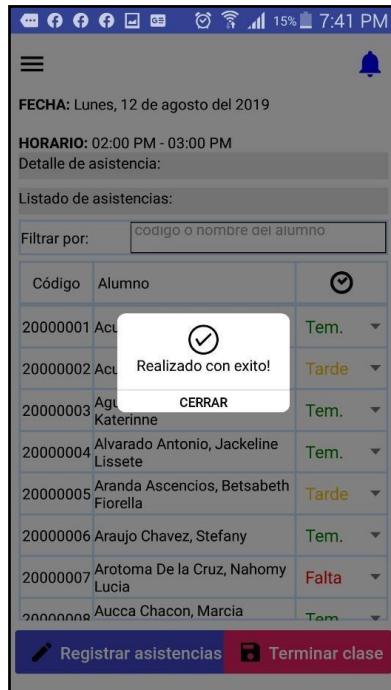


Figura 93.Interfaz del profesor: Visualizar clases de hoy – Registrar modificación de asistencias de alumnos (Elaboración propia)

Para terminar la clase, el docente debe indicar la opción: “Terminar clase”. Luego de esto la clase estará terminada y no se podrá modificar sus asistencias. (véase Figura 94)



Figura 94.Interfaz del profesor: Visualizar clases de hoy – Terminar clase(Elaboración propia)

Al hacer clic en la opción “Asistencias” del menú lateral, el sistema muestra el listado de cursos correspondientes al docente autenticado. (véase Figura 95)

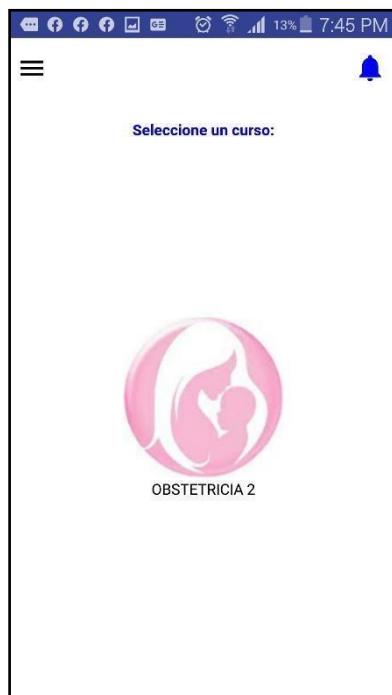


Figura 95.Interfaz del profesor: Visualizar asistencias – Listar cursos (Elaboración propia)

Se elige un curso y el sistema muestra el listado de agrupaciones curriculares correspondientes al curso seleccionado. Se le muestra todos los grupos dada la relación de clases que el profesor se encuentra asignado, el profesor principal tiene la opción de ver todos los grupos. (véase Figura 96)

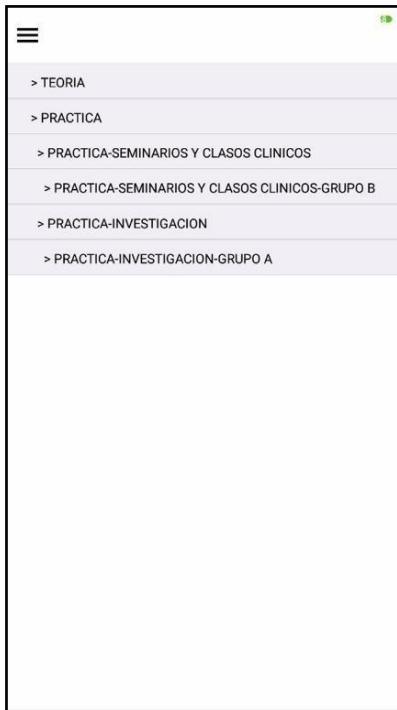


Figura 96.Interfaz del profesor:Visualizar asistencias – Listar agrupaciones curriculares
(Elaboración propia)

Luego de seleccionar una agrupación curricular, el sistema muestra el listado de clases que le corresponde, es decir, todas las clases ejercidas a lo largo del semestre que corresponden a dicha agrupación curricular. En color blanco figuran las clases pendientes, en color amarillo las clases iniciadas, por último, en color verde las clases finalizadas. (véase Figura 97)



Figura 97. Interfaz del profesor: Visualizar asistencias – Listarclases de una agrupación curricular (Elaboración propia)

Al seleccionar alguna de las clases listadas anteriormente, el sistema muestra el detalle de la clase, así como el listado de alumnos con sus respectivas asistencias a modo de consulta. (véase Figura 98)

FECHA:	Lunes, 12 de agosto del 2019	
HORARIO:	02:00 PM - 03:00 PM	
Detalle de asistencia:		
Clase Inaugural		
Previsión de infecciones intrahospitalarias		
<u>Detalle tema:</u>		
Importancia del curso. Objetivos. Metodología de enseñanza. Desarrollo del curso. Evaluaciones. Pre Test		
<u>Profesor asignado:</u>	<u>Control de la clase.</u>	
Clara Díaz Tinoco	CLASE FINALIZADA	
<u>Hora Inicio:</u>	<u>Hora Fin:</u>	
Listado de asistencias:		
Filtrar por:		
Código	Alumno	⊗
20000001	Diaz Diaz, Susana Jazmin	Tem. ▾
20000002	Solis Solis, Evelyn Sofia	Tarde ▾
20000003	De la Cruz De la Cruz, Deysi Katerinne	Tem. ▾
20000004	Antonio Antonio, Jackeline Lissette	Tem. ▾

Figura 98. Interfaz del profesor: Visualizar asistencias – Detalle de la clase (Elaboración propia)

Al hacer clic en la opción “Calificaciones” del menú lateral, el sistema muestra el listado de cursos correspondientes al docente autenticado similar a asistencias. (véase Figura 95) y se listan las agrupaciones curriculares (véase Figura 96)

Al seleccionar una agrupación curricular, el sistema muestra el detalle de las diferentes calificaciones con las que serán evaluados los alumnos. Es decir, que variables intervienen en el promedio específico de dicho grupo seleccionado, así como sus respectivos pesos. Además, se muestra el listado de calificaciones(código, nombre completo y calificación), se puede filtrar por el código o nombre del alumno. (véase Figura 99)

La captura de pantalla muestra una interfaz web móvil con un encabezado que incluye iconos para cámara, video, Wi-Fi, señal móvil y batería (9%), y la hora 7:53 PM. En la parte superior, hay dos pestañas: "NOTAS ESPECÍFICAS" (en azul) y "PROMEDIO GENERAL". Abajo de estas pestañas, se detallan los datos del curso: "CURSO: OBSTETRICIA 2", "GRUPO: TEORIA(40%)", "FÓRMULA: EXP" y la fórmula matemática "EXP = (EXP1 + EXP2 + EXP3 + EXP4)/4". Se presentan dos desplegables: "EXAMEN PARCIAL" y "EXAMEN PARCIAL 1". A continuación, hay un campo de búsqueda con la etiqueta "Filtrar por: código u nombre de alumno". Una lista de alumnos es mostrada en una tabla:

Código	Nombre	Calificación
20000001	Acuña Diaz, Susana Jazmin	0
20000002	Acuña Solis, Evelyn Sofia	0
20000003	Aguirre De la Cruz, Deysi Katerinne	0
20000004	Alvarado Antonio, Jackeline Lissete	0
20000005	Aranda Ascencios, Betsabeth Fiorella	0
20000006	Araujo Chavez, Stefany	0

En la parte inferior, hay dos botones: "Deshacer" (naranja) y "Guardar" (verde).

Figura 99. Interfaz del profesor: Visualizar calificaciones - Listado de calificaciones según variable específica seleccionada (Elaboración propia)

Para el registro de calificaciones correspondientes a una variable específica, el docente debe seleccionar una combinación de variables con los desplegables previamente mostrados. Luego debe digitar las notas correspondientes a cada alumno a través de las cajas de texto. (véase Figura 100)

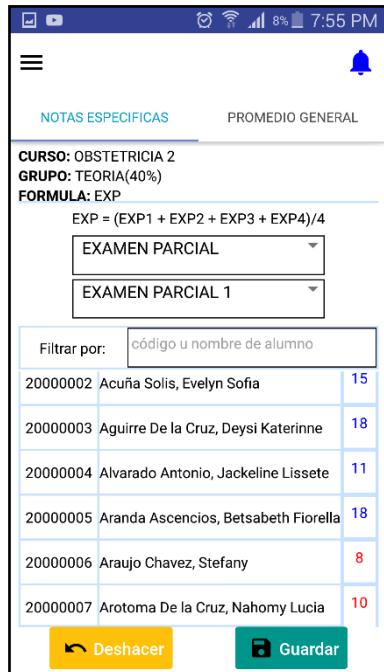


Figura 100.Interfaz del profesor: Visualizar calificaciones – Modificar listado de calificaciones(Elaboración propia)

Finalmente, confirmar los cambios realizados con la opción: “Guardar”. Además, se presenta la opción de deshacer los cambios realizados en memoria (calificaciones modificadas sin haber hecho clic en “Guardar”). (véase Figura 101)

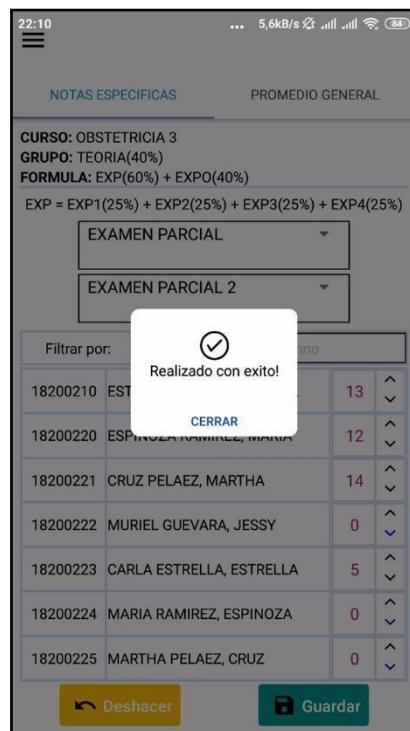


Figura 101.Interfaz del profesor:Visualizar calificaciones – Registro de calificaciones(Elaboración propia)

Si se elige el submenú promedio general se muestra el listado de alumnos que se puede filtrar por nombre o código para ver su promedio general. (véase Figura 102)



Figura 102.Interfaz del profesor: Visualizar promedio general (Elaboración propia)

Si se elige a un alumno, se puede ver el promedio total y cada sub-promedio (nota, porcentaje de peso) (véase Figura 103).



