



**UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Prototipo de sistema de control y gestión de
ayudas estudiantiles para la Escuela de Ingeniería
Civil en Computación**

DIEGO ANTONIO MATUS CAMPOS

Profesor Guía: RODRIGO PAVEZ MADARIAGA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Enero, 2021

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



UNIVERSIDAD DE TALCA
DIRECCIÓN
SISTEMA DE BIBLIOTECAS

UNIVERSIDAD DE TALCA
SISTEMA DE BIBLIOTECAS
CAMPUS CURICO

Curicó, 2022

Dedicado a mi familia.

AGRADECIMIENTOS

Agradecer a mi padre y mi madre por todo el apoyo que me han brindado en esta etapa. Agradezco los valores y todo el sacrificio que han realizado para darme la posibilidad de vivir esta etapa. También dar gracias por darme siempre palabras de aliento cuando estaba a punto de rendirme y tirar la toalla, ya que ellos saben todo el sacrificio y esfuerzos que tuve que realizar para poder terminar esta etapa. Gracias a ellos soy quien soy ahora.

Quiero dar gracias también a todos mis amigos de la universidad, no me imaginó haber vivido esta etapa sin su apoyo y su amistad. Gracias por todos esos buenos momentos y anécdotas que pasamos. Sin ustedes nada hubiera sido igual. También agradecer a mis amigos de fuera de la universidad gracias por su apoyo y sus buenas vibras siempre, y darme fuerzas para no decaer, seguir adelante con esta etapa.

Por último agradecer a cada uno de los profesores que formaron parte en mi educación universitaria, en especial a mi profesor guía, por apoyarme en este proyecto.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	IX
Índice de Tablas	XVI
Resumen	XVII
1. Introducción	18
1.1. Presentación del problema	18
1.2. Objetivos	19
1.2.1. Objetivo general	19
1.2.2. Objetivos específicos	19
1.3. Propuesta de solución	20
1.4. Alcances	20
1.5. Resumen del capítulo	21
2. Marco Teórico	22
2.1. Trabajo relacionado	22
2.2. Ayudas estudiantiles	25
2.2.1. Conceptos relacionados	26
2.2.2. Procesos de gestión de ayudas estudiantiles	26
2.3. Desarrollo del software	34
2.3.1. Conceptos relacionados	35
2.3.2. Tecnologías aplicación web	36
2.3.3. Tecnologías aplicación móvil	37
2.3.4. Almacenamiento de datos y lado del servidor	39
2.4. Metodología de desarrollo	40

2.5.	Metodología de evaluación	42
2.6.	Resumen del capítulo	43
3.	Metodología de desarrollo	44
3.1.	Metodología	44
3.2.	Metodología de desarrollo ágil Scrumban	44
3.2.1.	Product Backlog	45
3.2.2.	Priorización y estimación de esfuerzo de historias de usuario	48
3.2.3.	Planificación	50
3.2.4.	Tablero Kanban	52
3.2.5.	Pruebas Unitarias de Caja Negra	52
3.3.	Resumen del capítulo	53
4.	Desarrollo de la aplicación web	54
4.1.	Flujo de trabajo de la aplicación web	54
4.2.	Diseño de la aplicación web	55
4.2.1.	Arquitectura física	56
4.2.2.	Arquitectura lógica	56
4.2.3.	Diagrama de clases	59
4.2.4.	Base de datos	61
4.2.5.	Implementación de Firebase Cloud Firestore	61
4.2.6.	Implementación de Firebase Authentication	62
4.2.7.	Implementación del servicio Firebase Cloud Messaging	63
4.3.	Ejecución de las iteraciones	64
4.3.1.	Iteración 1 - Carga masiva de datos de estudiantes, cursos y docentes en aplicación web	64
4.3.2.	Iteración 2 - Autenticación de usuarios y visualización de ayudantías por cursos en la aplicación web	67
4.3.3.	Iteración 5 - Asignar estudiantes a docentes tutores	68
4.3.4.	Iteración 6 - Cálculo de horas de ayudantías y resumen de ayudantes del semestre	70
4.3.5.	Iteración 8 - Mostrar información de ayudantías, tutorías y generación y descarga de información del proceso en formato XLS	73

4.4. Pruebas	73
5. Desarrollo aplicación móvil	76
5.1. Flujo de trabajo de la aplicación móvil	76
5.1.1. Flujo para el proceso de ayudantías	76
5.1.2. Flujo para el proceso de tutorías docentes en la aplicación . .	77
5.1.3. Flujo para el proceso de tutorías estudiantiles en la aplicación	78
5.2. Diseño de la aplicación móvil	78
5.2.1. Arquitectura Física	78
5.2.2. Arquitectura lógica	79
5.2.3. Diagrama de clases	81
5.2.4. Implementación Cloud Firestore	87
5.2.5. Implementación de Firebase Authentication	87
5.2.6. Implementación de Firebase Cloud Messaging	87
5.3. Ejecución de iteraciones	88
5.3.1. Iteración 1 - Autenticación de usuarios en aplicación móvil . .	88
5.3.2. Iteración 2 - Activación de cuentas de usuarios estudiantes . .	88
5.3.3. Iteración 3 - Registrar reuniones de ayudantías en la aplicación móvil	90
5.3.4. Iteración 4 - Registro de planificación, asistencia de ayudantías y evaluaciones de ayudantes por parte del estudiante en apli- cación móvil	91
5.3.5. Iteración 5 - Recepcionar, agendar reuniones y registrar ano- taciones de tutorías docentes	98
5.3.6. Iteración 6 - Postular a una ayudantía y seleccionar un ayu- dante	100
5.3.7. Iteración 7 - Registrar planificación, asistencia de tutorías es- tudiantiles y evaluación de tutores estudiantiles	102
5.4. Pruebas	104
5.5. Resumen del capítulo	109
6. Resultados	110
6.1. Prueba de usabilidad de la aplicación móvil	110
6.2. Prueba de usabilidad plataforma web	111

6.3. Verificación del cumplimiento de los objetivos	112
6.3.1. Comprobación de objetivo específico N° 1	112
6.3.2. Comprobación de objetivo específico N° 2	112
6.3.3. Comprobación de objetivo específico N° 3	117
7. Conclusiones y trabajo futuro	120
7.1. Conclusiones	120
7.2. Trabajo Futuro	121
Glosario	122
Anexos	
A: Flujo de trabajo de la aplicación web	126
A.1. Carga de datos masiva de cursos, estudiantes y docentes.	126
A.2. Cálculo de horas propuestas de ayudantías	127
A.3. Asignación de estudiantes a docente tutor	128
A.4. Asignación de estudiantes a tutor estudiantil	129
B: Flujo de trabajo de la aplicación móvil	130
B.1. Activación de cuenta de estudiante	131
B.2. Postulación a una oferta de ayudantía	132
B.3. Selección de un ayudante	133
B.4. Evaluar un ayudante	134
B.5. Crear una reunión de ayudantías	135
B.6. Registrar asistencia de ayudantías	136
B.7. Agendar reuniones con estudiantes tutor docente	137
B.8. Registrar anotaciones de sesiones	138
B.9. Reportar el avance que tuvo el estudiante durante el semestre	139
B.10. Crear planificación para tutorías estudiantiles	140
C: Diagrama de clases de la aplicación móvil	141
D: Navegación de la aplicación web	150
D.1. Navegación de apartado de ayudantías	150
D.2. Navegación de tutorías docentes	152

D.3. Navegación de tutorías estudiantiles	154
E: Navegación de la aplicación móvil	156
E.1. Navegación del perfil de ayudante	156
E.2. Navegación perfil estudiante	158
E.3. Navegación perfil de docente	161
E.4. Navegación del perfil de tutor docente	163
E.5. Navegación tutorías estudiantiles	165
E.6. Navegación para postulaciones a ayudantías	167
F: Evidencia de pruebas de aplicación web	168
F.1. Prueba de carga masiva de datos	169
F.1.1. Entradas de la prueba	169
F.1.2. Salidas de la prueba	170
F.2. Prueba de asignación de estudiantes a docentes tutores	171
F.2.1. Entradas de la prueba	171
F.2.2. Salidas de la prueba	172
F.3. Prueba de asignación de estudiantes a tutores estudiantiles	173
F.3.1. Entrada de la prueba	173
F.3.2. Salidas de la prueba	174
F.4. Prueba de cálculo de horas propuestas de horas de ayudantías	175
F.4.1. Entrada de la prueba	175
F.4.2. Salidas de la prueba	175
G: Evidencia de pruebas de la aplicación móvil	176
G.1. Prueba de registro de reuniones de ayudantías	176
G.1.1. Entradas de la prueba	176
G.1.2. Salidas de la prueba	177
G.2. Prueba de registro de planificaciones de ayudantías	178
G.2.1. Entradas de la prueba	178
G.2.2. Salidas de las pruebas	179
G.3. Prueba de registro de asistencia de ayudantías ayudante	180
G.3.1. Entradas de la pruebas	180
G.3.2. Salidas de la pruebas	181
G.4. Prueba de registro de asistencia de ayudantías estudiante	182

G.4.1. Entradas de la prueba	182
G.4.2. Salidas de la prueba	183
G.5. Prueba de postulación de ayudante	185
G.5.1. Entrada de la prueba	185
G.5.2. Salidas de la prueba	186
G.6. Prueba de selección de ayudante	187
G.6.1. Entradas de la prueba	187
G.6.2. Salidas de la prueba	187
G.7. Prueba de evaluación de ayudantes	189
G.7.1. Entradas de la prueba	189
G.7.2. Salidas de la prueba de evaluación de ayudante	191
G.8. Prueba de evaluación de ayudantes por parte del docente	192
G.8.1. Salidas de la prueba	195
G.9. Prueba de evaluación de tutores docentes	196
G.9.1. Entradas de la prueba	196
G.9.2. Salidas de la prueba	198
H: Resumen de cuestionarios aplicados	199
H.1. Resumen de resultados del cuestionario de usabilidad de la aplicación móvil	199
H.2. Resumen de los resultados del cuestionario de usabilidad para la apli- cación móvil.	200
H.3. Escala de puntajes utilizado para medir el resultado de usabilidad . .	201

ÍNDICE DE FIGURAS

	página
2.1. Diagrama de modelado de negocio del proceso de gestión de ayudantías por parte de la directora	26
2.2. Diagrama de modelado de negocio del proceso de gestión de ayudantías por parte del docente	28
2.3. Diagrama de modelado de negocio del proceso de gestión de tutorías docentes para el rol de directora de escuela	29
2.4. Diagrama de modelado de negocio del proceso de gestión de tutorías docentes llevado por el docente	31
2.5. Diagrama de modelado de negocio del proceso de gestión de tutorías docentes llevado por el estudiante	32
2.6. Diagrama de modelado de negocio del proceso de gestión de tutorías estudiantiles llevado por el tutor	33
2.7. Diagrama de modelado de negocio del proceso de gestión de tutorías estudiantiles llevado por el estudiante	34
2.8. Proceso para seguir la metodología Scrumban	41
3.1. Tablero Scrumban	52
4.1. Arquitectura física - Cliente Servidor de la aplicación web	56
4.2. Arquitectura lógica - MVC del sistema a alto nivel	57
4.3. Arquitectura lógica - MVC aplicación web	58
4.4. Diagrama de clases del sistema	60
4.5. Modelo de datos en la base de datos Cloud Firestore	61
4.6. Instalación de librerías de los servicios de Firebase en Angular	62
4.7. Consola de Firebase - Servicio Authentication	62
4.8. URL para envío de notificaciones push a través de la API de Firabase Cloud Messaging	63
4.9. Estructura JSON para envío de notificaciones push mediante el servicio FCM	64
4.10. Input de tipo file para selección del archivo Excel de carga de datos masiva	65

4.11. Selección del archivo Excel de carga de datos masiva	65
4.12. Almacenamiento de datos en Cloud Firestore de Estudiantes	66
4.13. Visualización de estudiantes cargados en la aplicación web	67
4.14. Vista de asignación de estudiantes a docentes tutores	68
4.15. Almacenamiento de tutor docente en la base de datos	69
4.16. Notificación enviada al tutor docente	70
4.17. Implementación de calcular propuesta de horas de ayudantías	71
4.18. Diálogo para calcular la propuesta de horas de ayudantías	71
4.19. Notificación de propuesta de horas de ayudantías de un curso	72
4.20. Resumen de ayudantes del semestre	72
5.1. Arquitectura física - Cliente-Servidor aplicación móvil	79
5.2. Arquitectura lógica - MVC aplicación móvil	80
5.3. Parte del diagrama de clases del rol de estudiante para ingresar a la aplicación y realizar una postulación a ayudantías	82
5.4. Parte del diagrama de clases del rol de estudiante para visualizar las reuniones de ayudantías y tutorías	83
5.5. Parte del diagrama de clases del rol de estudiante para las evaluaciones de ayudantes y tutor docente	84
5.6. Parte del diagrama de clases del rol de estudiante para el registro de asistencia para ayudantías y tutorías estudiantiles	85
5.7. Implementación del servicio Cloud Firestore en Android Studio	87
5.8. Implementación del servicio Authentication en Android Studio	87
5.9. Implementación del servicio Firebase Cloud Messaging en Android	87
5.10. Proceso de activación de cuentas de estudiantes	89
5.11. Enlace de verificación de correo electrónico	89
5.12. Vistas implementadas para registrar una reunión de ayudantía	90
5.13. Vistas de planificación de ayudantías	92
5.14. Almacenamiento de una planificación de ayudantías en la base de datos	93
5.15. Vistas implementadas para tomar asistencia por parte del ayudante	95
5.16. Vistas de registro de asistencia de ayudantías por parte del estudiante	96
5.17. Vistas implementadas para la evaluación de ayudantes	97
5.18. Almacenamiento de tutor docente en la base de datos	98
5.19. Vistas implementadas en la agenda de reuniones de tutores docentes	99