Figura 103.Interfaz del profesor: Visualizar promedio general por alumno (Elaboración propia)

Para listar las preguntas y/o respuestas se listan primero los cursos, al igual que los anteriores (véase Figura 95), se selecciona el curso y se listan los grupos (véase Figura 96), por último, se listan las clases (véase Figura 97) y se elige una clase. Al haber elegido esa clase se muestra la relación de preguntas relacionadas a la clase. (véase Figura 104)

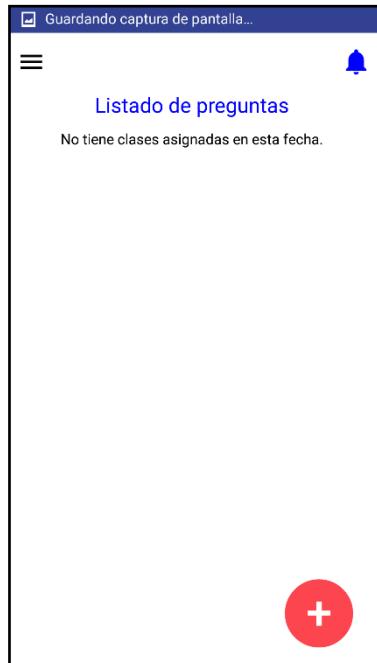


Figura 104.Interfaz del profesor: Visualizar preguntas (Elaboración propia)

Al presionar la opción + se visualiza la interfaz en la que se registra la pregunta que se va a publicar, para hacerlo se debe indicar la opción “Publicar pregunta”. (véase Figura 105)

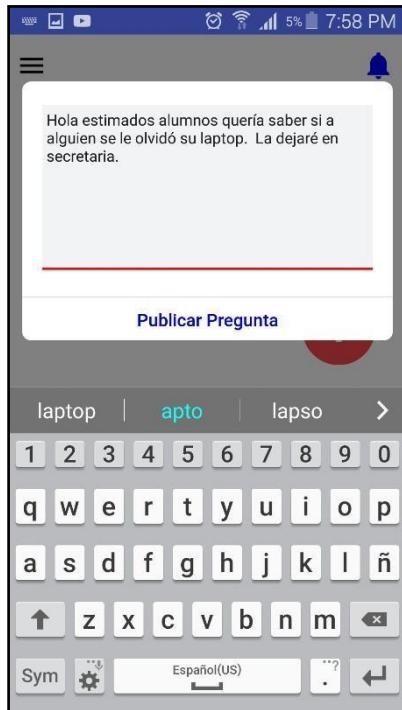


Figura 105.Interfaz del profesor: Publicar pregunta (Elaboración propia)

Luego se vuelve al listado de preguntas en la que se puede visualizar la persona que publicó hace cuento tiempo lo hizo e información del curso y grupo. También la opción para agregar respuestas adicionales a la pregunta en la que pueden participar alumnos o profesores. (véase Figura 106)

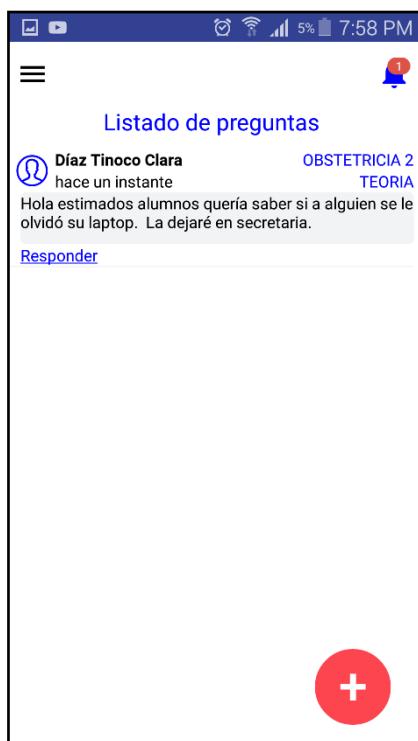


Figura 106.Interfaz del profesor: Pregunta publicada (Elaboración propia)

Ahora se muestra que el alumno hizo una respuesta, y la directora interactúa con ella con una respuesta. (véase Figura 107)

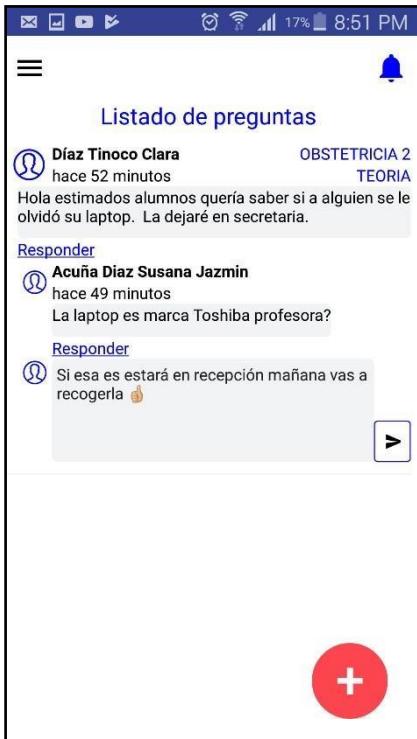


Figura 107.Interfaz del profesor: Realizar respuesta (Elaboración propia)

Luego se muestra la pregunta registrada (véase Figura 108)

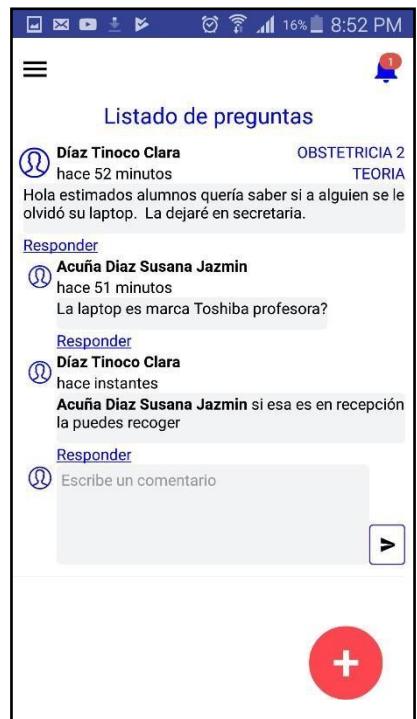


Figura 108.Interfaz del profesor: Respuesta realizada (Elaboración propia)

Para listar los anuncios se listan primero los cursos, al igual que los anteriores (véase Figura 95), se selecciona el curso y se listan los grupos (véase Figura 96) y se elige un

grupo. Al haber elegido el grupo se muestra la relación de preguntas relacionadas. (véaseFigura 109)

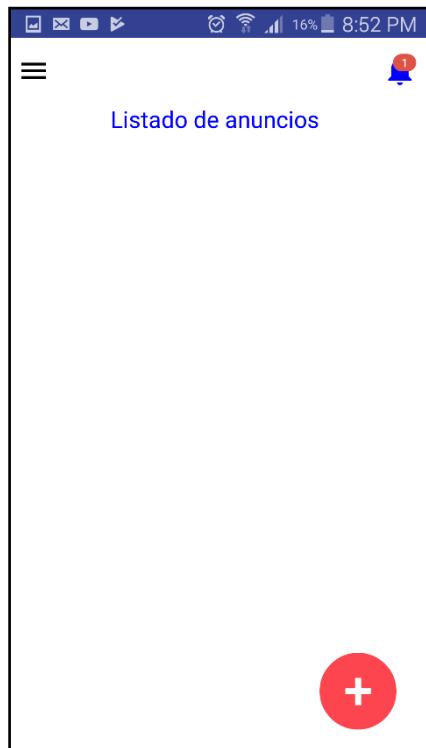


Figura 109.Interfaz del profesor: Visualizar anuncios. (Elaboración propia)

Al indicar la opción + se puede publicar un anuncio en el grupo se muestra los campos (título y descripción). Al finalizar indicar la opción “Publicar anuncio” (véaseFigura 110).



Figura 110.Interfaz del profesor: Publicar anuncio (Elaboración propia)

Se puede ver el anuncio publicado con su título y su respectiva descripción. (véase Figura 111)

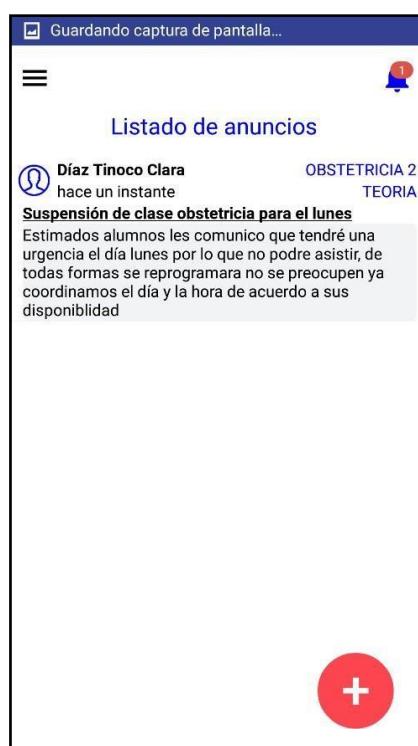


Figura 111.Interfaz del profesor: Anuncio publicado (Elaboración propia)

Para consultar la fórmula del grupo se listan primero los cursos al igual que los anteriores (véase Figura 95), se selecciona el curso y se listan los grupos (véase Figura 96) y se elige un grupo. Al haber elegido el grupo se muestra la fórmula del grupo que puede ser personalizada o no personalizada, se puede personalizar la fórmula al agregar nuevas variables. (véaseFigura 112)