5.20. Fragmentos implementados para postular a una ayudantía	100
5.21. Fragmentos implementados para la selección de ayudantes	101
5.22. Fragmentos implementados para la funcionalidad de planificación de tutorías estudiantiles	102
5.23. Fragmentos implementados para la funcionalidad de tomar asistencia de sesiones de tutorías estudiantiles	103
5.24. Fragmentos implementados para la funcionalidad de marcar asistencia de sesiones de tutorías estudiantiles para el estudiante	104
6.1. Tiempo de cálculo de horas propuestas de horas de ayudantía antes de usar la aplicación web	112
6.2. Tiempo de cálculo de horas de propuestas de horas al usar la aplicación web	113
6.3. Tiempo de postulación a ayudantías antes de usar la aplicación móvil	113
6.4. Tiempo de postulación a ayudantías al usar la aplicación móvil	114
6.5. Tiempo de selección de ayudantes después de usar la aplicación móvil	114
6.6. Tiempo de selección de un ayudante después de usar la aplicación móvil	115
6.7. Tiempo de obtención de resumen de ayudantes del semestre antes de usar la aplicación web	115
6.8. Tiempo de obtención de resumen de ayudantes del semestre al usar la aplicación web	116
6.9. Tiempo de obtención de informe de pagos de ayudantes del semestre antes de usar la aplicación web	116
6.10. Tiempo de obtención de resumen de ayudantes del semestre al usar la aplicación web	117
6.11. Apartado de agenda de reuniones de tutor docente a estudiante	118
6.12. Apartado de generación de anotaciones de sesiones	118
6.13. Interacción entre roles	119
A.1. Flujo para realizar la carga de datos masiva de cursos, docentes y estudiantes	126
A.2. Flujo para realizar el calculo de horas propuestas de ayudantías	127
A.3. Flujo que se realiza para la asignación de estudiantes a docentes	128
A.4. Flujo que se realiza para la asignación de estudiantes a tutores estu- diantiles	129

B.1. Flujo que se realiza para activar una cuenta de estudiante en la aplicación móvil	131
B.2. Flujo que se realiza para postular a una oferta de ayudantía	132
B.3. Flujo que se realiza para seleccionar a un ayudante	133
B.4. Flujo de realizado para evaluar un ayudante	134
B.5. Flujo que se realiza para crear reunión de ayudantía	135
B.6. Flujo que se realiza para registro de asistencia reunión de ayudantía .	136
B.7. Flujo que se realiza para agendar reuniones de tutorías docentes . . .	137
B.8. Flujo para registrar una anotación de una sección	138
B.9. Flujo que se realiza para reportar el avance que obtuvo el estudiante durante el semestre	139
B.10. Flujo que se realiza para realizar una planificación para un tutoría estudiantil	140
C.1. Parte del diagrama de clases del rol de ayudante para el agenda de reuniones y planificación de ayudantías	142
C.2. Parte del diagrama de clases del rol de ayudante para el registro de asistencia de ayudantías	143
C.3. Parte del diagrama de clases del rol de docente visualización de cursos y difusión de ayudantías	144
C.4. Parte del diagrama de clases del rol de docente visualización de postulaciones y selección de ayudantes	145
C.5. Parte del diagrama de clases del rol de docente visualización de ayudantes y evaluación	146
C.6. Parte del diagrama de clases del rol de tutor docente visualización de asignaciones y reuniones con estudiantes	147
C.7. Parte del diagrama de clases del rol de tutor docente sesiones y anotaciones con estudiantes	148
C.8. Parte del diagrama de clases del rol de tutor estudiantil planificaciones de tutorías estudiantiles	149
D.1. Navegación entre ventanas en el aparatado de ayudantías	150
D.2. Vista de ayudantías por curso	151
D.3. Vista de resumen de ayudantías	151
D.4. Navegación entre ventanas en el aparatado de tutorías docentes . . .	152

D.5. Vista de tutorías por docente	152
D.6. Vista de estudiantes de tutorías docentes	153
D.7. Vista de resumen de tutorías docentes	153
D.8. Navegación entre ventanas en el aparatado de tutorías estudiantiles .	154
D.9. Vista de asignación de estudiantes de tutores estudiantiles	154
D.10.Vista de estudiantes en el aparado de tutorías estudiantiles	155
E.1. Navegación del menú de ayudantes	156
E.2. Vistas de agregar reunión y tomar asistencia de una ayudantía	157
E.3. Vistas para agregar una planificación de ayudantías	157
E.4. Navegación del perfil de estudiante	158
E.5. Vistas para visualizar reuniones y solicitudes en el perfil de estudiante	158
E.7. Vistas para ingresar el código para marcar asistencia en un ayudantía y tutoría estudiantil	159
E.6. Vistas para marcar asistencia para ayudantías y tutorías estudiantiles	159
E.8. Vistas para evaluar un ayudante	160
E.9. Vistas para evaluar un tutor docente	160
E.10.Navegación entre vistas perfil del docente	161
E.11.Vistas para agregar una oferta de ayudantía	161
E.12.Vistas para visualizar las postulaciones de ayudantías	162
E.13.Vistas para evaluar un ayudante	162
E.14.Navegación del perfil tutor docente	163
E.15.Vistas para visualizar asignaciones y reportar avance del estudiante .	163
E.16.Vistas para agregar una reunión de tutorías docentes	164
E.17.Vistas para visualizar sesiones y anotaciones	164
E.18.Navegación del perfil de tutor estudiantil	165
E.19.Vistas para agregar una planificación e tutorías estudiantiles	165
E.20.Vistas para tomar asistencia de tutorías estudiantiles	166
E.21.Vistas para postular a una ayudantía	167
F.1. Entradas para la prueba de carga masiva de datos - Archivo XLS . .	169
F.2. Salidas de la prueba de carga masiva de datos - registro de colección estudiantes en base de datos	170
F.3. Salidas de la prueba de carga masiva de datos - registro de colección docentes en base de datos	170

F.4. Salidas de la prueba de carga masiva de datos - registro de colección cursos en base de datos	171
F.5. Entradas de las prueba de asignación - estudiantes y docente	171
F.6. Salidas de la prueba de asignación de estudiantes a tutores docentes - registro en base de datos	172
F.7. Entradas de las prueba de asignación - estudiantes y tutor	173
F.8. Salida de las prueba de asignación de estudiantes a tutor estudiantil - registro de tutor estudiantil	174
F.9. Salidas de la prueba de asignación de estudiantes a tutores docentes - notificación a tutor	174
F.10. Entradas para la prueba de cálculo de horas - seleccionar un curso . .	175
F.11. Entradas para la prueba de cálculo de horas - ingreso de horas de ayudantías y/o laboratorio	175
F.12. Salidas para la prueba de cálculo de horas - curso con la horas calculadas	175
G.1. Entradas para agregar una reunión de ayudantías	176
G.2. Salidas para la prueba de agregar una reunión de ayudantía	177
G.3. Muestra de a reunión agregada	177
G.4. Entradas de registro de planificación - Primer paso	178
G.5. Entradas de registro de planificación - Segundo paso	178
G.6. Salidas para prueba de registro de planificación	179
G.7. Entradas para registrar asistencia - Seleccionar una reunión	180
G.8. Entradas para registrar asistencia - Generar código	181
G.9. Salida de la prueba de registro de asistencia - Actualización del código	181
G.10. Entradas de la prueba de registro de asistencia estudiante	182
G.11. Salidas de la prueba de registro de asistencia estudiantes	183
G.12. Salidas de la prueba de registro de asistencia estudiante - Actualiza- ción en base de datos	184
G.13. Entradas de la prueba de selección de ayudante - Selección de postulación	185
G.14. Entradas de la prueba de selección de ayudante - Ingresar datos de postulación	186
G.15. Salidas de la prueba de selección de ayudante - Notificación a docente	186
G.16. Entradas para la prueba de selección - Selección de postulaciones . .	187
G.17. Salidas para la prueba de selección - Notificación de aceptación	187

G.18.Salidas para prueba de selección de ayudante - Registro de ayudante en base de datos	188
G.19.Entradas para la prueba de evaluación de ayudantes - Selección de de ayudante a evaluar	189
G.20.Entradas para la prueba de evaluación de ayudantes - Respuestas del cuestionario de evaluación	190
G.21.Entradas para la prueba de evaluación de ayudantes - Comentario de al ayudante	190
G.22.Salida de la prueba de evaluación de ayudantes - Registro en base de datos	191
G.23.Entradas para la prueba de evaluación de ayudantes - Selección de ayudante a evaluar	192
G.24.Entradas para la prueba de evaluación de ayudantes - Respuesta del cuestionario de evaluación	193
G.25.Entradas para la prueba de evaluación de ayudantes - Comentario a ayudante	194
G.26.Salida de la prueba de evaluación a ayudante - Registro en base de datos	195
G.27.Entradas para la prueba de evaluación de tutores docentes - Selección de tutor docente	196
G.28.Entradas para la prueba de evaluación de tutores docentes - Respuestas de cuestionario	197
G.29.Entradas para la prueba de evaluación de tutores docentes - Comentario a tutor docente	197
G.30.Salidas para prueba de evaluación de tutores docentes - Registro en base de datos	198
H.1. Escala de puntaje utilizada para analizar el resultado de las pruebas de usabilidad	201

ÍNDICE DE TABLAS

	página
2.1. Tabla comparativa de software similar al propuesto	24
3.1. Historias de usuario ayudantías	45
3.2. Historias de usuario tutorías docente	47
3.3. Historias de usuario tutorías estudiantiles	48
3.4. Historias de usuario generales	48
3.5. Priorización y estimación de historias de usuario	49
3.6. Planificación del proyecto por iteraciones.	50
3.7. Formato de pruebas de documentación de pruebas unitarias	53
4.1. Prueba de carga masiva de datos de estudiantes, cursos y docentes.	73
4.2. Prueba de asignación de estudiantes a un docente tutor	74
4.3. Prueba de asignación de estudiantes a un docente estudiantiles	74
4.4. Prueba de cálculo de horas de ayudantías.	75
4.5. Prueba de generación de información de resumen del proceso ayu- dantías y tutorías.	75
5.1. Prueba de registro de reuniones de ayudantías.	105
5.2. Prueba de registro de reuniones de ayudantías.	105
5.3. Prueba de registro de asistencia de ayudantías ayudante.	106
5.4. Prueba de registro de asistencia de ayudantías estudiante.	106
5.5. Prueba de selección de ayudante.	107
5.6. Prueba de postulación a oferta de ayudantía.	107
5.7. Prueba de evaluación de ayudantes.	108
5.8. Prueba de evaluación de ayudantes por parte del docente.	108
5.9. Prueba de evaluación de tutores docentes.	109
H.1. Tabla resumen del cuestionario de usabilidad de la aplicación móvil	199
H.2. Tabla resumen del cuestionario de usabilidad de la aplicación web	200

RESUMEN

El presente documento detalla el proceso de creación de una solución tecnológica que ayuda a gestionar y controlar un proceso interno de la Escuela Civil en Computación como lo son las ayudantías y tutorías docentes, ya que actualmente este se realiza de forma manual utilizando software de ofimática, provocando un gran pérdida de tiempo debido a lo dispersa que se almacena la información en diferentes archivos.

Para solucionar esta problemática, se pretende construir un prototipo de sistema de gestión y control de ayudas estudiantiles (SGCAE), el cual consta del desarrollo de una aplicación web y una aplicación móvil. Donde la aplicación móvil le permitirá al estudiante postular a una oferta de ayudantía, crear y registrar asistencia de ayudantías. A su vez podrán evaluar a sus ayudantes. Luego los docentes podrán seleccionar sus ayudantes y evaluarlos. Por último el tutor docente podrá ver sus asignaciones, agendar reuniones y generar anotaciones de las sesiones que tiene con sus estudiantes asignados.

Por otro lado tenemos la aplicación web la cual le permitirá a la asistente calcular las horas propuestas de ayudantías de cada curso, ver el resumen de los ayudantes seleccionados en el semestre, generar información de los montos a pagar a cada ayudante. Por último también se permitirá la asignación de estudiantes a un docente, es decir, tutor docente. A su vez también se mostrará las evaluaciones que le han hecho los estudiantes a un docente

El documento presenta lo esencial para el entendimiento del proyecto, mostrando la teoría, herramientas y tecnologías que son necesarias para implementar la solución. El desarrollo de estas dos aplicaciones se hace utilizando la metodología ágil Scrumban, realizando 8 iteraciones de duración variable de entre 1 a 3 semanas dependiendo de la complejidad de las tareas de cada iteración.

Para evaluar el proyecto se usa un proceso de experimentación en un ambiente controlado para verificar el cumplimiento de los objetivos, mostrando los resultados de este experimento. Por último se realiza una conclusión general del proyecto y se exponen los trabajos futuros a realizar.

1. Introducción

En la Facultad de Ingeniería, específicamente en la escuela de Ingeniería Civil en Computación, cada semestre se lleva a cabo un proceso interno para gestionar las ayudas estudiantiles que se realizan. Estas se dividen en ayudantías realizadas por estudiantes, tutorías dictadas por docentes y tutorías ejecutadas por estudiantes.

Una parte del proceso es la gestión de ayudantías, la que incluye una propuesta de horas de ayudantías para cada curso, que es notificada a cada docente. Siguiendo con la obtención de la información de difusión de las ayudantías, la cual es solicitada a los docentes a cargo de cada curso. Posteriormente se hace la difusión y reclutamiento de ayudantes mediante el uso del correo institucional. Para finalizar el proceso el selecciona un ayudante, reportando la información del estudiante seleccionado, con el fin de generar la resolución de la gestión de ayudantías, donde se realiza la justificación de los montos pagados a los ayudantes durante el semestre.

En cambio la otra parte, lleva a cabo el proceso de tutorías donde se realiza el análisis de los estudiantes en riesgo, para ser derivados según sea el caso a tutorías docentes o tutorías estudiantiles. Una vez han sido derivados a sus tutores, estos últimos deben contactarse con los estudiantes que se le han asignado mediante el correo institucional, con el fin de tener sesiones periódicas para evaluar el nivel de avance que van teniendo los estudiantes. Para finalmente, reportar al finalizar el semestre a la directora de escuela si se han podido contactar con el estudiante y algún comentario, en el caso de los tutores docentes.

1.1. Presentación del problema

Tanto el proceso de gestión de ayudantías como en el proceso para realizar tutorías, están siendo gestionadas solamente por la directora de la escuela y su asistente.

Estos procesos son realizados de forma manual utilizando plantillas Excel, generando muchas veces que la información este dispersa en varios archivos lo que provoca un aumento el tiempo en que se demora en obtener la información. Por otra parte, utilizan como vía de comunicación el correo institucional, lo que provoca muchas veces pérdida de información quedando en el limbo información importante que debió haber llegado a sus destinatarios, o bien muchas veces es enviado a correo no deseado. Adicionalmente se ha detectado que estos procesos no ha sido formalizados provocando la dificultad mantener el control del correcto funcionamiento de las ayudas estudiantiles.

1.2. Objetivos

Los objetivos que se pretenden lograr con la implementación del proyecto en el área de coordinación de ayudas estudiantiles en la escuela de Ingeniería Civil en Computación.

1.2.1. Objetivo general

Establecer un mecanismo de control y gestión interno para el proceso de ayudantías y tutorías docentes realizadas por la escuela de Ingeniería Civil en Computación, mediante el desarrollo de una aplicación móvil y aplicación web de administración.

1.2.2. Objetivos específicos

- Construir una aplicación móvil y web que permita gestionar el proceso de ayudantías y tutorías.
- Disminuir el tiempo invertido en la gestión y obtención de información del proceso de ayudantías.
- Proveer un entorno de monitoreo y control para el docente tutor de los estudiantes que tiene asignados.

1.3. Propuesta de solución

Para dar solución al problema planteado se pretende definir un proceso de control, desarrollando un sistema que integre variados apartados para la gestión de ayudantías y tutorías. El cual constará de dos aplicaciones; una web administrativa y una móvil, las cuales serán detalladas a continuación.

- Por una parte, la aplicación web administrativa podrá ser accedida solamente por la directora de escuela y su asistente. En donde ellas pueden visualizar, generar resúmenes de la información generada de las ayudas estudiantiles, siendo principalmente datos del proceso de ayudantías como la asistencia y evaluaciones.
- Por otro lado la aplicación móvil, podrá ser utilizada por estudiantes y docentes. Donde los estudiantes serán capaces de postular como ayudantes, registrar asistencia de ayudantías y tutorías, además evaluar a su tutor docente y ayudantes. En la misma aplicación los docentes podrán ver y seleccionar a sus ayudantes y evaluarlos, adicionalmente si son asignados como tutores podrán gestionar sus reuniones con los estudiantes y sus sesiones. Conjuntamente, la aplicación notificará a cada usuario al momento de gestionar una postulación, selección o reunión, para otorgar mayor interacción entre los usuarios del sistema.

1.4. Alcances

Los límites establecidos para el proyecto contempla el ambiente en que se desarrollará las funcionalidades que se realizarán y las que no se implementará, las cuales se especifican a continuación.

- El sistema será desarrollado solamente para la escuela de Ingeniería Civil en Computación.
- El sistema constará del desarrollo de una aplicación web administrativa y una aplicación móvil.
- La aplicación móvil sólo será desarrollada nativamente para el sistema operativo Android.

- La aplicación web solamente emitirá un resumen sobre asistencia y evaluaciones de las ayudantías y tutorías estudiantiles.
- La asignación de estudiantes en riesgo las podrá emitir únicamente la directora de escuela a través de la plataforma web administrativa.
- La descarga de los resúmenes generados por la aplicación web será solamente en formato de archivo XLS.
- La plataforma web no generará la resolución final del apartado de ayudantías.
- La aplicación web no contempla la implementación de un apartado de súper administrador para agregar nuevos usuarios a la aplicación web.
- La carga masiva de datos será realizada siguiendo el formato de la nómina de estudiantes inscritos del año 2019.

1.5. Resumen del capítulo

En este capítulo se realizó un primer acercamiento al proyecto de titulación desarrollado. Partiendo en detallar la situación actual del proceso de gestión y control de la escuela de Ingeniería Civil en Computación con respecto a las ayudas estudiantiles disponibles. Posteriormente, se habló sobre el problema detectado en este proceso, el cual apunta el tiempo y esfuerzo que se requiere actualmente por la ausencia de un mecanismo que controle y gestione las ayudas estudiantiles. Por último se expusieron los objetivos y solución propuesta para abordar el problema encontrado, que consiste en desarrollar una aplicación móvil y una plataforma web administrativa que provea de un mecanismo de control.

A continuación en el capítulo dos se planteará el marco de referencia, exponiendo los trabajos relacionados o similares que se han desarrollado para solucionar la gestión de reuniones. Posteriormente, se hablará sobre los conceptos fundamentales para entender el proyecto, también de las herramientas, tecnologías y metodologías que serán usadas para lograr los objetivos.

2. Marco Teórico

En este capítulo se abordan los trabajos similares, relacionando diferentes aspectos en el software desarrollados para la gestión y control de reuniones que ya existen en el mercado. Posteriormente se dará conocimiento de los conceptos básicos e información utilizada para desarrollar la solución propuesta, comenzando por explicar los conceptos relacionados con el proceso de gestión utilizado por la escuela de Ingeniería Civil en Computación. Después se hablará sobre las herramientas y tecnologías para el desarrollo web y móvil elegidas, para terminar hablando sobre la justificación de su elección. Por último se describirán las metodologías de desarrollo de software y evaluación que serán implementadas.

2.1. Trabajo relacionado

Actualmente el desarrollo de aplicaciones, ya sea web o móvil a crecido considerablemente en todas las áreas, resolviendo una amplia variedad de problemas de diferente índole. Por esta razón es importante revisar las opciones disponibles en el mercado del desarrollo de software, con el fin de implementar soluciones novedosas o que complementan a las que ya existen. Es por eso que a continuación se presentarán los trabajos relacionados con la propuesta de solución, la que está orientada principalmente a gestionar y controlar ayudas estudiantiles.

Si pensamos bien las ayudas estudiantiles pueden ser consideradas como reuniones de un grupo de personas con fines académicos. Considerado este aspecto existen software que permiten controlar y gestionar la planificación de reuniones como:

- Calendly: software desarrollado con el fin de programar reuniones sin tener que mandar miles de correos. Este nos permite integrarlo con diferentes calendarios,

controlar horarios de reuniones, programar recordatorios los cuales nos sean notificados y por último generar métricas [1].

- Doodle: plataforma desarrollada con el fin de agilizar la organización de reuniones, permitiendo proponer horarios, invitar a los participantes y que ellos seleccionen el horario que más le acomode [2].
- SoftExpert Reunión: es un software para control de reuniones corporativas. Fue desarrollado para optimizar todo el proceso de gestión de las reuniones, desde la planificación, hasta la ejecución y finalización, agilizando las actividades de todos los colaboradores y participantes involucrados. Permitiendo el acompañamiento de las reuniones por medio de calendarios, o de las notificaciones enviadas por e-mail [3].

Luego de describir brevemente las soluciones de software similares al proyecto que se pretende implementar, se realizará una comparativa de funcionalidades entre cada uno de ellos y la solución propuesta, tomando como referencia los siguientes criterios de comparación:

Criterios de comparación

- Planificación de reuniones: el software permite establecer título, objetivos de la reunión y actividades a realizar.
- Programación de reuniones : el software proporciona hacer programaciones de reuniones con antelación.
- Calendarización de reuniones: el software posibilita llevar registro de las reuniones en un calendario.
- Registro de asistencia: el software da la opción de registrar quien asistió a la reunión.
- Evaluación de reuniones: el software permite evaluar la calidad de la reunión al final de su realización.
- Notificaciones: el software cuenta con un sistema de notificaciones.
- Generación de métricas y reportes: el software es capaz de generar métricas y reportes de las reuniones realizadas.

- Aplicación web y móvil: el software cuenta con una aplicación web y aplicación móvil.
- Uso: el tipo de uso que se le da al software, es decir, si es usado por alguna institución, empresa, corporación, industrial, etc.

Cuadro 2.1: Tabla comparativa de software similar al propuesto

Criterios	Softwares			
	SGCAE	Calendly	Doodle	SoftExpert Reunión
Planificación de reuniones	Si	Si	Si	Si
Calendarización de reuniones	No	Si	No	Si
Registro de asistencia	Si	No	No	SI
Generación de métricas y reportes	Si, Resumen del proceso.	Si, métricas y reportes.	No	Si
Notificaciones	Si, Notificaciones Push	Si	Si, Email y SMS	Si, Email
Aplicación web y móvil	Si	No	No	No
Uso	Institución Educativa	Empresarial	General	Corporativo

Como se aprecia en el Cuadro 2.1, la mayor parte de los software comparados cumplen con las funcionalidades principales que serán desarrolladas en la solución propuesta, pero cada uno de estos si bien tiene el mismo fin que es gestionar reunio-

nes, se les da un enfoque diferente los usuarios que lo van a utilizar.

La solución propuesta se diferencia de los demás, ya que propone un sistema con funcionalidades específicas para una institución de educación y sus ayudas estudiantiles. Además que la solución involucra más procesos que solo gestionar reuniones, por que estos software cubren solo una parte de los procesos que se quieren agilizar.

Por otra parte tenemos software de ofimática que nos permite llevar control de los procesos usando hojas de cálculo como:

Microsoft Excel: que es un software de Microsoft Corp, el cual nos permite llevar tareas contables y/o financieras trabajando con hojas de cálculo. La ventaja de usar este software es que posee una gran variedad de funciones y la posibilidad de realizar filtros en la información alojada en una hoja de cálculo. La desventaja es que la información gestionada puede verse dispersa en varias hojas de cálculo, dificultando visualizar y obtener información. Otra desventaja es que los datos deben ser ingresados de forma manual por el usuario.

Hojas de cálculos en Google Drive: al igual que el software anterior permite realizar operaciones contables y/o financieras, con la diferencia que este permite sincronización en la nube, queda un respaldo que puede ser visualizado por más de un usuario a la vez. La desventaja es que de igual forma el usuario debe ingresar los datos de forma manual y pueden estar dispersos en varios archivos en la nube.

Estos dos software si bien permiten gestionar los procesos actuales no son la mejor opción ya que solo lleva una parte de los procesos, sumado a la cantidad de archivos que se puede generar es considerable, por lo que demora demasiado encontrar la información que se requiere dentro de estos archivos. En cambio la solución propuesta tiene como fin que estos datos estén centralizados y sean generados de forma semi-automática.

2.2. Ayudas estudiantiles

Dentro de la Universidad de Talca existen diferentes tipos de ayudas estudiantiles que apoyan el avance académico de los estudiantes. El proyecto se trabajarán con 3 de ellas, las cuales se describen a continuación.

2.2.1. Conceptos relacionados

Ayudantías: es un apoyo que se imparte en la mayoría de los módulos de forma complementaria a las cátedras dictadas por cada docente. Esta la realiza un estudiante, el cual elabora actividades de repaso.

Tutorías docente: es un tipo de ayuda brindada por la escuela de Ingeniería Civil en Computación. En este un docente realiza tutorías a un grupo de estudiantes en riesgo. Este les brinda orientación durante un periodo de tiempo con el fin de acompañarlo y ver el avance que tiene el estudiante.

Tutorías estudiantiles: es un tipo de ayuda donde un estudiante está encargado de un grupo de estudiantes dándoles apoyo académico.

2.2.2. Procesos de gestión de ayudas estudiantiles

Cada una de las ayudas estudiantiles descritas en la apartado anterior son gestionadas por la dirección de escuela al inicio de cada semestre, siguiendo el proceso que se describirá posteriormente.

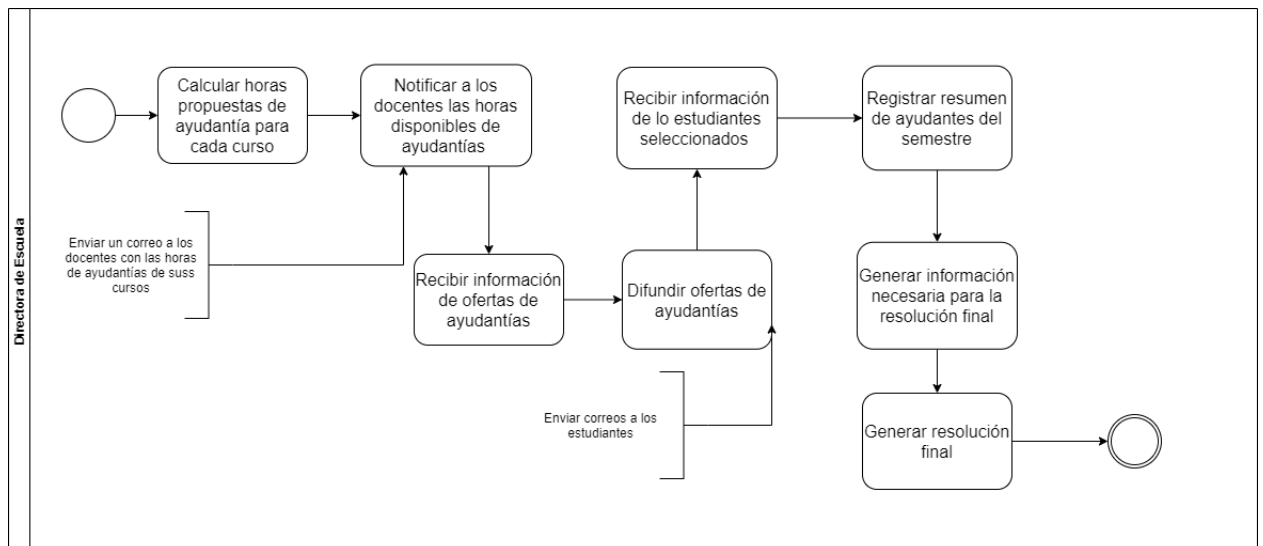


Figura 2.1: Diagrama de modelado de negocio del proceso de gestión de ayudantías por parte de la directora

Ayudantías En la Figura 2.1 se representa el modelado del proceso que lleva a cabo la dirección de escuela para realizar la gestión de ayudantías, siguiendo las siguientes tareas.