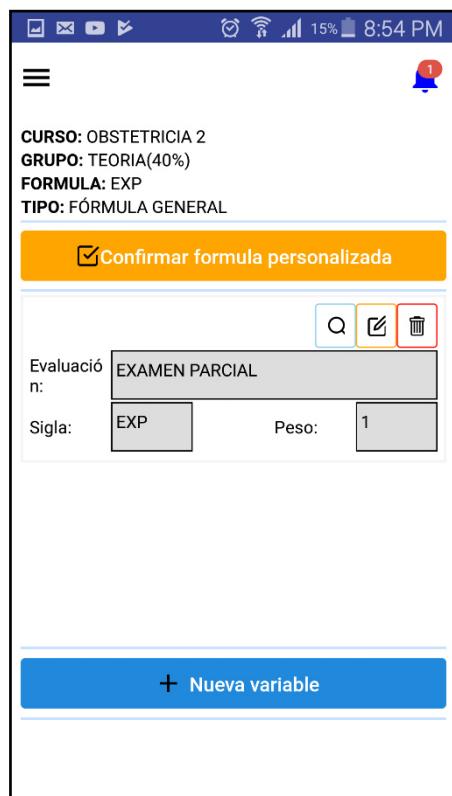


Figura 112.Interfaz del profesor: Ver fórmula de grupo (Elaboración propia)

Se indica agregar nueva variable donde se ingresa (nombre de evaluación, sigla, peso), luego de eso debe detallar su composición con la opción de lupa (ver detalle). También hay otras opciones como modificar o eliminar en caso ya no se use esa su subfórmula (véaseFigura 113)

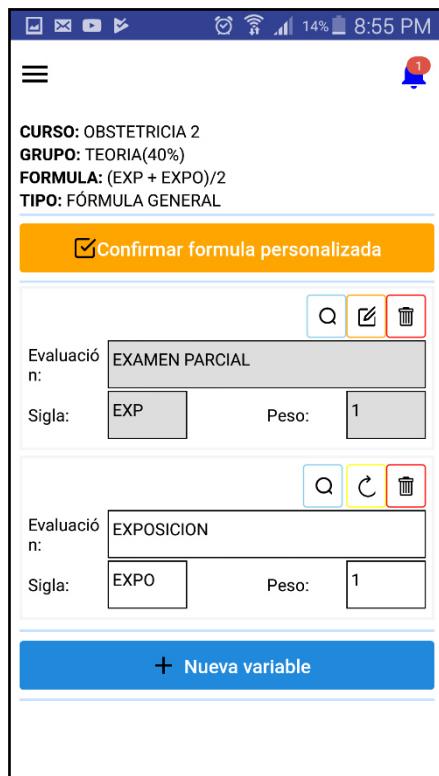


Figura 113.Interfaz del profesor:Personalizar fórmula (Elaboración propia)

Luego de haber indicado la opción de ver el detalle de la fórmula personalizada se puede indicar cada evaluación detallada con la información (nombre de evaluación, sigla, peso). (véase Figura 114)

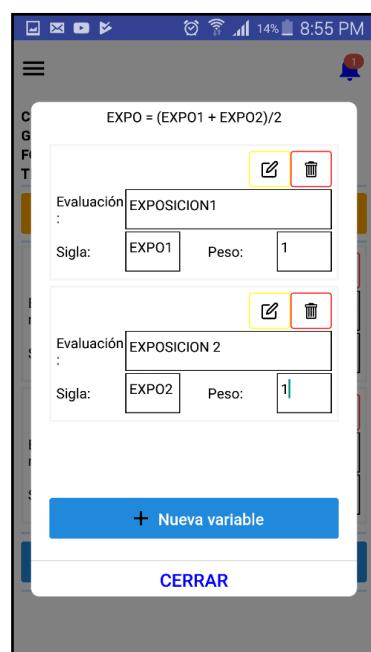


Figura 114.Interfaz del profesor:Personalizar subfórmulas(Elaboración propia)

Por último, se vuelve a la interfaz anterior e indica confirmar fórmula personalizada, se muestra un mensaje de confirmación del cambio, se reinician las notas que hayan sido ingresadas con anterioridad. (véaseFigura 115)

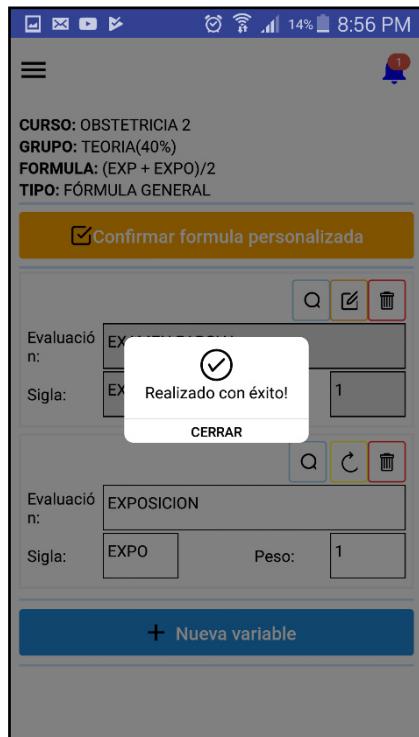


Figura 115.Interfaz del profesor: Fórmula confirmar personalización (Elaboración propia)

En esta interfaz se puede ver que la fórmula ya es personalizada para el grupo, y se puede seguir modificando o volver a la fórmula general que era igual para todos los grupos. (véaseFigura 116)

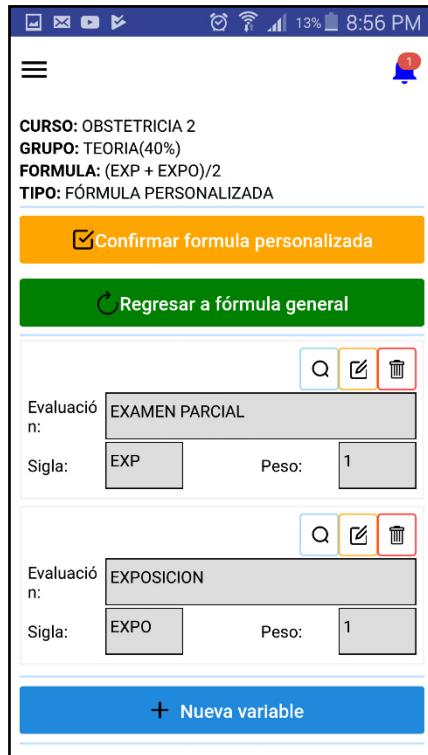


Figura 116.Interfaz del profesor: Fórmula correctamente personalizada (Elaboración propia)

4.2.3.3.3 Módulo del alumno

Menúalumno

Por defecto se listan las clases que corresponden al presente día, ordenadas a través del horario. Muy similar al GUI de los docentes, pero a diferencia, se indica el estado de las asistencias del alumno autenticado (Clase no iniciada, Asistió, Tardanza, Inasistencia). (véase Figura 117)

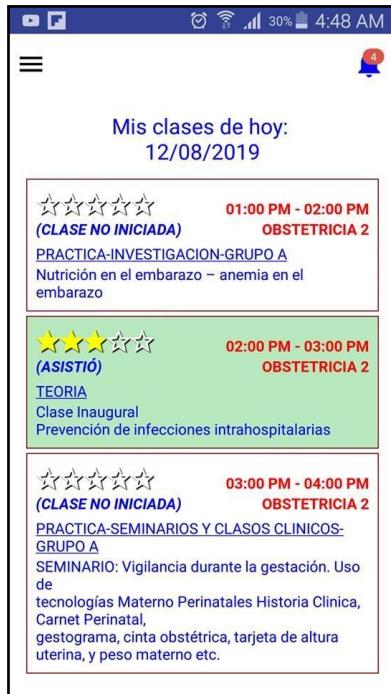


Figura 117.Interfaz del alumno: Visualizar las clases de hoy (Elaboración propia)

Al deslizar de izquierda a derecha, se puede visualizar un menú lateral, donde se encuentran las opciones disponibles para el docente (Clases del presente día, control de asistencia, control de calificaciones, preguntas y respuestas, etc.). (véase Figura 118)

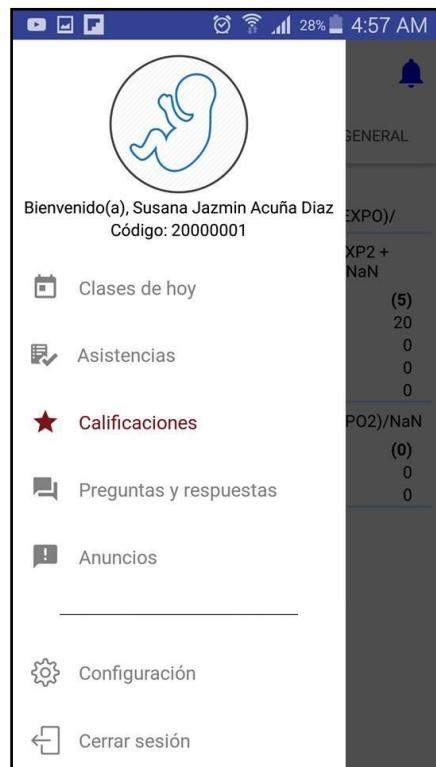


Figura 118.Interfaz del alumno: Visualizarmenúlateral (Elaboración propia)

Luego de seleccionar una clase en específico, el sistema muestra las secciones; en una de ellas, se muestra el detalle de la clase, tales como, nombre del docente, estado, horas de inicio y fin, valoración, además se añade el estado de la asistencia del propio alumno autenticado. En la otra sección, siempre y cuando la clase al menos haya comenzado, se muestra la encuesta con un total de 10 preguntas cada una de ellas en un rango de 1 a 5 estrellas. (véase Figura 119)

FECHA: Lunes, 12 de agosto del 2019

HORARIO: 02:00 PM - 03:00 PM

Detalle de asistencia:

Nombre tema:
Clase Inaugural
Prevención de infecciones intrahospitalarias

Detalle tema:
Importancia del curso. Objetivos. Metodología de enseñanza. Desarrollo del curso. Evaluaciones. Pre Test

Encuesta de la clase:

Listado de preguntas		
1	Demuestra dominio de la asignatura	★★★★★
2	Explica con claridad	★★★★★
3	Absuelve las preguntas de los estudiantes sobre la asignatura	★★★★★
4	Asistió puntualmente	★★★★★
5	Demuestra valores éticos en su actuar diario	★★★★★
6	Su trato es respetuoso y cordial	★★★★★
7	Promueve la participación activa de los alumnos	★★★★★

Finalizar encuesta

Figura 119. Interfaz del alumno: Visualizarencuesta (Elaboración propia)

Para realizar la encuesta de satisfacción, el alumno debe valorizar cada pregunta dentro del rango de estrellas establecido. Para finalmente, confirmar sus puntuaciones a través de la opción: “Finalizar encuesta”. (véase Figura 120)