- **Calcular horas propuestas de ayudantías:** esta tarea la realiza la directora de escuela, con el fin de calcular un número estimado de horas que debe cumplir un ayudante, esto dependiendo de la cantidad de estudiantes inscritos y las horas de ayudantías y/o laboratorios de cada curso.
- **Notificar a los docentes las horas que tiene disponible de ayudantías:** al docente se le envía un correo con la cantidad de horas estimadas disponibles de ayudantías para que realice un ayudante. Una vez que el docente las ha recibido, el puede solicitar un aumento de horas, que debe aprobar la directora de escuela.
- **Recibir ofertas de ayudantías** la directora de escuela recibe la información del docente para difundir las ayudantías, donde es indicado las tareas a realizar, el horario que debe tener disponible el ayudante, entre otra observación que quiera dar el docente para generar la oferta de ayudantía.
- **Difundir ofertas de ayudantía:** las ofertas generadas son difundidas a los estudiantes mediante el correo institucional, con el fin que ellos postulen.
- **Recibir información de los estudiantes seleccionados:** la dirección de escuela debe recibir la información de los estudiantes que ha sido seleccionados como ayudante de cada curso.
- **Registrar resumen de ayudantes del semestre:** la directora de escuela generar un archivo excel con la información de los ayudantes del semestre.
- **Generar información necesaria para la resolución final:** en esta tarea se generar un archivo excel con los montos de todos los ayudantes.
- **Generar resolución final:** finalmente se genera la resolución final, detallando los montos de pagos a los ayudantes, entre otra información que puede justificar esa información.

Luego tenemos el proceso que debe seguir el docente con el fin de encontrar los ayudantes para cada uno de sus cursos.

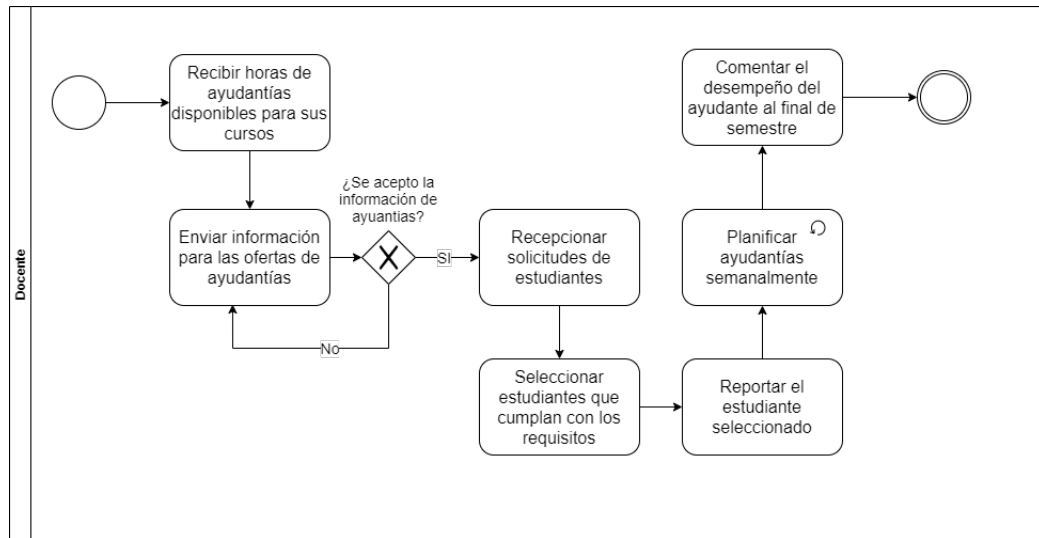


Figura 2.2: Diagrama de modelado de negocio del proceso de gestión de ayudantías por parte del docente

En la Figura 2.2 se ve representado el proceso que debe seguir un docente para encontrar un ayudante a sus cursos, las tareas que contempla se describen a continuación.

- **Recibir horas de ayudantías disponibles para sus cursos:** el docente recibe un correo con la cantidad de horas que tiene para proponer de ayudantías en cada curso que tiene asignado, teniendo la posibilidad de pedir un aumento, si el lo estima pertinente.
- **Enviar información para las ofertas de ayudantías:** el docente debe enviar las tareas que se realizarán, el horario de disponibilidad que debe tener los estudiantes, además de alguna otra observación que estime pertinente.
- **Recepcionar solicitudes de estudiantes:** una vez difundido las ofertas de ayudantías a los estudiantes, estos realizan sus postulaciones, las cuales son recibidas por el docente mediante correo electrónico.
- **Seleccionar estudiantes que cumplan con los requisitos:** el docente debe seleccionar un estudiante, el cual debe cumplir con los requisitos de haber

aprobado con nota superior 5,0 el curso al cual esta postulando. Una vez que el docente a escogido su candidato a ser ayudante, debe contactarse con el.

- **Reportar los estudiantes seleccionados:** el docente debe reportar a la directora de escuela los estudiantes que selecciono para que sean ayudantes durante el semestre.
- **Planificar ayudantías semanalmente:** el docente en conjunto con su ayudante generalmente, planifican ayudantías y/o laboratorios semanalmente.
- **Comentar el desempeño del ayudante al final del semestre:** el docente hace un comentario del desempeño que tuvo el ayudante durante el semestre.

Tutorías Las personas involucradas en este proceso de gestión de tutorías son la directora de escuela, los estudiantes y los tutores, siendo así a continuación se presentan los modelos de negocios del proceso llevado por cada uno.

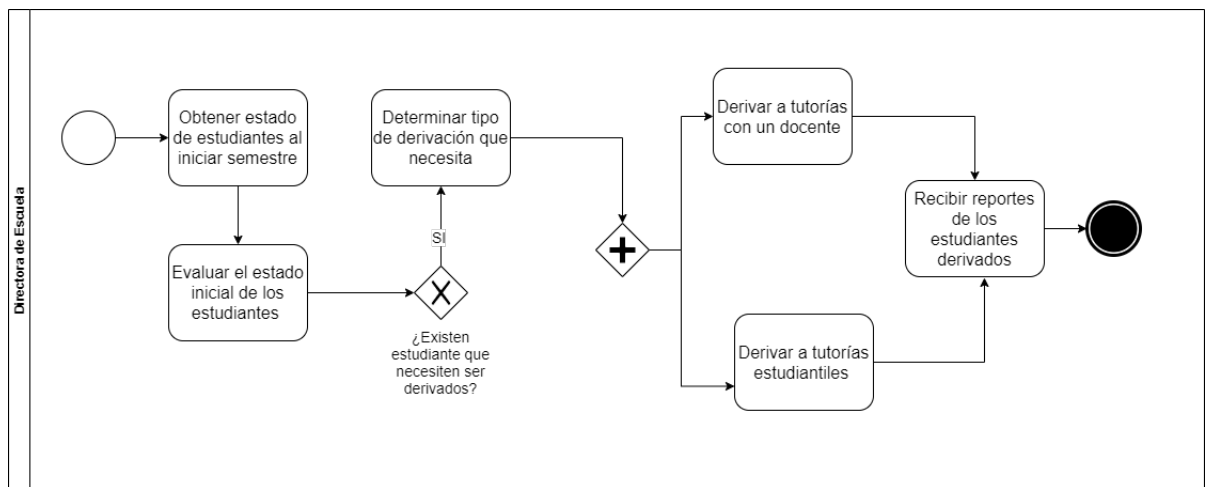


Figura 2.3: Diagrama de modelado de negocio del proceso de gestión de tutorías docentes para el rol de directora de escuela

La Figura 2.3 muestra el proceso que sigue la directora de escuela para gestionar las tutorías, las que pueden ser tutorías docente o tutorías estudiantiles, mostrando las tareas que debe realizar, siendo las siguientes.

- **Obtener estado de estudiantes al iniciar el semestre:** la directora de

escuela revisa el estado de los estudiantes, si alguno esta en riesgo para ser derivado a tutorías.

- **Evaluar el estado inicial de los estudiantes:** la directora analiza los casos para determinar que tipo de derivación necesitan.
- **Determinar el tipo de derivación que necesita:** se determina cual es la situación del estudiante, y se elige a quien derivarlo.
- **Derivar a tutorías con un docente:** el estudiante es derivado con un docente, con el fin de que este le brinde orientación y lo supervise durante el semestre.
- **Derivar a tutorías estudiantiles:** el estudiante es derivado a otro estudiante que será su tutor, el cual le brinda apoyo académico en los módulos que le son mas difíciles.
- **Recibir reportes de los estudiantes derivados:** al final del semestre los tutores deben reportar la evolución que percibieron en los estudiantes que tiene a cargo.

Por otro lado tenemos un desglose en el proceso de gestión de las tutorías dependiendo del tipo de derivación, encontrándonos con tutorías docentes o tutorías estudiantiles. Partiendo con el proceso de tutorías docentes donde tenemos involucrados a los estudiantes y el docente a cargo de estos, los cuales deben seguir el siguiente proceso.

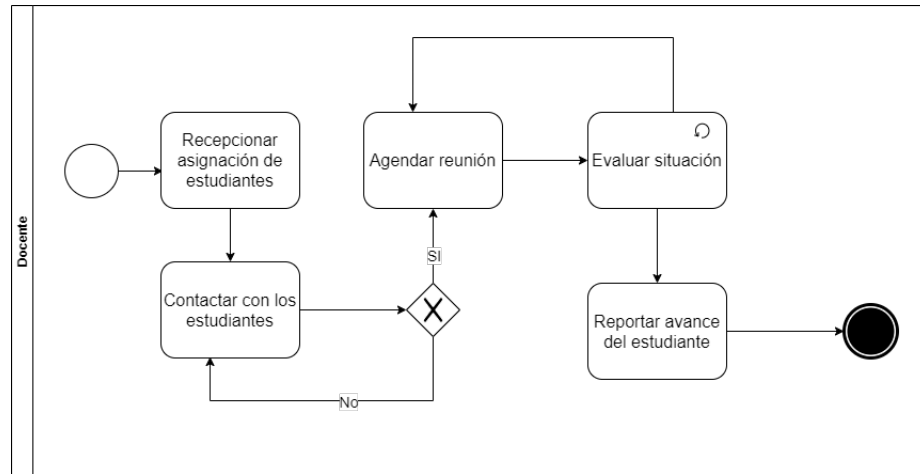


Figura 2.4: Diagrama de modelado de negocio del proceso de gestión de tutorías docentes llevado por el docente

En la Figura 2.4 podemos observar el proceso que lleva un docente cuando es asignado como tutor a un estudiantes, tiene las siguientes tareas a realizar.

- **Recepcionar asignación de estudiantes:** el docente recibe un correo con asignación de estudiantes al inicio del semestre.
- **Contactar con los estudiantes:** el docente una vez que ha sido asignados a estudiantes, debe contactarlos con el fin de reunirse periódicamente.
- **Agendar reunión:** el docente agenda reuniones con los estudiantes periódicamente para ver el avance del estudiante.
- **Evaluar situación:** el docente cada sesión evalúa la situación actual del estudiante.
- **Reportar avance del estudiante:** al final del semestre el docente debe reportar si este se logró contactar con los estudiantes y alguna otro observación que estime pertinente.

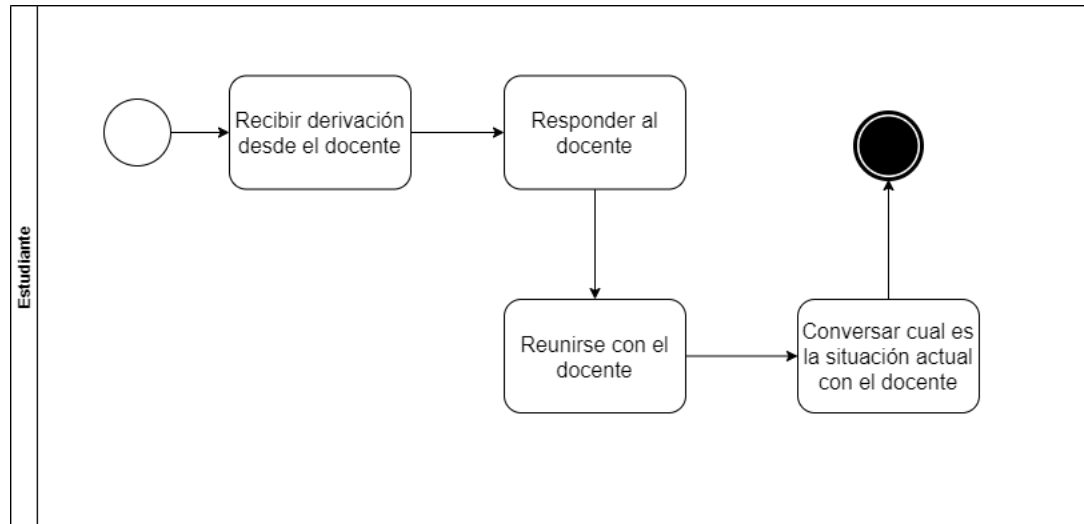


Figura 2.5: Diagrama de modelado de negocio del proceso de gestión de tutorías docentes llevado por el estudiante

La Figura 2.5 muestra el proceso seguido por un estudiante que ha sido derivado a tutorías docentes, el cual debe realizar las tareas que se describen a continuación.

- **Recibir derivación desde el docente:** el estudiante es contactado por el tutor docente que le fue asignado, a través de correo electrónico.
- **Responder al docente:** el estudiante le responde al docente tutor, con el fin de agendar una reunión.
- **Conversar cual es su situación actual con el docente**

Por otra parte nos encontramos con el proceso llevado para gestionar las tutorías estudiantiles, en el cual intervienen los estudiantes y un tutor que también es un estudiante. Estos deben desarrollar las tareas que se describen a continuación.