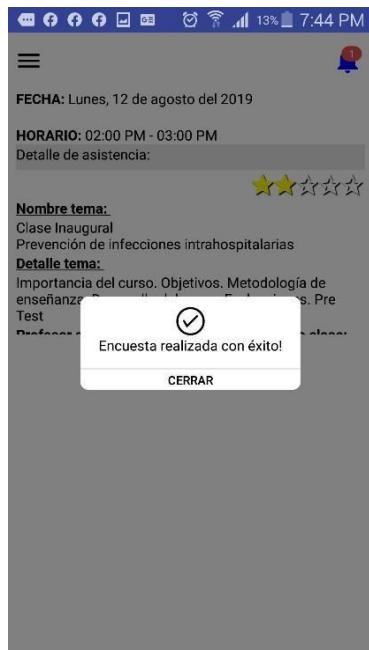


Figura 120.Interfaz del alumno:Encuesta realizada (Elaboración propia)

Al hacer clic en la opción “Asistencias” del menú lateral, el sistema muestra el listado de cursos correspondientes al alumno autenticado. (véase Figura 121)



Figura 121.Interfaz del alumno:Visualizar asistencias – Listar cursos (Elaboración propia)

Se selecciona un curso y el sistema muestra el listado de agrupaciones curriculares correspondientes al curso previamente seleccionado. (véase Figura 122)

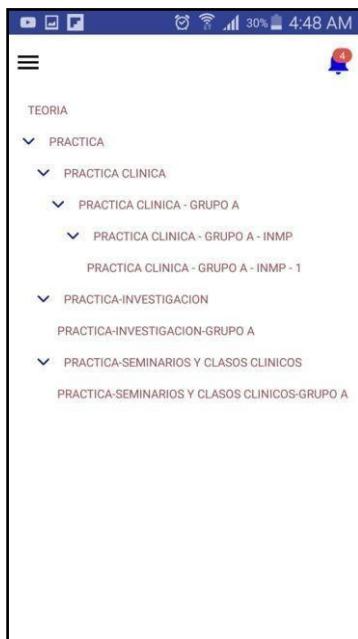


Figura 122.Interfaz del alumno: Visualizar asistencias – Listar agrupaciones curriculares (Elaboración propia)

Luego de seleccionar una agrupación curricular, el sistema muestra el listado de clases que le corresponde, es decir, todas las clases ejercidas a lo largo del semestre que corresponden a dicha agrupación curricular. (véase Figura 123)



Figura 123.Interfaz del alumno:Visualizar asistencias – Listado de clases de una agrupación curricular (Elaboración propia)

Al seleccionar alguna de las clases listadas anteriormente, el sistema muestra el detalle de la clase, con las características mencionadas en párrafos anteriores. Esta limitado por

espacio, pero se puede deslizar y ver información detallada de la asistencia como (Promedio de calificación, nombre de tema, detalle de tema, profesor asignado, estado de la clase, asistencia del alumno, hora de inicio de la clase y hora de fin de la clase) (véaseFigura 124)

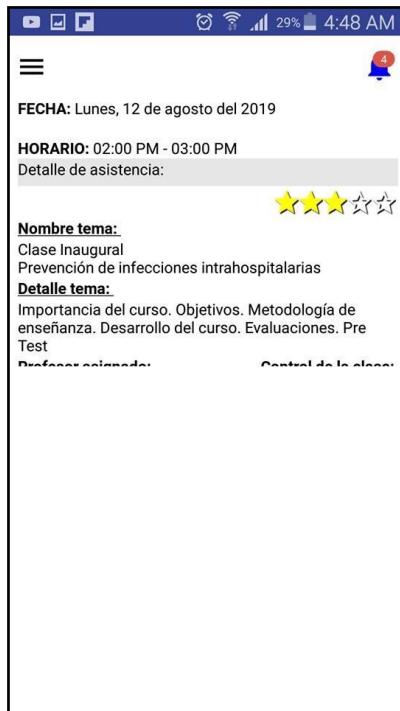


Figura 124.Interfaz del alumno: Visualizar asistencias – Detalle de la clase
(Elaboración propia)

Al hacer clic en la opción “Calificaciones” del menú lateral, el sistema muestra el listado de cursos correspondientes al alumno autenticado similar al anterior listado de cursos en asistencias. (véaseFigura 121). Luego el sistema muestra el listado de agrupaciones curriculares correspondientes al curso previamente seleccionado similar al listado de grupos de asistencias (véaseFigura 122)

Al seleccionar una agrupación curricular, el sistema muestra los valores de cada una de las variables que intervienen en el promedio específico, así como la fórmulaespecífica y el valor del promedio específico. (véaseFigura 125)

NOTAS ESPECIFICAS		PROMEDIO GENERAL
CURSO: OBSTETRICIA 3		
GRUPO: TEORIA(40%)		
FORMULA: EXP(60%) + EXPO(40%)		
EXP = EXP1(25%) + EXP2(25%) + EXP3(25%) + EXP4(25%)		
EXAMEN PARCIAL	(2.5)	
EXAMEN PARCIAL 1 (EXP1)	10	
EXAMEN PARCIAL 2 (EXP2)	0	
EXAMEN PARCIAL 3 (EXP3)	0	
EXAMEN PARCIAL 4 (EXP4)	0	
EXPO = EXPO1(25%) + EXPO2(25%) + EXPO3(25%) + EXPO4(25%)		
EXPOSICIONES	(0)	
EXPOSICION 1 (EXPO1)	0	
EXPOSICION 2 (EXPO2)	0	
EXPOSICION 3 (EXPO3)	0	
EXPOSICION 4 (EXPO4)	0	
TOTAL:	1.5	

Figura 125.Interfaz del alumno:Visualizar calificaciones – Listado de calificaciones
(Elaboración propia)

En el caso de las preguntas y respuestas se realiza de la misma forma que la hace el profesor. Para listar preguntas (véaseFigura 104), para publicar pregunta (véaseFigura 105), para ver la pregunta publicada (véaseFigura 106), para realizar la respuesta (véaseFigura 107), para ver la respuesta realizada (véaseFigura 108).

Para el caso de los anuncios el alumno no tiene permitido realizar anuncios, pero si consultarlos(véaseFigura 126).

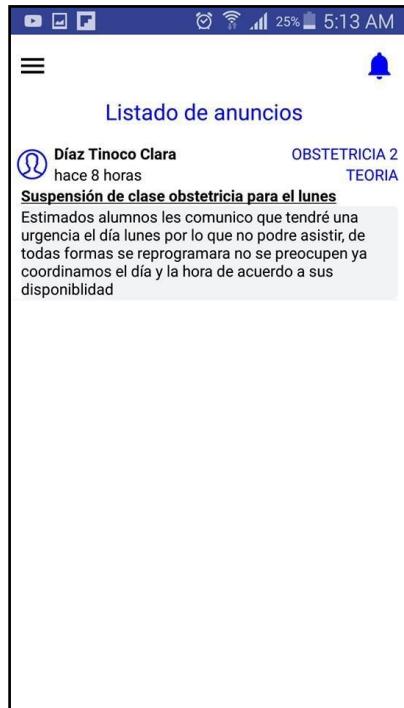


Figura 126.Interfaz del alumno: - Visualizar anuncios (Elaboración propia)

La consulta de notificaciones se puede realizar para cualquier usuario en la opción de campana en la cual al presionarla se listan todas las notificaciones. (véase Figura 127)

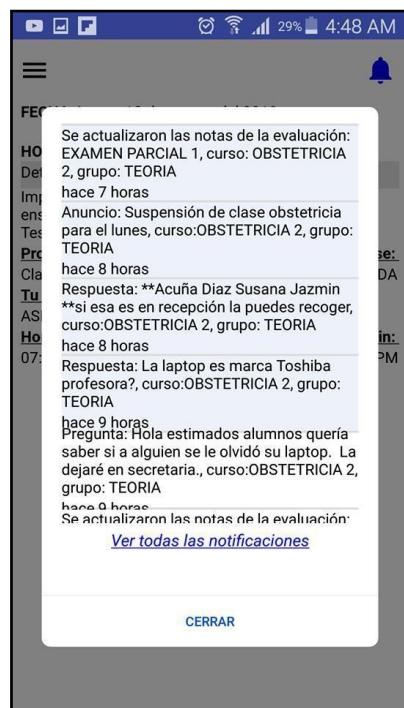


Figura 127.Interfaz del alumno: - Ver notificaciones (Elaboración propia)

4.2.3.3.4 Módulo director(a) de escuela

Menú director(a) de escuela

A continuación, se muestran las opciones del menú(véase Figura 128) al deslizar a la izquierda o presionando el ícono de menú.

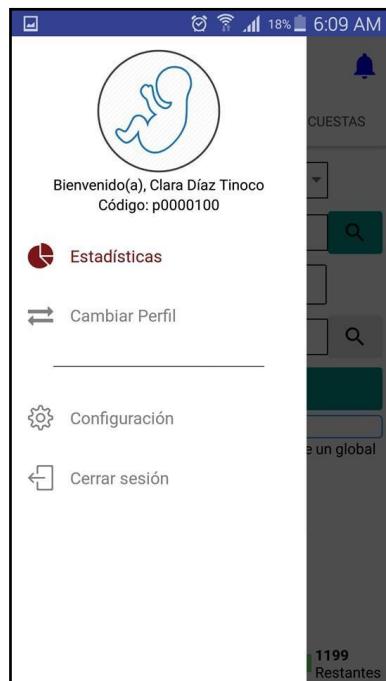


Figura 128. Interfaz de director(a) de escuela – Visualizar menú lateral (Elaboración propia)

Esta interfaz muestra 3 submenús (asistencias notas y encuesta), por defecto está seleccionado asistencias en el cual se pueden ver varios filtros para personalizar las estadísticas, como filtrar por curso en la que se selecciona un curso, luego los grupos de ese curso, el buscar por indica si se mostrarán estadísticas de alumnos o profesores, y por último buscar por alumno en el que se filtra por el alumno inscrito en un determinado grupo. En la parte de estadística se muestra el avance con respecto al total de asistencias que debería tener ya sea alumno o profesor, y de la misma forma se muestra un gráfico estadístico circular que indica los tipos de asistencias (a tiempo, tardanza y falta como también las asistencias restantes) (véase Figura 129)

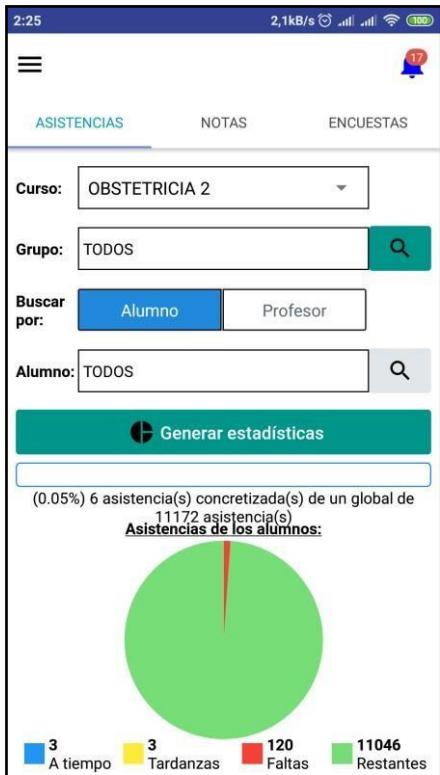


Figura 129. Interfaz dedirector(a) de escuela: Visualizar estadísticas de asistencias
 (Elaboración propia)

En esta interfaz se puede apreciar las estadísticas de las notas en forma de gráfica circular (las subdivisiones están comprendidas entre el promedio de notas obtenidas como 0-5,6-10,11-15, 16-20), se puede filtrar por curso, grupo y alumnosen ese orden. (véaseFigura 130)



Figura 130. Interfaz dedirector(a) de escuela - Visualizar estadísticas de calificaciones
(Elaboración propia)

En esta interfaz se puede apreciar las estadísticas de las encuestas en base a las preguntas en la que se puede observar un promedio general, y un promedio por cada pregunta de la encuesta, además se puede filtrar por curso, grupo y profesor en ese orden. (véase Figura 131)

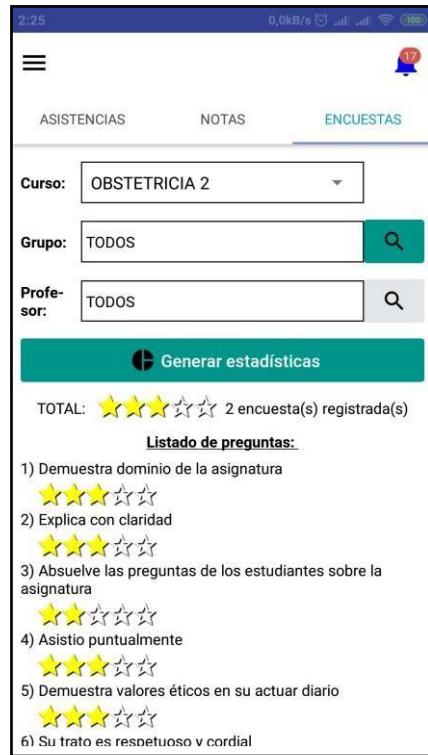


Figura 131. Interfaz dedirector(a) de escuela:Visualizar estadísticas de encuestas
(Elaboración propia)

5 CONCLUSIONES

- Este sistema propuesto sobre el control académico cumplió con los objetivos planteados inicialmente en base al problema definido relacionado a la EP Obstetricia UNMSM, éste mismo fue aprobado por la dirección de escuela (véase Anexo 2), que abarcan los siguientes puntos:
 - Permite el control de asistencias de clases de los profesores.
 - Permite el control de asistencias de clases de los alumnos.
 - Brinda la medición del desempeño del profesor en base a encuestas realizadas de cada clase por los alumnos.
 - Permite el control de notas de los alumnos.
 - Brinda estadísticas e indicadores a la dirección de escuela sobre asistencias, notas y encuestas.
 - Proporciona un medio interactivo que facilite la comunicación entre alumnos y profesores que se consiguió mediante los anuncios, las preguntas y respuestas.
 - Brinda notificaciones a los usuarios en tiempo real.
 - La información puede estar disponible todo el día para todos.
 - La información se muestra de forma centralizada y ordenada mediante el acceso al sistema.
 - El tiempo de búsqueda de la información es menor al que se hacía anteriormente debido a que ésta se encuentra accesible con mayor facilidad desde cualquier smartphone en cualquier lugar.
- El uso de la tecnología reactnative permite mayor versatilidad que enfocarse solo a un tipo de aplicativo móvil, ya que puede crear aplicativos para Android como IOS a la vez y así poder abarcar una mayor cantidad de usuarios.
- El tipo de arquitectura es bastante robusto, es flexible, permite el uso de componentes independientes, pero también conlleva otros desafíos como la de manejar información compuesta de varios servicios, pruebas integrales y despliegue.
- Es importante mencionar que el equipo responsable del desarrollo de este tipo de aplicativos trabaje de forma coordinada y que el equipo conozca los principios que rigen esta arquitectura.

Trabajos futuros

- Hacer que el sistema abarque todos los cursos del periodo académico actual en la EP de Obstetricia de la UNMSM.
- Hacer el sistema disponible para otras facultades de la UNMSM u otras universidades.
- Agregar al sistema el desarrollo web del aplicativo en el lado frontend (lado cliente).

6 BIBLIOGRAFÍA

- Abernethy, M. A., & Chua, W. F. (1996). A Field Study of Control System "Redesign": The impact of institutional Processes on Strategic Choice. *Contemporary Accounting Research*, 13(2), 569-606. doi:<https://doi.org/10.1111/j.1911-3846.1996.tb00515.x>
- Acevedo, Y. L. (2018). *IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA MEJORA DEL PROCESO ADMINISTRATIVO ACADÉMICO DE LA INSTITUCIÓN EDUCATIVA "WARI-VILCA"-HUAYUCACHI (Tesis de Pregrado)*. Universidad Nacional del Centro del Perú, Ingeniería de Sistemas, Huancayo.
- Alférez, A. J. (2018). *Estudio comparativo de frameworks multiplataforma para desarrollo de aplicaciones móviles (Tesis Postgrado)*. Universidad Stuttgart. Obtenido de <http://oa.upm.es/53692/>
- Álvarez, R. B., & Mayo, I. C. (2009). Las tecnologías de la información y la comunicación en la educación superior. *Revista Iberoamericana de Educación*, 1-12.
- Association for Computing Machinery. (2019). *The ACM Computing Classification System (CCS)*. Recuperado el 6 de Junio de 2019, de Association for Computing Machinery: https://dl.acm.org/ccs/ccs_flat.cfm
- Berrospi, R. A., & Pilar, J. M. (2013). *Implementación de un sistema web para optimizar la gestión académica en la I.E. "Villa corazón de Jesús" del distrito San Juan de Lurigancho (Tesis de pregrado)*. Universidad de Ciencias y Humanidades, Facultad de Ciencias e Ingeniería, Escuela Profesional de Ingeniería de Sistemas e Informática, Lima.
- Beuren, I. M., & Teixeira, S. A. (2014). Evaluation of management control systems in a higher education institution with the performance management and control. *Scielo*. doi:<http://dx.doi.org/10.4301/S1807-17752014000100010>
- Branstein, M., & Branstein, N. (2017). *The NativeScript Book building mobile apps with skills you already have*.
- Brito, H., Gómez, A., Santos, A., & Bernardino, J. (2018). JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development. IEEE. *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6. Obtenido de <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8399283&isnumber=8398632>
- Brito, J. A. (2016). Comparación de metodologías y procesos de desarrollo de software mediante un instrumento basado en CMMI. *Scientia et Technica Año XXI*, 21, 150-155.
- Cañaveral, A. G. (2012). *Re-diseño de la arquitectura del ERP de fuente abierta Adempiere, utilizando el marco de referencia SOA, personalización con Spring Frameworks e inyección de dependencias (Tesis de Pregrado)*. Universidad de Las Américas, Facultad de Ingeniería y Ciencias Aplicadas. Quito: Universidad de Las Américas. Obtenido de <http://dspace.udla.edu.ec/handle/33000/2588>

- Cerny, T., Donahoo, M., & Pechanec, J. (2017). Disambiguation and Comparison of SOA, Microservices and Self-Contained Systems. *RACS '17 Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, 228-235.
doi:10.1145/3129676.3129682
- Chacón, L. (2016). *Diseño e implementación de una app sobre desarrollo sostenible con back-end de arquitectura basada en microservices y de una react native front-end app (Tesis de Pregrado)*. Universitat Politècnica de Catalunya, Barcelona. Obtenido de <http://hdl.handle.net/2117/97021>
- Christoph, J., Rösch, D., & Schuster, T. (2018). Cross-Platform Development Suitability of Current Mobile Application Frameworks. *The Eighth International Conference on Advanced Collaborative Networks, Systems and Applications*, 13-20. Obtenido de https://www.thinkmind.org/index.php?view=article&articleid=colla_2018_2_10_50017
- COMEXPERU. (2011). Educación Superior: un diagnóstico. *Semanario COMEXPERU*(636), 5-6. Recuperado el 9 de Agosto de 2016, de <http://www.comexperu.org.pe/media/files/semanario/SEMANARIO%20COMEXPERU%20636.PDF>
- De la Cruz, P. E., Espinoza, M. H., & Cuba, O. (2019). Propuesta de arquitectura de microservicios, metodología Scrum. *Hamut'ay*, 141-158.
doi:<http://dx.doi.org/10.21503/hamu.v6i2.1781>
- Delgadillo, R., & Fermín, A. (2018). Acerca de computación y sus líneas de investigación. *Revista Peruana de Computación y Sistemas* 2018, 3-8.
doi:<https://doi.org/10.15381/rpcs.v1i1.14852>
- Eisenman, B. (2016). *Learning React Native (First Edition)*. (M. Foley, Ed.) United States of America: O'Reilly Media Inc.
- Erl, T. (2019). *SOA: Principles of Service Design*.(P. Hall, Ed.) Recuperado el 23 de junio de 2019, de Arcitura: https://patterns.arcitura.com/wp-content/uploads/2019/03/SOA_Principles_Poster.pdf
- Facultad de Ingeniería de Sistemas e Informática UNMSM. (2019). *Programas y Líneas de Investigación*. Recuperado el 21 de junio de 2019, de Facultad de Ingeniería de Sistemas e Informática UNMSM: sistemas.unmsm.edu.pe/investigacion/unidad/lneas-de-investigacion
- Fernandez, J. M., & Cadelli, S. (2014). *Convivencia de metodologías: Scrum y Rup en un proyecto de gran escala (Tesina de Licenciatura)*. UNLP, Facultad de Informática. Ciudad de La Plata, Argentina: UNLP. Obtenido de <http://hdl.handle.net/10915/47082>
- Ferreira, A., & Otley, D. (2006). The design and use of performance management systems: an extended framework for analysis. *Management Accounting Research*, 20(4), 263-282.
- Flor, L. D. (2014). *Sistema de seguimiento de syllabus (Tesis de Pregrado)*. Quito, Ecuador: UISRAEL. Obtenido de <https://repositorio.uisrael.edu.ec/handle/47000/938>