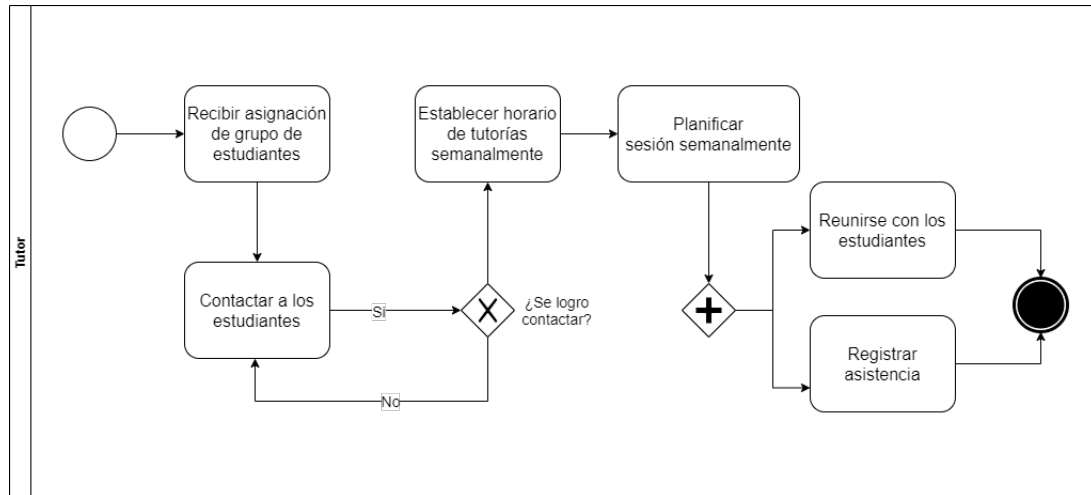


Figura 2.6: Diagrama de modelado de negocio del proceso de gestión de tutorías estudiantiles llevado por el tutor

La Figura 2.6 representa el proceso que debe realizar un tutor estudiante dentro del gestión de tutorías estudiantiles, el cual es expuesto a continuación.

- **Recibir asignación de grupo de estudiantes:** el estudiante tutor recibió por correo electrónico la lista de estudiantes que le fueron asignados.
- **Contactar los estudiantes:** el tutor estudiantil debe contactar a sus estudiantes, generalmente es hecho por medio del correo electrónico.
- **Establecer horario de tutorías semanalmente:** en conjunto con los estudiantes a cargo el tutor elige un horario semanal en el cual se realizan las sesiones de tutorías.
- **Planificar sesión semanalmente:** el tutor debe planificar la sesión semanalmente, preparando lo que se verá dentro de esta.
- **Reunirse y tomar asistencia:** el tutor ejecuta la sesión de tutoría, se reúne con los estudiantes y debe tomar asistencia a los estudiantes.

Por último tenemos el proceso seguido por estudiante que fue derivado a tutorías estudiantiles, el cual es especificado en la figura siguiente.

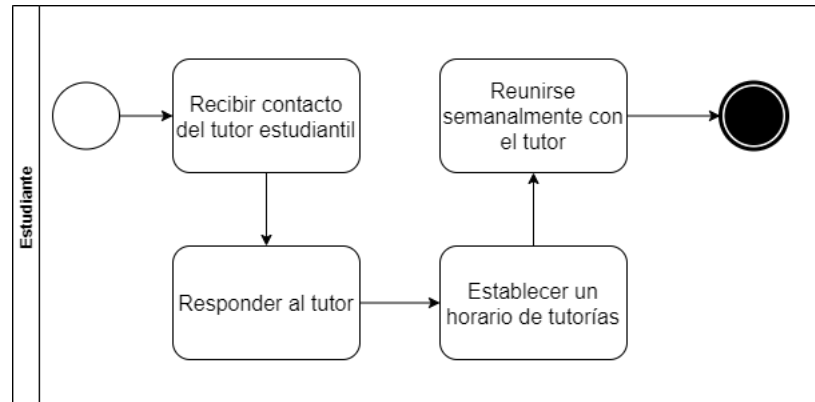


Figura 2.7: Diagrama de modelado de negocio del proceso de gestión de tutorías estudiantiles llevado por el estudiante

En la Figura 2.7 se presenta las tareas que debe realizar un estudiante dentro del proceso de tutorías estudiantiles, siendo las siguientes:

- **Recibir contacto del tutor estudiantil:** el estudiante recibe el correo de contacto del tutor estudiantil.
- **Responder al tutor:** el estudiante confirma el contacto con el tutor, fijando una reunión.
- **Establecer un horario de tutorías:** el estudiante junto al tutor escogen un horario para realizar las sesiones de tutorías.
- **Reunirse semanalmente con el tutor:** el estudiante asiste a las sesiones de tutorías semanalmente.

2.3. Desarrollo del software

La parte más fundamental para la implementación de la solución es el desarrollo de software que se llevará a cabo. Actualmente existen diversas formas de llevar un desarrollo de software a la vida real, debido a que existen diferentes formas de interacción dependiendo del dispositivo, sistema operativo y problema a solucionar. Para este caso el sistema a desarrollar consta de dos aplicaciones una web y otra móvil, las cuales tendrá múltiples interacciones diferenciadas por el tipo de usuario que las esté usando.

Para comprender de mejor manera el desarrollo y la forma de solucionar el problema, es necesario explicar ciertos conceptos, que se detallan a continuación describiendo las tecnologías utilizadas para la aplicación web y móvil respectivamente.

2.3.1. Conceptos relacionados

Software Frameworks: Un framework es considerado una plataforma concreta o conceptual donde se utiliza código común para generar funcionalidades genéricas, estas pueden ser selectivamente especializadas por usuarios o desarrolladores de software. Generalmente los framework toman forma de librerías que tiene bien definida una interfaz de la aplicación del programa, de modo que sea reutilizable en cualquier parte del código del software [4].

Existe una gran variedad de frameworks dependiendo de las tecnologías utilizadas, las ventajas y desventajas son las que se mencionan a continuación [4].

Ventajas

- Reutilización de código preconstruido y comprobado previamente. Aumenta la confiabilidad de nuevas aplicaciones y reduce el tiempo de programación y comercialización.
- Ayuda a establecer buenas prácticas de programación, uso apropiado de patrones de diseño y nuevas herramientas de programación.
- Puede proporcionar una nueva funcionalidad, un rendimiento mejorado o una calidad mejorada sin tener que programar de más.
- Puede extender el comportamiento del desarrollo de un software.

Desventajas

- Creación toma demasiado tiempo y esfuerzo.
- La curva de aprendizaje puede ser elevada.
- Con el tiempo puede llegar ser más complejo.

Aplicación en la nube: es una aplicación que delega el procesamiento y almacenamiento de datos a un tercero. Este brinda la infraestructura y almacenamiento

necesario, permitiendo acceder a esa información en tiempo real desde cualquier parte [5].

Base de datos en tiempo real: es una base de datos diseñada para manejar cargas de trabajo cuyo estado cambia constantemente. Este sistema difiere de las bases de datos tradicionales que contienen datos persistentes, en su mayoría no afectados por el tiempo. Donde las transacciones tienen plazos o limitaciones de tiempo. Por este motivo se utiliza en aplicaciones que necesitan acceder a los datos de forma oportuna [6].

Notificaciones push: estas son un mecanismo basado en eventos donde el servidor remoto envía eventos a las aplicaciones clientes, que están instalados en los dispositivos móviles [7].

Android: es un sistema operativo móvil diseñado por Google de código abierto, el cual está basado en un kernel Linux. [8].

2.3.2. Tecnologías aplicación web

Una aplicación web es una aplicación basada en la arquitectura cliente/servidor, donde el cliente es el usuario que accede desde un navegador web a los recursos almacenados en el servidor web, donde esta alojado el código de aplicación en sí. El cliente se comunica con el página web a través de internet donde esta alojado el servidor [9]. Hoy en día existe una gran cantidad de herramientas y tecnologías para desarrollar una aplicación web, que simplifican de gran manera la implementación y estructuración.

La siguiente sección entrega información sobre las tecnologías utilizadas en el desarrollo de la aplicación web. Esto incluye lenguajes de programación, librerías y frameworks principales para su implementación.

Angular

Para la implementación del lado del cliente se utilizó Angular 9. Este es un framework web, mantenido por Google y desarrollado en TypeScript, el cual nos permite el desarrollo de aplicaciones web de una sola página, el cual es muy completo y la mayoría de las veces no es necesario utilizar librería externas. Este utiliza el patrón MVC para la estructuración de un proyecto.

Otra alternativa que se barajada, es la utilización de React, el cual es una librería para desarrollo UI basada en Javascript. Al ser una librería es necesario incluir un sin fin de librerías para dar funcionalidades más complejas a una aplicación web.

La elección de usar Angular para la implementación de la presentación al cliente es la estructuración del proyecto, el uso de dependencias y componentes lo hace más reutilizable al usar el modelo vista controlador. En cambio React deja a libre elección al desarrollador la estructuración del proyecto lo que lo hace susceptible a malas practicas. Otra de las razones es el dominio previo en Angular si bien no tiene un curva de aprendizaje baja, ya se ha trabajado anteriormente con este, en proyectos y cursos en la Universidad, lo agilizó el tiempo de aprendizaje que haber empezado desde cero a aprender React.

2.3.3. Tecnologías aplicación móvil

Una aplicación móvil es una aplicación adaptada para ser utilizada en un dispositivo móvil, el cual tiene dimensiones mucho más cometidas que un dispositivo de escritorio o laptop, aprovechando los recursos y periféricos del dispositivo para realizar aplicaciones multipropósito.

En la actualidad se ha masificado el desarrollo de aplicaciones móviles en todas las áreas por la versatilidad que ofrecen. Por este motivo se escogió desarrollar una aplicación móvil, aparte de la plataforma web.

A continuación se describen las tecnologías utilizadas para el desarrollo móvil

Android

Es un sistema operativo y una plataforma de software, para dispositivos móviles, basado en Linux. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java o Kotlin, aplicaciones sobre una máquina virtual Dalvik 2. Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso, modificar el propio sistema operativo, dado que Android es de código libre [10].

Otra alternativa a considerar fue desarrollar una aplicación móvil para el sistema operativo iOS de la marca Apple. iOS, al igual que Android, cuenta con un SDK

que permite a terceros desarrollar aplicaciones móviles nativas con acceso a distintos componentes del dispositivo móvil.

La elección de desarrollar solamente para el sistema operativo Android, porque es multiplataforma, puede ser desarrollado usando cualquier sistema operativo Windows (mayor a XP), Mac OS X (10.5.8 o mayor) y Linux (con kernel 2.6 o superior). En cambio para desarrollar en iOS se necesita computadores Macintosh que tengan sistema operativo Mac OS X 10.6.

Kotlin

Este es un moderno lenguaje compatible con Android que arregla muchos errores de Java como null pointer exceptions o código excesivo. Kotlin es un lenguaje inspirado por Swift, Scala, Groovy, C# y mucho otros lenguajes. Otras alternativas a utilizar es Java, lenguaje de programación orientado a objetos, desarrollado por James Gosling y su equipo en Sun Microsystems en California, el cual tiene un amplio recorrido en el desarrollo móvil.

La razón porque se escogió Kotlin en ves de Java, fue por que actualmente las aplicaciones móviles nativas esta siendo migradas a Kotlin gracias a las mejoras que trae consigo, como por ejemplo el uso más eficiente de hilos utilizando las denominadas corrutina, lo que simplifica trabajar con otro hilos que no sean el hilo destinado a la interfaz gráfica, lo que evita que sufra fallos inesperados o errores productos de un mal uso del hilo principal. Además de la reducción de código, es decir, hace lo mismo en código en Java pero en menos lineas.

Entorno de programación Android Studio

Android Studio es un IDE que fue desarrollado por Google y que permite al desarrollador usar una herramienta que facilita el proceso de creación de una aplicación en Android. Android Studio es el IDE oficial para desarrollar en esta plataforma.

Otra alternativa a usar es el IDE Eclipse, el cual nos permite también desarrollar aplicaciones móviles en Android, pero el cual no tiene soporte nativo para emulación de dispositivos.

Se escogió Como se mencióno en la descripción de la herramienta Android Studio, este es el IDE oficial para desarrollar en Android. Al ser el IDE oficial y de la empresa Google tiene más aceptación y cuenta con más referencia en caso de problemas

funcionalidades del proyecto. Usando Firebase Authentication y Firebase Cloud Messaging para la administración de usuarios y envío de notificaciones entre dispositivos y plataforma web. Por otra parte, la suite nos permite hacer uso de una base de datos en tiempo real No SQL llamada Firestore, la cual nos permite realizar consultas a los datos al instante. Si la comparamos con una base de datos tradicional MySQL, haciendo uso de un CRUD (Create Read Update and Delete) el tiempo de respuesta en las operaciones de creación, lectura, actualizaciones y eliminación de registros es menor según una comparación realizada en International Conference on Computer Science and Computer Intelligence en 2019, utilizando el método Wilcoxon Signed-Rank Test [11]. Esta es una de las razones principales porque se escogió la suite de desarrollo Firebase, frente a la utilización de un backend convencional con una base de datos relacional, ya que unos de los objetivos del proyecto era minimizar el tiempo de obtención de información, sumado a las razones anteriormente mencionadas es el porque se eligió Firebase.

2.4. Metodología de desarrollo

Uno de los puntos importantes para la planificación del proyecto es la metodología de trabajo que se utilizará, particularmente para un proyecto de desarrollo de software. Para el cual se usará una metodología ágil, la que establece una serie de características, las que son: [12].

- Planificación adaptativa.
- Desarrollo iterativo y evolutivo.
- Rápida y flexible ante cambios.
- Promueve la comunicación.

Teniendo en cuenta las características anteriores, se ha decidido utilizar la metodología Scrumban, caracterizada por adoptar aspectos tanto de Scrum y Kanban.

Para hacer uso de esta tenemos que tener en cuenta los siguientes elementos [13].

- **Iteraciones:** tienen una duración corta que no supera más allá de dos semanas. Esto permite poder adaptarse a un entorno cambiante.

- **Planificación de trabajo bajo demanda:** se basa en la demanda de tareas pendientes. Las que se activan cuando sobrepasa un número determinado de tareas pendientes, definido por el equipo.
- **Priorización:** se puede hacer agregando números a las tareas u ordenando tareas por prioridad en la columna, donde las tareas más importantes se colocan en la parte superior, y a continuación las tareas menos importantes, siendo clasificadas con una escala de criticidad, donde el valor mas alto es Crítica y el valor menor valor es Muy Bajo.
- **Tablero:** el tablero básico de Scrumban se compone de tres columnas: tareas pendientes, en proceso y listo. Después de la Reunión de Planificación, las tareas se agregan a la columna tareas pendientes; cuando un miembro del equipo está listo para trabajar en una tarea, la mueve a la columna en proceso y cuando la completa, la mueve a la columna listo.

Esta metodología sigue un proceso iterativo involucrando todos los elementos antes mencionados los cuales son utilizados en proceso que se debe seguir para aplicarla. Este es descrito se a posteriormente en la Figura 2.8

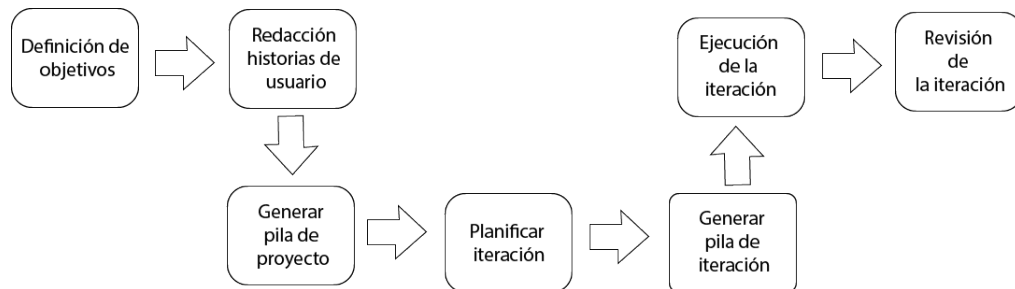


Figura 2.8: Proceso para seguir la metodología Scrumban

1. **Definición de objetivos:** se establecen los objetivos del proyecto a largo plazo, generalmente a un año.
2. **Redacción de historias de usuarios:** se realiza la especificación de historias de usuario, su priorización y su estimación.
3. **Generación de pila de proyectos:** se genera la lista de todas las tareas que se realizaran durante el proyecto.

4. **Planificación de la iteración:** se planifica la iteración actual, se eligen las tareas a realizar.
5. **Generación de la pila de la iteración:** se genera la lista de tareas que serán implementadas durante la iteración.
6. **Ejecución de la iteración:** se diseñan, desarrollas y prueban las tareas que se tiene prevista para cada iteración.
7. **Revisión de la iteración:** al final de la iteración se hace una retrospectiva para saber el estado en que quedo la iteración, si se desarrollaron todas las tareas o no.

2.5. Metodología de evaluación

Aparte de la metodología de desarrollo también existe la de evaluación, la cual verifica el cumplimiento de los objetivos propuestos en el proyecto. Para esto se utilizará tres tipos de metodología de evaluación las cuales son:

1. **Pruebas unitarias:** las pruebas unitarias de software permiten evaluar por separado el correcto funcionamiento de los códigos que lo componen [14]. Generalmente se le llama una unidad de código a un módulo del sistema.
2. **Pruebas de usabilidad:** son utilizadas para evaluar la interacción de usuario con el sistema desarrollado, si para él fue fácil utilizarlo o no. El método a utilizar será el de pruebas de usabilidad de escalas. El cual consiste en una encuesta simple de diez enunciados que miden subjetivamente la usabilidad. El usuario deberá calificar de acuerdo a su nivel de acuerdo o desacuerdo 10 incisos usando una escala Likert de cero a cuatro [15].
3. **Encuestas:** es un conjunto de preguntas dentro de un cuestionario para obtener información y opinión de personas, en este caso usuarios de la aplicación. Para este aspecto se harán encuestas antes y después de haber implementado el sistema, de esta manera se busca medir si el rendimiento y/o tiempo se ha reducido tras el uso del sistema.

Estas varían dependiendo del objetivo a evaluar.

2.6. Resumen del capítulo

En este capítulo se hizo una revisión de los trabajos relacionados que existen en el mercado y que son similares a la solución propuesta, describiendo sus soluciones, para luego compararlas entre si mediante un conjunto de criterios. Posteriormente se explicó el proceso implementado por el área de coordinación de la escuela de Ingeniería Civil en Computación, para impartir las ayudas estudiantiles durante cada semestre. Finalmente se habló sobre los conceptos relacionados y herramientas utilizadas tanto para la aplicación web administrativa y la aplicación móvil que se desarrollará. A continuación en el, capítulo tres, se hablará sobre como se ajustará la metodología de desarrollo para aplicarla al proyecto, para luego detallar las historias de usuarios capturadas, su prioridad, y su estimación de esfuerzo para definir cada iteración.

3. Metodología de desarrollo

En el capítulo tres se habla como se ajustó la metodología de desarrollo para ser aplicada en el proyecto. Posteriormente se expondrán las historias de usuario capturadas desde el cliente, la priorización y estimación hecha por cada historia de usuario, explicando las técnicas utilizadas. Para finalmente hablar sobre la definición de los elementos de metodología.

3.1. Metodología

Como se mencionó anteriormente, en el capítulo dos, la metodología es una parte fundamental del desarrollo de software, si la elección no se adapta correctamente a la solución que queremos desarrollar, la probabilidad de que el proyecto sea exitoso se reduce considerablemente. En este caso, para el proyecto se escogió la metodología ágil Scrumban, la que nos permite olvidarnos de la definición de roles, ceremonias y el mínimo de personas necesarias para poder utilizarla.

Tomando en cuenta lo anterior y que el proyecto a realizar es desarrollado por una sola persona y las funcionalidades pueden variar durante el desarrollo de este, se escogió implementar dicha metodología de software.

3.2. Metodología de desarrollo ágil Scrumban

La metodología Scrumban combina aspectos de las metodologías de desarrollo Scrum y Kanban, mejorando las falencias de cada una de estós métodos de desarrollo. Scrumban está compuesta de una serie de elementos que hay que tener en cuenta para aplicarla, dichos elementos son explicados luego.

3.2.1. Product Backlog

Un product backlog no es más que la lista de todas las historias de usuarios obtenidas en la captura de requisitos desde el cliente, donde cada una de estas se desglosan en pequeñas tareas que conforman la lista de tareas que se realizarán durante todo el proyecto.

En el caso concreto del proyecto se establecieron 25 diferentes historias de usuarios, las cuales se dividieron según el tipo de ayuda estudiantil y usuario a la cual apuntan. Estas se detallan en los siguientes cuadros.

Cuadro 3.1: Historias de usuario ayudantías

ID	Rol	Enunciado	Entorno
H001	Como ayudante,	necesito planificar mis ayudantías, con la finalidad de prepararlas de mejor manera.	Móvil
H002	Como un ayudante,	necesito llevar un registro de la asistencia de las ayudantías, con la finalidad de controlar la cantidad de estudiantes que asisten a la ayudantía	Móvil
H003	Como un estudiante,	necesito evaluar a un ayudante al final de semestre, con la finalidad de dar retroalimentación para mejorar la calidad de las ayudantías.	Móvil
H004	Como directora,	necesito descargar reportes en excel de las evaluaciones hechas en las ayudantías, con la finalidad de comprobar el desempeño de los ayudantes	Web
H005	Como directora,	necesito visualizar la planificación, asistencia y evaluaciones realizadas en las ayudantías, con la finalidad de ver la utilidad de las ayudantías.	Web

H006	Como asistente,	necesito realizar la propuesta de horas de ayudantía de cada curso con la finalidad de establecer un cantidad definida de horas de ayudantías para los cursos.	Web
H007	Como asistente,	necesito difundir las ayudantías a los estudiantes, con la finalidad de que los estudiantes se postulen como ayudante.	Web
H008	Como asistente,	necesito recepcionar los estudiantes que ha sido seleccionados como ayudantes, con la finalidad de generar la información necesaria para la resolución final.	Web
H009	Como asistente,	necesito generar el resumen de ayudantes del semestre con la finalidad de generar la información necesaria para la resolución final.	Web
H010	Como asistente,	necesito generar la información para la resolución final con la finalidad de justificar la gestión de ayudantes.	Web
H011	Como estudiante,	necesito postular a una ayudantía de un curso, con la finalidad de que me seleccionen como ayudante.	Móvil
H012	Como docente,	necesito enviar la selección de ayudantes de mis cursos, con la finalidad de tener registro de los estudiantes que he seleccionados para las ayudantías de mis cursos.	Móvil
H013	Como docente,	necesito evaluar a un ayudante, con la finalidad de tener registro del desempeño de los estudiantes como ayudantes.	Móvil

Cuadro 3.2: Historias de usuario tutorías docente

ID	Rol	Enunciado	Entorno
H014	Como docente tutor,	necesito recepcionar asignación de estudiantes, con la finalidad de contactar a los estudiantes que se me han asignado.	Móvil
H015	Como docente tutor,	necesito agendar reuniones con los estudiantes que tengo a cargo, con la finalidad de orientar al estudiante.	Móvil
H016	Como docente tutor,	necesito registrar anotaciones de las sesiones con los estudiantes que tengo a cargo, con la finalidad de tener registro de que se habló cada sesión.	Móvil
H0017	Como docente tutor,	necesito reportar el avance que tienen los estudiantes que tengo a cargo, con la finalidad de mantener al tanto el avance que tienen los estudiantes.	Móvil
H018	Como docente tutor,	necesito programar recordatorios de sesiones con el estudiante, con la finalidad de no olvidar asistir a las sesiones que tengo programadas	Móvil
H0019	Como estudiante,	necesito evaluar a un docente tutor al final del semestre, con la finalidad de dar retroalimentación para mejorar las tutorías docentes.	Móvil
H020	Como directora,	necesito descargar reportes en excel de las observaciones hechas por el docente tutor de los reportes que ha hecho, con la finalidad de comprobar la utilidad real de las tutorías.	Web

Cuadro 3.3: Historias de usuario tutorías estudiantiles

ID	Rol	Enunciado	Entorno
H021	Como docente estudiante,	necesito planificar mis tutorías, con la finalidad de prepararlas de mejor manera.	Móvil
H022	Como docente estudiante,	necesito registrar la asistencia de mis tutorías, con la finalidad de llevar registro, de cuantos asisten a las tutorías.	Móvil
H023	Como estudiante,	necesito evaluar al tutor estudiantil, con la finalidad de dar retroalimentación del desempeño de los tutores.	Móvil
H024	Como directora,	necesito descargar reportes en excel de las evaluaciones hechas en tutorías, con la finalidad de comprobar el desempeño de los tutores.	Móvil

Cuadro 3.4: Historias de usuario generales

ID	Rol	Enunciado	Entorno
H000	Como directora,	necesito cargar y visualizar los datos de estudiantes, cursos y docentes de cada semestre, con la finalidad de administrarlos.	Móvil
H022	Como directora,	necesito visualizar las evaluaciones y asistencia con la finalidad de comprobar la utilidad de las ayudas estudiantiles.	Móvil

3.2.2. Priorización y estimación de esfuerzo de historias de usuario

La priorización de tareas es dar un nivel de importancia a cada una de las historias de usuario. Existen varias técnicas para priorizar y estimar el esfuerzo que nos puede

llevar a realizar una tarea.

Para la implementación del proyecto se utilizó una priorización de importancia en una escala cualitativa de criticidad, siendo el valor más alto Crítico y el menor valor Muy Bajo. Por el lado de la estimación se aplicó la técnica de los 100 puntos, la cual consiste en asignar un puntaje a cada historia de usuario que al sumar de como resultado 100 puntos. Utilizando estas técnicas el resultado obtenido fue el siguiente.

Cuadro 3.5: Priorización y estimación de historias de usuario

ID historia usuario	Prioridad	Estimación de esfuerzo (Puntos)
H000	Crítica	2
H001	Crítica	3
H002	Alta	6
H003	Crítica	6
H004	Media	4
H005	Crítica	3
H006	Media	3
H007	Alta	2
H008	Crítica	3
H009	Crítica	5
H010	Crítica	3
H011	Alta	3
H012	Crítica	5
H013	Crítica	3
H014	Crítica	3
H015	Alta	5
H016	Alta	3
H017	Alta	2
H018	Media	6
H019	Media	5
H020	Crítica	5
H021	Alta	4

H022	Alta	5
H023	Alta	4
H024	Alta	5
H025	Alta	2

Una vez priorizado y estimado el esfuerzo se podrá realizar cada una de las historias de usuario, desglosándolas en tareas más pequeñas, las cuales serán desarrolladas en cada una de las iteraciones planificadas.

3.2.3. Planificación

En este elemento de la metodología se planifica las iteraciones que se llevarán a cabo, solo se planificará nuevamente si al término de una iteración quedan demasiadas tareas pendientes, en caso contrario se mantendrá la planificación inicial.

En el caso concreto del proyecto se estableció que si quedan más de cinco tareas pendientes, se debe planificar nuevamente. La planificación del proyecto se dividió en 8 iteraciones las cuales tendrán una duración variable de entre 1 a 3 semanas, dependiendo del desempeño que se logre en cada iteración. A continuación se muestra la planificación en el Cuadro 3.6.

Cuadro 3.6: Planificación del proyecto por iteraciones.

	Historia Usuario	Funcionalidades a Realizar
Iteración 1	H000-C-2	Autenticación de usuarios aplicación móvil.
		Carga masiva de datos estudiantes, cursos y docentes.
Iteración 2	H000-C-2	Auntenticación usuarios plataforma web
		Activar cuenta de usuario estudiante aplicación móvil.
		Visualizar datos de ayudantías por curso.
Iteración 3	H001-C-3	Agendar reuniones de ayudantías

Iteración 4	H002-A-6	Registro de planificaciones ayudantías
	H003-C-6	Registro de asistencia de ayudantías
		Evaluación de ayudantes
Iteración 5	H014-C-3	Asignar estudiantes a tutores docentes aplicación web
	H015-A-5	Recepcionar estudiantes aplicación móvil.
	H016-A-3	Registrar anotaciones de sesiones con estudiantes
		Agendar reuniones con estudiantes
Iteración 6	H006-M-3	Calcular propuesta de horas de ayudantía aplicación web.
	H009-C-5	Mostrar resumen de ayudantes aplicación web
	H010-C-3	Postular a un ayudantía estudiante aplicación móvil.
	H011-A-3	Seleccionar un ayudante aplicación móvil aplicación móvil.
	H013-C-3	
Iteración 7	H021-A-4	Registrar planificación de tutorías estudiantiles aplicación móvil.
	H022-A-5	Registrar asistencia de tutorías estudiantiles aplicación móvil.
	H023-A-4	Evaluar al tutor estudiantil aplicación móvil.
Iteración 8	H004-M-4	Reportar avances directora de escuela aplicación móvil.
	H005-C-3	Evaluar docente tutor aplicación móvil
	H006-M-3	Evaluar ayudante por parte del docente aplicación móvil.
	H007-A-2	Mostrar montos de ayudantes aplicación web
	H008-C-3	Mostrar evaluaciones, asistencia y planificaciones aplicación web.