- Flutter Official Documentation.* (2018). Obtenido de <https://flutter.io/>
- Fowler, M., & Lewis, J. (s.f.). *Microservices Resource Guide*. Recuperado el 09 de Junio de 2016, de Martin Fowler: <https://martinfowler.com/microservices/>
- Freire, W. A. (2015). *Sistema de seguimiento Estudiantil (Tesis de Pregrado)*. Universidad Central del Ecuador, Facultad de Ingeniería, Ciencias Físicas y Matemática: Carrera de Ingeniería Informática. Quito, Ecuador: Universidad Central del Ecuador. Obtenido de <http://www.dspace.uce.edu.ec/handle/25000/5475>
- Ganchoso, J., & Vera, G. (2015). *Sistema Web de Gestión Académica en el Centro de Idiomas de la ESPAM MFL (Tesis de Pregrado)*. Calceta, Ecuador: ESPAMMFL. Obtenido de <http://repositorio.esspam.edu.ec/handle/42000/52>
- García, J. (2018). *Estudio comparativo de metodologías, herramientas y wiki de soporte para la gestión de desarrollo de software. Tesis de pregrado*. Universidad Oberta de Catalunya. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/73151/6/jgarcianavarroTFG0118memoria.pdf>
- Haimann, T. (2005). En *Dirección y gerencia: planificación, coordinación y control de las actividades de la empresa* (pág. 85).
- Horngren, C. T., Datar, S. M., & Foster, G. (2000). *Contabilidad de costos*. Rio de Janeiro: LTC.
- IEEE. (2019). *2019 IEEE Taxonomy - Version 1.0*. Recuperado el 18 de Junio de 2019, de IEEE: <https://www.ieee.org/content/dam/ieee-org/ieee/web/org/pubs/ieee-taxonomy.pdf>
- INEI. (2015). *Encuesta Nacional a Egresados Universitarios y Universidades 2014*. Lima. Obtenido de https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1298/Libro.pdf
- Ionic Framework Documentation*. (27 de 09 de 2019). Obtenido de Ionic Framework: <https://ionicframework.com/docs>
- Kalske, M., Mäkitalo, N., & Mikkonen, T. (2018). Challenges When Moving from Monolith to Microservice Architecture. En I. Garrigos, & M. Wimmer, *Current Trends in Web Engineering* (págs. 32-47). Springer. doi:https://doi.org/10.1007/978-3-319-74433-9_3
- Koontz, H., Weihrich, H., & Cannice, M. (2012). Sistema y Proceso de Control. En *Administracion una perspectiva Global y Empresarial 14.º ed.* (págs. 496-497). McGraw Hill.
- Krafzig, D., Banke, K., & Slama, D. (2004). *Enterprise SOA: Service-oriented Architecture Best Practices*. Prentice Hall.
- Kuz, A., Falco, M., & Giandini, R. (2016). *Los TICs como gestoras de cambio en las realidades educativas universitarias*. Recuperado el 2019 de Junio de 19, de http://extension.unicen.edu.ar/jem/subir/uploads/1230_2016.doc

- Lele, W. (2017). *Hackeroon*. Obtenido de What's Revolutionary about Flutter:
<https://hackeroon.com/whats-revolutionary-about-flutter-946915b09514>
- MarketShare Statistic for Internet Technologies*. (2018). Recuperado el 18 de 10 de 2019, de NetMarketShare: <https://netmarketshare.com/operating-system-market-share.aspx>
- Moromi, H. (2002). *La Influencia de la ejecución curricular y el uso de medios y materiales en el rendimiento académico de los estudiantes de la Facultad de Odontología de la Universidad Nacional Mayor de San Marcos (Tesis Postgrado)*. Universidad Nacional Mayor de San Marcos, Escuela de Postgrado - Unidad de Postgrado de la Facultad de Educación. Lima: Universidad Nacional Mayor de San Marcos. Obtenido de http://sisbib.unmsm.edu.pe/BibVirtual/Tesis/Human/Moromi_N_H/Moromi_N_H.htm
- Newman, S. (2015). *Building Microservices*. (M. L. MacDonald, Ed.) O'Reilly.
- Ospiva, A. (2019). *CROSS-PLATFORM MOBILE DEVELOPMENT FRAMEWORKS: CHOOSING THE MOST SUITABLE OPTION FOR SELECTED USE CASE*, Tesis de Maestría. Lappeenranta University of Technology, School of Engineering Science - Software Engineering. Obtenido de <http://urn.fi/URN:NBN:fi-fe2019062021448>
- Pardo, P. A., Triana, S., & Forero, N. G. (2014). UNA APROXIMACIÓN HOLÍSTICA A LAS METODOLOGÍAS ÁGILES DESDE LA PROGRAMACIÓN EXTREMA. *Revista Ingeniero Libre - Edición Nro. 13*, 36-45. Obtenido de <http://www.unilibre.edu.co/revistaingeniolibre/revista-13/r13a.pdf>
- Perez-Herrera, M. (2015). *Arquitecturas basada en microservicios*. Madrid, España: UPM.
- Pertcu, D., & Iordan, V. (2009). Understanding Service-Oriented Architectures in the classroom: From Web Services to Grid Services. En G. Papadopoulos, G. Wojtkowski, W. Wojtkowski, S. Wrycza, & J. Zupancic, *Information Systems Development: Towards a Service Provision Society* (págs. 831-838). Boston: Springer Science & Business Media. doi:https://doi.org/10.1007/b137171_87
- Porras, E. E. (2019). *Metodología Ágil Iconix en la calidad del producto de software (Tesis de postgrado)*. Lima. doi:<http://repositorio.unfv.edu.pe/handle/UNFV/2956>
- Quacquarelli Symonds. (2010). *QS University Rankings: Latin America 2011*. Recuperado el 6 de Junio de 2019, de Quacquarelli Symonds:
https://content.qs.com/supplement2011/Latin_American_supplement.pdf
- Quacquarelli Symonds. (2013). *QS University Rankings: the top universities latin america ranking 2014*. Recuperado el 6 de Junio de 2019, de <https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2014>
- Quacquarelli Symonds. (2015). *The top 300 Universities in Latin America 2016*. Recuperado el 6 de Junio de 2019, de QS University Rankings: :
<https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2016>

- Quacquarelli Symonds. (2017). *Top Ranking Latin America University*. Recuperado el 24 de junio de 2019, de Quacquarelli Symonds: <https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2018>
- Quacquarelli Symonds. (2019). *Top Ranking Latin Universities 2020*. Recuperado el 17 de Junio de 2019, de Quacquarelli Symonds: <https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2020>
- Quacquarelli Symonds. (2020). *Top Ranking Latin Universities 2021*. Obtenido de Quacquarelli Symonds: <https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2021>
- Rajesh, R. V. (2016). *Spring Microservices*. Packt Publishing.
- Real Academia Española. (2001). *Diccionario de la lengua española (23.ª ed.)*. Recuperado el 9 de Julio de 2016, de
http://buscon.rae.es/drae/SrvltConsulta?TIPO_BUS=3&LEMA=control
- Richardson, C. (2018). *Microservices Patterns: With Examples in Java*. Manning Publications.
- Rinaldi, B. (2016). *Making sense of the mobile development ecosystem*. Obtenido de Mobile Business Insights: <https://mobilebusinessinsights.com/2016/05/making-sense-of-the-mobile-development-ecosystem-part-1-the-mobile-web/>
- Rodriguez, C., & Herrera, L. (agosto de 2009). Análisis correlacional-predictivo de la influencia de la asistencia a clase en el rendimiento académico universitario. Estudio de caso en una asignatura. *Profesorado: Revista de currículum y formación del profesorado*, 13(2), 1-13. Recuperado el 09 de Marzo de 2017, de
<http://www.redalyc.org/articulo.oa?id=56711798017>
- Sarasty, H. F. (2015). *Documentación y análisis de los principales frameworks de arquitectura de software en aplicaciones empresariales (Tesis Doctoral)*. Universidad Nacional de la Plata(, Facultad de Informática. Obtenido de
<http://dspace.ups.edu.ec/handle/123456789/16588>
- Satei, M. (2019). *OTT Video-Oriented Mobile Applications Development Using Cross-Platform UI Frameworks*. Tesis de Postgrado. Obtenido de <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1343759&dswid=326>
- SINEACE. (2016). *Modelo de Acreditación para Programas de Estudios de Educación Superior Universitaria*. Lima, Perú. Recuperado el 15 de Julio de 2016, de
<https://www.sineace.gob.pe/wp-content/uploads/2016/03/Anexo-a-la-Resoluci%C3%B3n-N%C2%BA0022-2016-CDAH-P.pdf>
- SINEACE. (12 de Mayo de 2020). Obtenido de Buscador de Instituciones y Carreras Acreditadas: <https://app.sineace.gob.pe/buscador/acreditacion.aspx>
- Tolentino, L. A. (15 de 01 de 2014). *Desempeño didáctico y académico del docente relacionado a la satisfacción de los estudiantes del programa de Complementación Pedagógica de la Universidad Nacional Mayor de San Marcos, 2013-II (Tesis Postgrado)*. Universidad Nacional Mayor de San Marcos, Facultad de Educación. Lima-Perú: Universidad

Nacional Mayor de San Marcos. Obtenido de
<http://cybertesis.unmsm.edu.pe/handle/cybertesis/3981>

Torres, M., & Paz, K. (2006). Tamaño de una muestra para una investigación de mercado.
Boletín Electrónico vol 2 (2006), 1-13.