	H18-M-6	Generar resúmenes de evaluaciones, y asistencia aplicación web.
	H019-M-5	
	H020-C-5	
	H024-A-5	
	H025-A-2	

En el cuadro 3.6 nos encontramos con las funcionalidades que se van desarrollar en cada una de las iteraciones, además de las historias de usuarios asociadas que se desarrollaran en cada iteración, en las cuales se indica su prioridad Critica (C), Alta (A), Media (M), Baja (B), Muy Baja (MB), seguido del puntaje de estimación.

3.2.4. Tablero Kanban

Un tablero no es más que una representación gráfica de las tareas que se tiene que desarrollar, esta dividido en cuatro columnas To-do, In progress, Test/Review y Done, las cuales se pueden ver a continuación en la Figura 3.1.

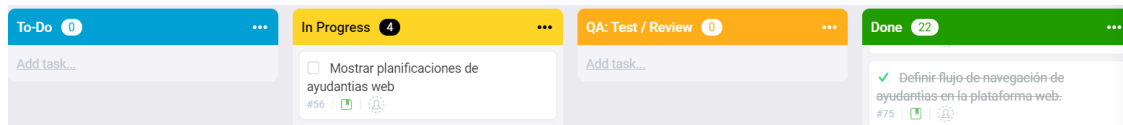


Figura 3.1: Tablero Scrumban

La columna To-do, hace referencia a las tareas que están pendientes en la iteración actual. Luego la columna In progress hace referencia a las tareas que están siendo realizadas cuando estas son terminadas se mueve la tarea a la columna Test/Review que indica que dicha tarea esta en testeo, una vez pasado el testeo la tarea finalmente es movida a la columna Done, donde están las tareas que han sido terminadas.

3.2.5. Pruebas Unitarias de Caja Negra

Una prueba de caja negra no es mas que una prueba a una unidad de código, donde se ingresan un conjunto de salidas y tenemos como resultado una salida esperada. Para el proyecto se documentan las pruebas de caja negra, utilizando el siguiente formato.

Cuadro 3.7: Formato de pruebas de documentación de pruebas unitarias

Nombre de la prueba	
Proposito	XXXXXXXXXXXXXXXX
Prerrequisitos	XXXXXXXXXXXXXXXXXXXX
Entradas	XXXXX, XXXXXX, XXXXX
Salidas	XXXXXXXXXXXXXXXXXXXX
Resultado de la prueba	XXXXXXXXXXXX.

En el Cuadro 3.7 se muestra el formato que seguirán la documentación de las pruebas, teniendo en cuenta el nombre de la prueba, el propósito, que prerrequisitos son necesarios para realizar la prueba, las entradas, salidas esperadas y el resultado de la prueba el cual puede ser incorrecto, medianamente correcto y correcto.

3.3. Resumen del capítulo

En este capítulo se revisó la forma en que se aplicará la metodología escogida, presentando los ajustes que se realizaron para el proyecto. Empezando con la captura de historias de usuario del cliente, posteriormente se procedió a redactar las historias de usuario, para así generar el product backlog o también llamado pila de proyecto. Siguiendo con la priorización y estimación de las historias de usuario. Por último se ilustró la utilización de un tablero Kanban y la distribución de columnas que se están utilizando.

En el capítulo cuatro se detallará el proceso de ejecución de las tareas programadas en cada una de las iteraciones del proyecto, comenzando con el diseño, luego la implementación, para luego terminar con las pruebas unitarias.

4. Desarrollo de la aplicación web

En el presente capítulo se aborda la manera en que la metodología es llevada a cabo, para guiar la fase de implementación del proyecto que corresponde al desarrollo de la aplicación web, donde se aborda la puesta en marcha de la metodología Scrumban y como la planificación de las iteraciones fueron de ayuda en la construcción del sistema. Dentro de este se expone el flujo de trabajo de la aplicación web, que nos permite comprender el flujo que sigue la aplicación. Posteriormente se presenta el diseño incluyendo la arquitectura física y lógica, además del diagrama de clases. Para finalizar con la ejecuciones de las iteraciones correspondientes a la aplicación web.

4.1. Flujo de trabajo de la aplicación web

En esta sección se detallará el flujo de trabajo que sigue la aplicación web, para esto se utilizará un diagrama de actividades el cual es una representación del flujo de una actividad a otra dentro de un sistema, además una actividad muestra un conjunto de actividades, mostrando el flujo secuencial o ramificado de una actividad a otra. Este tipo de diagrama se utiliza para mostrar una vista dinámica de un sistema y sus funciones [16].

La primera actividad que se realiza en la aplicación web es la carga masiva de datos, ya que es la base de información para llevar a cabo las demás actividades. Para esta actividad se debe seleccionar el archivo excel que contiene la nomina de estudiantes inscritos del semestre, la cual es parseada, trasformada y procesada finalmente en un formato JSON, una vez que se procesa esta se almacena en la base de datos.

El detalle de esta actividad se encuentra en el Anexo A.1.

Luego de tener la fuente de información carga en base de datos, tenemos el flujo que se produce en el proceso de ayudantías, donde se realiza el cálculo de horas propuestas de ayudantías, donde se debe seleccionar el curso al que se quiera calcular sus horas, luego se debe ingresar el número de horas de ayudantías y/o laboratorio impartidas, e internamente la aplicación calcula las horas propuestas, las cuales serán actualizadas en el registro del curso que se encuentra en base de datos.

El detalle de esta actividad se encuentra en el Anexo A.2.

Por otra parte tenemos el flujo de trabajo para el proceso de tutorías, donde se realiza la asignación de estudiantes a un docente. Donde primero se debe cargar la lista de estudiantes en riesgo, la cual es un archivo XLS con la matricula y nombre del estudiante. Una vez cargado los datos, se debe arrastrar un docente a la casilla de docente y los estudiantes que se le quieren asignar a la casilla de estudiante. Por ultimo se debe seleccionar el semestre y presionar el botón asignar. Este subproceso se documenta en el Anexo A.3.

Para finalizar tenemos el flujo que se sigue para el proceso de tutorías estudiantiles, el cual consiste en la asignación de estudiantes a un tutor estudiantil, al igual que la asignación de tutores docentes, se debe carga la lista de estudiantes en riesgo, además de cargar la lista de tutores estudiantiles, que es un archivo excel con la matricula y nombre de los tutores. Ya que han sido cargados, se debe arrastrar un tutor estudiantil a la casilla tutor y arrastrar los estudiantes que se quieren asignar a la casilla estudiantes. El detalle del diagrama de actividades se encuentra disponible en el el Anexo A.4.

4.2. Diseño de la aplicación web

Una vez capturadas y priorizadas las historias de usuario que están especificadas en el capítulo tres, se procedió a elegir la arquitectura física, diseñar la arquitectura lógica y el diagrama de clases del sistema. Además de presentar los detalles de la iteración cero, la cual contempla la instalación de los entornos de desarrollo y framework, adicionalmente las configuraciones iniciales para vincular el servicio de base de datos y autenticación de Firebase.

4.2.1. Arquitectura física

La arquitectura física intenta modelar la interacción y comunicación entre los componentes físicos que están involucrados en el proyecto, siendo estos los clientes y servidores como servidor web y servidor de base de datos, entre otros. Para este proyecto se eligió una arquitectura Cliente-Servidor, donde el cliente es un dispositivo de escritorio que es usado por la dirección de escuela. Esto es representado a continuación en la Figura 4.1.

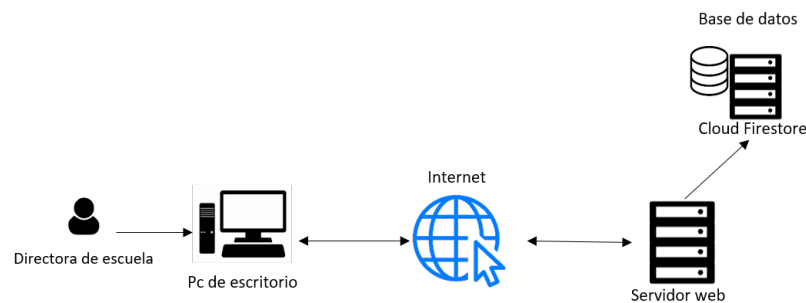


Figura 4.1: Arquitectura física - Cliente Servidor de la aplicación web

En la Figura 4.1 se ilustra cómo los clientes interactúan con el servidor de base de datos mediante peticiones y respuestas por medio de internet. Cada vez que uno de los clientes realice una petición este será procesada internamente y enviada al servidor de Firebase, la cual es recibida, procesada y enviada de vuelta con una al cliente. Un ejemplo de esto es una petición de creación de una asignación de un estudiante a un tutor docente, la asistente o directora de escuela, el servidor la procesa, es decir la almacena, y envía una respuesta de verificación de éxito o fallo de la petición.

4.2.2. Arquitectura lógica

La arquitectura lógica nos muestra la interacción de los componentes lógicos como subsistemas o macro-funcionalidades del sistema que participan en la solución propuesta para este proyecto y como se relacionan entre ellos. Para representar la arquitectura se descompuso los componentes cliente y servidor, en los subsistemas aplicación móvil y aplicación web. Para luego realizar un acercamiento más detallado a la aplicación web en este capítulo y en el capítulo cinco se detalla la aplicación móvil y sus subsistemas que las componen de forma independiente utilizando el patrón

arquitectónico MVC (Modelo Vista Controlador). Esta es simbolizado en la Figura 4.2 y Figura 4.3.

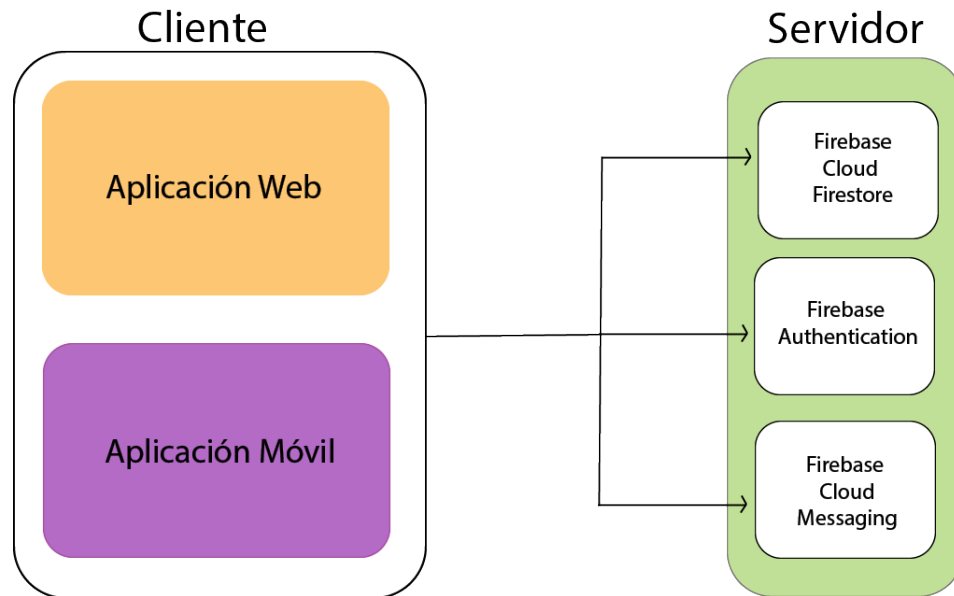


Figura 4.2: Arquitectura lógica - MVC del sistema a alto nivel

Por un lado en la Figura 4.2 podemos ver la descomposición del cliente y servidor a alto nivel, notando que el cliente aloja las aplicaciones móvil y web las cuales se comunican al servidor utilizando los subsistemas integrados en Firebase como Cloud FireStore, Authentication y Cloud Messaging.

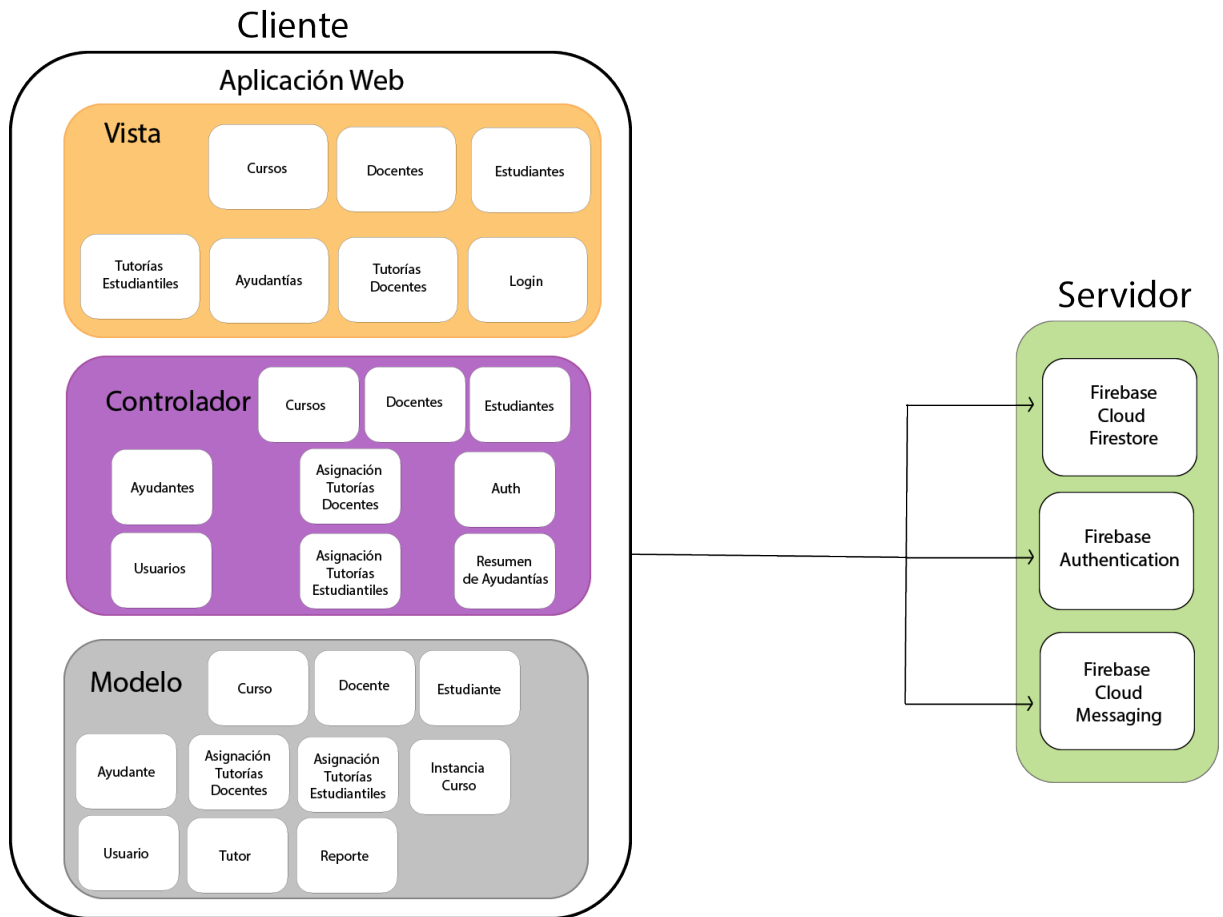


Figura 4.3: Arquitectura lógica - MVC aplicación web

Ahora bien en la Figura 4.3 se descompone la aplicación web en subsistemas, donde el cliente contiene las vistas, controladores y modelos. Encontramos las vistas del apartado visual de gestión de ayudantías, tutorías docentes y tutorías estudiantiles. Además de las vistas de cursos, docentes, estudiantes y login. Cada una de estas cumple la función de mostrar información al usuario. Luego tenemos los controladores, los cuales manejan métodos que el usuario invoca para realizar peticiones al servidor, ya sean para leer, escribir o eliminar información de la base de datos (Cloud Firestore), autenticar usuarios (Firebase Authentication) y enviar o recibir notificaciones entre aplicaciones (Firebase Cloud Messaging). Además encontramos los modelos, que implementan la lógica de la aplicación para posteriormente ser mostradas en la vista por medio de los controladores.