Yanada, G., Rivera, M., & Castro, J. (2012). *Educación Superior en el Perú: Retos para el Aseguramiento de la Calidad*. Perú: SINEACE. Obtenido de
<http://repositorio.minedu.gob.pe/handle/123456789/937>

Zona Económica. (2019). *Definición de Control*. Recuperado el 23 de junio de 2019, de Zona Económica: <https://www.zonaeconomica.com/control>

ANEXOS

Anexo 1: Encuestas a alumnos en Google Forms

Las encuestas se realizaron por internet mediante el uso de Google Forms, se compartió el enlace de la encuesta a los alumnos para que ellos puedan llenar la encuesta.

A continuación, se muestra el formato de encuesta (véanse Figura 132 y Figura 133) que fue enviada a los alumnos:

The form is titled "ENCUESTA DE CONTROL ACADÉMICO A ALUMNOS". It includes a note about the survey being anonymous and for academic evaluation. Questions are numbered 1 through 7, each with a specific instruction and a rating scale from 1 to 5 or Muy poco to Demasiado.

1. Código(opcional)

* Required

2. ¿Que tan lejos vive de su centro de estudios(campus universitario) y lugares relacionados como hospitales? *
Mark only one oval.
1 2 3 4 5
Muy Cerca _____ Muy Lejos _____

3. En que año se encuentra cursando en la universidad o indicar si es egresado. Ojo: La opción Internado es lo mismo que 5 año. *
Mark only one oval.
1 Año _____
2 Año _____
3 Año _____
4 Año _____
5 Año _____
Internado _____
Egresado menos de 1 año _____
Egresado menos de 2 años _____
Egresado mas de 2 años _____

4. ¿(Cumplen/Cumplieron) los profesores con todos las clases del syllabus?
Mark only one oval.
1 2 3 4 5
Muy pocas _____ Todas sin excepción _____

5. ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las clases avanzadas por el profesor? *
Mark only one oval.
1 2 3 4 5
Muy fácil _____ Muy difícil _____

6. ¿Qué tan difícil (es/fue) llevar un control de asistencias que has realizado en tus cursos? *
Mark only one oval.
1 2 3 4 5
Muy fácil _____ Muy difícil _____

7. ¿Considera que (hay/hubo) retrasos en el tiempo en el que te enterabas de tus notas? *
Mark only one oval.
1 2 3 4 5
Muy poco _____ Demasiado _____

Figura 132. Primera parte del formato de la encuesta (Elaboración propia)

<p>8. ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las notas que has obtenido en tus cursos? *</p> <p><i>Mark only one oval.</i></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center; width: 10px;">1</td> <td style="text-align: center; width: 10px;">2</td> <td style="text-align: center; width: 10px;">3</td> <td style="text-align: center; width: 10px;">4</td> <td style="text-align: center; width: 10px;">5</td> </tr> <tr> <td style="text-align: left;">Muy facil</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/> Muy dificil</td> </tr> </table>	1	2	3	4	5	Muy facil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Muy dificil	<p>12. ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento(al instante)? *</p> <p><i>Mark only one oval.</i></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center; width: 10px;">1</td> <td style="text-align: center; width: 10px;">2</td> <td style="text-align: center; width: 10px;">3</td> <td style="text-align: center; width: 10px;">4</td> <td style="text-align: center; width: 10px;">5</td> </tr> <tr> <td><input type="radio"/> Demasiada disponibilidad</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td>Muy poca disponibilidad</td> </tr> </table>	1	2	3	4	5	<input type="radio"/> Demasiada disponibilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy poca disponibilidad
1	2	3	4	5																			
Muy facil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Muy dificil																		
1	2	3	4	5																			
<input type="radio"/> Demasiada disponibilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy poca disponibilidad																		
<p>9. ¿Cree usted que es mejor evaluar al profesor por cada clase en vez de hacerlo una vez cada semestre, de tal forma que se pueda obtener una retroalimentación más rápida y realista? *</p> <p><i>Mark only one oval.</i></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center; width: 10px;">1</td> <td style="text-align: center; width: 10px;">2</td> <td style="text-align: center; width: 10px;">3</td> <td style="text-align: center; width: 10px;">4</td> <td style="text-align: center; width: 10px;">5</td> </tr> <tr> <td><input type="radio"/> Muy poco</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/> Bastante</td> </tr> </table>	1	2	3	4	5	<input type="radio"/> Muy poco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Bastante	<p>13. ¿Considera que la información del curso se encuentra descentralizada y desorganizada? Por ejemplo: el uso del papel, archivos excel, sistemas web, redes sociales.*</p> <p><i>Mark only one oval.</i></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center; width: 10px;">1</td> <td style="text-align: center; width: 10px;">2</td> <td style="text-align: center; width: 10px;">3</td> <td style="text-align: center; width: 10px;">4</td> <td style="text-align: center; width: 10px;">5</td> </tr> <tr> <td><input type="radio"/> Muy poco</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/> Demasiado</td> </tr> </table>	1	2	3	4	5	<input type="radio"/> Muy poco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Demasiado		
1	2	3	4	5																			
<input type="radio"/> Muy poco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Bastante																			
1	2	3	4	5																			
<input type="radio"/> Muy poco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Demasiado																			
<p>10. ¿Demora/Demoraba mucho resolver las dudas sobre el curso? Por ejm: se espera a que haya (una clase o a un delegado o disponibilidad del profesor) para hacer preguntas o en cualquier momento y/o lugar se puede realizar. *</p> <p><i>Mark only one oval.</i></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center; width: 10px;">1</td> <td style="text-align: center; width: 10px;">2</td> <td style="text-align: center; width: 10px;">3</td> <td style="text-align: center; width: 10px;">4</td> <td style="text-align: center; width: 10px;">5</td> </tr> <tr> <td><input type="radio"/> No demora</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/> Demora mucho</td> </tr> </table>		1	2	3	4	5	<input type="radio"/> No demora	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Demora mucho												
1	2	3	4	5																			
<input type="radio"/> No demora	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Demora mucho																			
<p>11. ¿Cuanto demora/demoraba en enterarse de los anuncios del profesor tales como postergación de clases, reprogramación o cualquier otro relacionado al curso. *</p> <p><i>Mark only one oval.</i></p> <table border="0" style="width: 100%;"> <tr> <td style="text-align: center; width: 10px;">1</td> <td style="text-align: center; width: 10px;">2</td> <td style="text-align: center; width: 10px;">3</td> <td style="text-align: center; width: 10px;">4</td> <td style="text-align: center; width: 10px;">5</td> </tr> <tr> <td><input type="radio"/> Muy poco</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/> Demasiado</td> </tr> </table>		1	2	3	4	5	<input type="radio"/> Muy poco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Demasiado												
1	2	3	4	5																			
<input type="radio"/> Muy poco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Demasiado																			
<p>This content is neither created nor endorsed by Google.</p> <p>Google Forms</p>																							

Figura 133.Segunda parte del formato de la encuesta (Elaboración propia)

La encuesta se realizó antes del inicio de las clases del presente año 2020, es por ello que los alumnos que cursaban el año académico actual aún no tenían experiencia en este año académico.

La encuesta fue inicialmente orientada para todos los alumnos del 1.^{er} al 5.^o año académico y alumnos egresados, pero luego se realizaron filtros de las encuestas realizadas para delimitar mejor la población que se tomará en cuenta:

- Se tomaron en cuenta las encuestas realizadas por los alumnos de obstetricia que cursan actualmente del 3.^{er} al 5.^o año académico y a los egresados menores a un año de antigüedad debido a que cuentan con la experiencia más reciente y conocen a cabalidad las funciones, controles académicos y administrativos de la EPO.
- No se tomaron en cuenta encuestas realizadas por alumnos que cursaron el 1.^{er} año académico dado que ellos son ingresantes del presente año y llevarán en Estudios Generales no en la EPO.
- No se tomaron en cuenta las encuestas realizadas por alumnos que cursan el 2.^o año académico porque ellos en su mayoría llevaron el 1.^{er} año académico en Estudios Generales (EEGG) no en la EPO, dado ese filtro no se tomaron en cuenta una minoría de encuestas realizadas de alumnos que repitieron y siguen cursando el 2.^o año académico.

- Por último, no se tomaron en cuenta a encuestas realizadas por los alumnos egresados con más de un año de antigüedad debido a que su experiencia no es la más reciente.

El tamaño de la población que se consideró fue de 364 alumnos está distribuido de la siguiente forma:

- 93 alumnos de la base 18.
- 94 alumnos de la base 17.
- 114 internos.
- 63 egresados con menos de 1 año.

Para calcular la cantidad de las encuestas realizadas, nos apoyaremos en la pregunta nro. 3: ¿en qué año se encuentra cursando en la universidad o indicar si es egresado? Los resultados se muestran a continuación (véase Tabla 21).

Tabla 21
Distribución de encuestas realizadas por los alumnos

AÑO	CANTIDAD	PORCENTAJE
3 año	67	30.59%
4 año	57	26.03%
5 año(internado)	57	26.03%
Egresado menos de 1 año	38	17.35%
TOTAL	219	100%

(Elaboración propia)

La cantidad de alumnos que llenaron la encuesta fue de 219 de un total de población de 364 alumnos de la Escuela Profesional de Obstetricia de la UNMSM que representan un 60.16%.

Con el fin de validar que la cantidad de alumnos que llenaron la encuesta es una muestra probabilística con una buena representatividad de una población fija se valida con el cálculo del tamaño muestral en donde se usa la fórmula que se detalla a continuación (véase Figura 134).

$$n = \frac{N \times Z_a^2 \times p \times q}{d^2 \times (N - 1) + Z_a^2 \times p \times q}$$

en donde,

N = tamaño de la población

Z = nivel de confianza,

P = probabilidad de éxito, o proporción esperada

Q = probabilidad de fracaso

D = precisión (Error máximo admisible en términos de proporción)

Figura 134.Fórmula del cálculo del tamaño de la muestra de población fija (Torres & Paz, 2006)

Los parámetros que se mencionan (véase Figura 134) son para la presente tesis tienen los siguientes valores:

- Las variables p (probabilidad de éxito) y q (probabilidad de fracaso) por defecto se toman como 0.5 (50% para cada una) de acuerdo a (Torres & Paz, 2006) ya que no se cuenta con estudios anteriores.
- Se usa el 95% de nivel de confianza (variable $z = 1.96$ según (Torres & Paz, 2006)).
- Se usa el 5% de margen de error (variable d).
- Y un tamaño de población fija de 364 estudiantes (variable N).

El cálculo de la fórmula para el cálculo del tamaño muestral de una población fija (véase Figura 134) con los parámetros mencionados anteriormente dan un tamaño requerido de 188 estudiantes (variable n), por lo cual el tamaño muestral que obtuvimos de 219 estudiantes del presente trabajo cumple de sobra con esa cantidad y de esta manera se demuestra que la encuesta realizada tiene una muestra probabilística que tiene buena representatividad de la población.

A continuación, se muestra los resultados detallados de la encuesta por cada pregunta, se obvia la pregunta del código de alumno ya que solo fue referencial y la pregunta (véase Tabla 21) sobre qué año cursan debido a que ya se mencionó.

Pregunta: ¿Qué tan lejos vive de su centro de estudios (campus universitario) y lugares relacionados como hospitales? Donde el indicador 1 es “muy cerca” y 5 es “muy lejos”. A continuación, se muestran los resultados (véase Tabla 22).

Tabla 22

Resultados de la pregunta: ¿Qué tan lejos vive de su centro de estudios (campus universitario) y lugares relacionados como hospitales?

INDICADOR	CANTIDAD	PORCENTAJE
1	2	0.50%
2	19	8.70%
3	97	44.50%
4	79	36.20%
5	22	10.10%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿(Cumplen/Cumplieron) los profesores con todas las clases del syllabus? Donde 1 es “muy cerca” y 5 es “muy lejos”. A continuación, se muestran los resultados (véase Tabla 23)

Tabla 23

Resultados de la pregunta: ¿(Cumplen/Cumplieron) los profesores con todas las clases del syllabus?

INDICADOR	CANTIDAD	PORCENTAJE
1	1	0.46%

2	8	3.65%
3	46	21.00%
4	133	60.73%
5	31	14.16%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las clases avanzadas por el profesor? Donde 1 es “muy pocas” y 5 es “todas sin excepción”. A continuación, se muestran los resultados (véase Tabla 24).

Tabla 24

Resultados de la pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las clases avanzadas por el profesor?

INDICADOR	CANTIDAD	PORCENTAJE
1	2	0.91%
2	51	23.29%
3	126	57.53%
4	38	17.35%
5	2	0.91%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Qué tan difícil (es/fue) llevar un control de asistencias que has realizado en tus cursos? Donde 1 es “muy fácil” y 5: “muy difícil”. A continuación, se muestran los resultados (véase Tabla 25).

Tabla 25

Resultados de la pregunta: ¿Qué tan difícil (es/fue) llevar un control de asistencias que has realizado en tus cursos?

INDICADOR	CANTIDAD	PORCENTAJE
1	19	8.68%
2	61	27.85%
3	78	35.62%
4	58	26.48%
5	3	1.37%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Considera que (hay/hubo) retrasos en el tiempo en el que te enterabas de tus notas? Donde 1 es “muy fácil” y 5 es “muy difícil”. A continuación, se muestran los resultados (véase Tabla 26).

Tabla 26

Resultados de la pregunta: ¿Considera que (hay/hubo) retrasos en el tiempo en el que te enterabas de tus notas?

INDICADOR	CANTIDAD	PORCENTAJE
1	4	1.83%
2	26	11.87%
3	66	30.14%
4	94	42.92%
5	29	13.24%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las notas que has obtenido en tus cursos? Donde 1 es “muy poco” y 5 es “demasiado”. A continuación, se muestran los resultados (véase Tabla 27).

Tabla 27

Resultados de la pregunta: ¿Qué tan difícil (es/fue) llevar un control y seguimiento a las notas que has obtenido en tus cursos?

INDICADOR	CANTIDAD	PORCENTAJE
1	4	1.83%
2	28	12.79%
3	90	41.10%
4	78	35.62%
5	19	8.68%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Cree usted que es mejor evaluar al profesor por cada clase en vez de hacerlo una vez cada semestre, de tal forma que se pueda obtener una retroalimentación más rápida y realista? Donde 1 es “muy fácil” y 5 es “muy difícil”. A continuación, se muestran los resultados (véase Tabla 28).

Tabla 28

Resultados de la pregunta: ¿Cree usted que es mejor evaluar al profesor por cada clase en vez de hacerlo una vez cada semestre, de tal forma que se pueda obtener una retroalimentación más rápida y realista?

INDICADOR	CANTIDAD	PORCENTAJE
1	2	0.91%
2	5	2.28%
3	27	12.33%
4	96	43.84%
5	89	40.64%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Demora/Demoraba mucho resolver las dudas sobre el curso? Por ejemplo: se espera a que haya (una clase o a un delegado o disponibilidad del profesor) para hacer preguntas o en cualquier momento y/o lugar se puede realizar. Donde 1 es “muy poco” y 5 es “bastante”. A continuación, se muestran los resultados (véase Tabla 29).

Tabla 29

Resultados de la pregunta: ¿Demora/Demoraba mucho resolver las dudas sobre el curso?

INDICADOR	CANTIDAD	PORCENTAJE
1	17	7.76%
2	57	26.03%
3	90	41.10%
4	49	22.37%
5	6	2.74%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Cuánto demora/demoraba en enterarse de los anuncios del profesor tales como postergación de clases, reprogramación o cualquier otro relacionado al curso? Donde 1 es “no demora” y 5 es “demora mucho”. A continuación, se muestran los resultados (véase Tabla 30).

Tabla 30

Resultados de la pregunta: ¿Cuánto demora/demoraba en enterarse de los anuncios del profesor tales como postergación de clases, reprogramación o cualquier otro relacionado al curso?

INDICADOR	CANTIDAD	PORCENTAJE
1	20	9.13%
2	55	25.11%
3	65	29.68%
4	60	27.40%
5	19	8.68%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento (al instante)? Donde 1 es “muy poco” y 5 es “demasiado”. A continuación, se muestran los resultados (véase Tabla 31).

Tabla 31

Resultados de la pregunta: ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento (al instante)?

INDICADOR	CANTIDAD	PORCENTAJE
1	1	0.46%

2	16	7.31%
3	47	21.46%
4	64	29.22%
5	91	41.55%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Considera que la información del curso se encuentra descentralizada y desorganizada? Por ejemplo: el uso del papel, archivos excel, sistemas web, redes sociales. Donde 1 es “demasiada disponibilidad” y 5 es “muy poca disponibilidad”. A continuación, se muestran los resultados (véase Tabla 32).

Tabla 32

Resultados de la pregunta ¿Considera que la información del curso se encuentra descentralizada y desorganizada?

INDICADOR	CANTIDAD	PORCENTAJE
1	5	2.28%
2	25	11.42%
3	92	42.01%
4	70	31.96%
5	27	12.33%
TOTAL	219	100%

(Elaboración propia)

Pregunta: ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento (al instante)? Donde 1 es “muy poco” y 5 es “demasiado”. A continuación, se muestran los resultados (véase Tabla 33).

Tabla 33

Resultados de la pregunta ¿Se (encuentra/encontraba) la información del curso como asistencias o notas digitalizada, disponible para poder consultar en cualquier momento (al instante)?

INDICADOR	CANTIDAD	PORCENTAJE
SI	214	97.72%
NO	5	2.28%
TOTAL	219	100%

(Elaboración propia)

Anexo 2: Constancia de la validación del producto

Se obtuvo la aprobación de la dirección de la Escuela Profesional de Obstetricia de la Universidad Nacional Mayor de San Marcos, en la cual se aprobó el funcionamiento general del sistema informático y sus características, y afirmó el impacto positivo que tiene para la escuela (véase Figura 135) entre los puntos se mencionan:

- Brinda registro y control de asistencias de clases de los alumnos profesores.
- Brinda registro y control de notas de los alumnos.

- Brinda manejo de fórmulas de cálculo de notas de los grupos de los cursos.
- Brinda medición de desempeño de profesores mediante encuestas por clase.
- Brinda visualización de estadísticas de asistencias, notas y encuestas.
- Proporciona un medio interactivo mediante los anuncios, preguntas y respuestas.
- Permite visualizar las notificaciones en tiempo real.
- La información se muestra disponible, centralizada y en línea la cual es accesible mediante el aplicativo.
- El tiempo de búsqueda de la información es menor a comparación de antes.

Constancia de Validación de Especialista

Nombre: CLARA MARGARITA DÍAZ TINOCO
 Especialidad: OBSTETRICIA: SALUD SEXUAL REPRODUCTIVA
 DNI: 07403289

Por medio de la presente hago constar que he realizado la validación del Sistema de Control Académico elaborado por Omar Cuba Estrella y Maycol Henry Espinoza Ramírez quienes están realizando el trabajo de tesis titulado "APLICACIÓN MÓVIL DE CONTROL ACADÉMICO UTILIZANDO LA ARQUITECTURA DE MICROSERVICIOS BAJO LA METODOLOGÍA DE DESARROLLO SCRUM. CASO: ESCUELA PROFESIONAL DE OBSTETRICIA DE LA UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS".

Esta validación consistió en la evaluación general del funcionamiento del sistema informático y de la importancia para la escuela de contar con este sistema. Las características que tiene el sistema y que fueron validadas son las siguientes:

- Brinda registro y control de asistencias de clases de alumnos y de profesores.
- Brinda registro y control de notas de alumnos.
- Brinda manejo de fórmulas del cálculo de notas de los grupos del curso y de los cursos.
- Brinda medición de desempeño de profesores mediante encuestas por clase.
- Brinda visualización de estadísticas de asistencias, notas y encuestas.
- Proporciona un medio de interacción mediante anuncios, preguntas y respuestas.
- Permite visualizar las notificaciones en tiempo real.
- La información se muestra disponible, centralizada y en línea la cual es accesible mediante el aplicativo.
- El tiempo de búsqueda de información a comparación de antes es menor.

Lima, 13 de agosto de 2019


 CLARA MARGARITA DÍAZ TINOCO
 DIRECTORA
 E.P. DE OBSTETRICIA

Figura 135.Constancia de validación de la especialista (Elaboración propia)