4.2.3. Diagrama de clases

La aplicación web se construye bajo el paradigma de programación orientado a objetos (POO), utilizando el Framework Angular que usa como base el lenguaje Typescript. Para realizar un buen diseño utilizando este paradigma es recomendable realizar un diagrama de clases. Un diagrama de clases es utilizado para representar una vista estática del sistema, componiéndose de clases y relaciones. Podemos decir que una clase es una descripción de un concepto y que tiene atributos y operaciones asociadas, estas son modeladas en el diagrama como un rectángulo. Mientras que una relación se modela como un conexión lógica entre dos clases y se muestra gráficamente como una línea que une ambas clases [17].

En POO, una clase es un abstracción de un conjunto de objetos que en la vida real comparten los mismo atributos, operaciones, relaciones y semántica. Si tomamos un ejemplo de una clase del proyecto , tenemos la clase Docente, la cual comparte características comunes para un docente como lo son su nombre, apellidos, email, entre otras. En POO, esto se conoce como atributos de una clase.

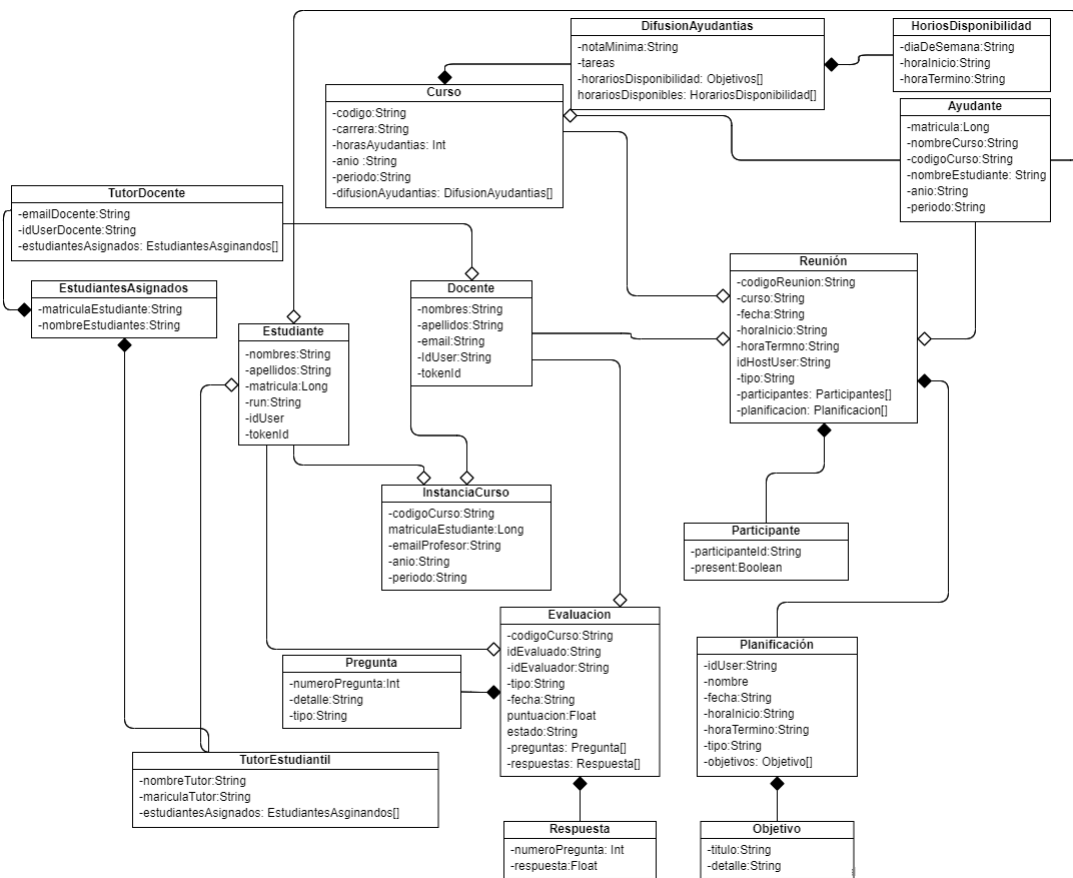


Figura 4.4: Diagrama de clases del sistema

En POO existen dos tipos de relaciones: dinámicas y estáticas. Las relaciones dinámicas, o también conocidas como composición (relación fuerte), simbolizada con un rombo llenado negro, mientras que las estáticas se le conoce como agregación (relación débil) y se simboliza con un rombo blanco. En la Figura 4.5 podemos observar que la clase Participante tiene una relación de composición con la clase Reunión ya que si la clase Reunión deja de existir también dejará de existir la clase Participante, ya que si no existe una reunión no habrá personas participado de esta, es decir es, una relación fuerte. Por otro lado tenemos la clase Ayudante la cual tiene un relación débil con la clase Curso, ya que si se elimina la clase Curso, el Ayudante puede seguir existiendo por si solo.

4.2.4. Base de datos

Cloud Firestore es una base de datos NoSQL alojada en la nube, la cual esta basada en el almacenamiento de datos utilizando documentos, los cuales contienen campos asignados con diferentes valores. Estos documentos son almacenados en colecciones, que son contenedores de documentos, lo que permite crear consultas expresivas, eficientes y flexibles. Un ejemplo de la representación de la estructura con que se rigen los documentos es la siguiente.

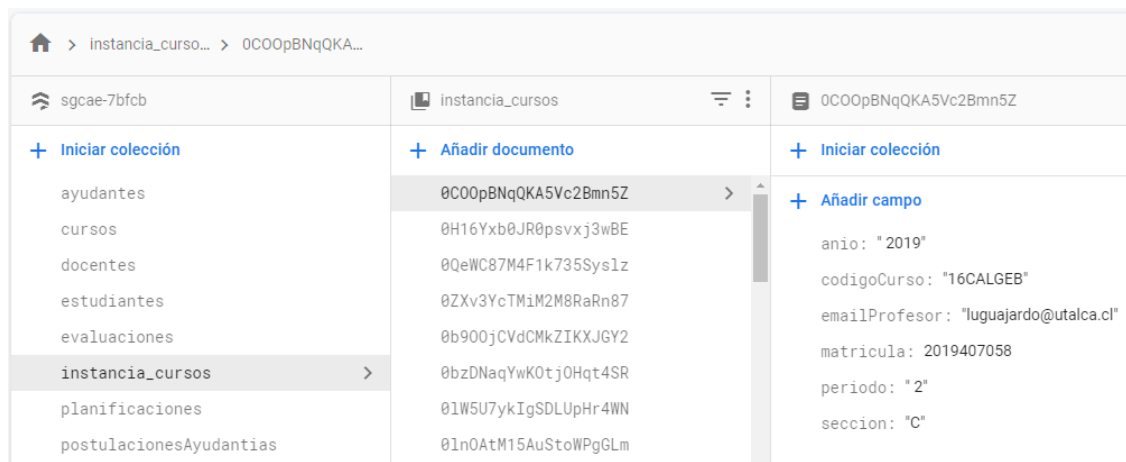


Figura 4.5: Modelo de datos en la base de datos Cloud Firestore

En la Figura 4.6, se puede observar en la parte izquierda la colección llamada instancia_cursos, la cual contiene varios documentos con id generados automáticamente por Cloud Firestore, si expandimos uno de estos documentos, nos encontramos con los campos anio, codigoCurso, emailProfesor, entre otros campos, como podemos ver en la parte derecha de la figura. Es esta la estructura que sigue el modelo de base de datos en Firebase.

4.2.5. Implementación de Firebase Cloud Firestore

Para implementar los servicios de Cloud Firestore en Angular se debe instalar las dependencias necesarias mediante el gestor de paquetes llamado NPM, como se muestra en la Figura 4.6

```
npm install firebase @angular/fire --save
```

Figura 4.6: Instalación de librerías de los servicios de Firebase en Angular

En la Figura 4.6 se muestra la inclusión de la librerías necesarias para el servicio Cloud Firestore, el cual nos permitirá agregar, leer, eliminar colecciones y/documentos en la base de datos NoSQL que nos proporciona Firebase.

Como se trabaja con documentos, se omitió el diagrama entidad relación, ya que no hay un equivalente para este tipo de base datos, pero si el más cercano es un diagrama de clase, como se muestra en la Figura 4.4.

4.2.6. Implementación de Firebase Authentication

Firebase Authentication es el servicio que nos provee Firebase para autenticar usuarios. Para el caso de Angular se debe instalar mediante el gestor de paquetes NPM. Como se muestra en Figura 4.6.

En la Figura 4.6 se ve el comando necesario para instalar la dependencias que contienen las librerías necesarias para hacer uso de los servicios de Firebase en Angular. Una vez instalado estas podemos utilizar los servicios de Firestore y Authentication, importando las dependencias correspondientes.

Una vez incluidos en el proyecto para la aplicación web, podemos hacer uso de la autenticación de usuario, permitiéndonos registrar usuarios desde este servicio, los cuales pueden ser visualizados a través de la consola web que nos provee Firebase.



The screenshot shows the Firebase Authentication console interface. At the top, there is a search bar with the placeholder text "Buscar por dirección de correo electrónico, número de teléfono o UID de usuario" and a blue "Agregar usuario" button. Below the search bar is a table with the following columns: "Identificador", "Proveedores", "Fecha de creación", "Fecha de acceso", and "UID del usuario". The table contains four rows of user data. At the bottom right, there is a pagination control showing "Filas por página: 50" and "1 - 4 of 4".

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID del usuario ↑
dmatus95@outlook.cl	✉	23 sep. 2020	18 oct. 2020	2f4vv1FMnVQhsV09ZKgl4Br143T2
rpavez@utalca.cl	✉	2 nov. 2020	12 nov. 2020	3K1anKaVh8R368XkQK7F178s3z02
dmatus15@alumnos.utalca.cl	✉	2 nov. 2020	12 nov. 2020	DwLiNb2i4Qb2vUSeUcQvSnYUfLG2
diegomatuscampos@gmail.co	✉	3 sep. 2020	8 oct. 2020	hQdzZqCieDbAtF2PvTviPw6KE7h2

Figura 4.7: Consola de Firebase - Servicio Authentication

La Figura 4.8 muestra la lista de usuarios que han sido registrados en el sistema, donde encontramos que la información de un usuario corresponde a un identificador, proveedor, fecha de creación, fecha de acceso y por último el id de usuario. Como ejemplo tenemos que el primer usuario tiene como identificador el email, luego su proveedor es correo electrónico, es decir, que se está autenticando mediante correo y contraseña, seguido de la fecha de creación del usuario, fecha de acceso y por último un id de usuario que el desarrollo de este proyecto es relevante para identificar cada usuario en el sistema.

4.2.7. Implementación del servicio Firebase Cloud Messaging

Este servicio nos provee la posibilidad de enviar y recibir notificaciones push, ya sea para una aplicación móvil y/o web. Para hacer uso del envío de notificaciones desde Angular se debe hacer a través de una petición POST a la URL que se muestra en la Figura 4.9

```
baseUrl: string = 'https://fcm.googleapis.com/fcm/send';
```

Figura 4.8: URL para envío de notificaciones push a través de la API de Firebase Cloud Messaging

En la Figura 4.9 se nos muestra la URL mediante la cual nos permite enviar notificaciones push a uno o varios dispositivos. Esto utilizando el formato de archivo JSON, en donde se incluye token del servidor, el cual es otorgado por el servicio, token id del dispositivo que es un identificador único del dispositivo, y luego está el contenido de la notificación. Esto es enviado a través de una petición POST.

```
let mensaje = {
  to: docente.tokenId,
  notification: {
    title: 'Tutorías Docentes',
    body:
      docente.nombre +
      docente.apellidos +
      ' se te han asignado estudiantes',
  },
}
```

Figura 4.9: Estructura JSON para envío de notificaciones push mediante el servicio FCM

La Figura 4.10 muestra como se construye la estructura JSON necesaria para enviar una notificación. Esta contiene una serie de campos mínimos necesarios, los cuales se describen a continuación.

- `to`: este campo asigna el token identificador del dispositivo destinatario, es decir, enviaría una notificación al token que corresponde a ese dispositivo en específico.
- `notificación`: este es campo contiene el contenido que llevara la notificación.
- `tittle`: define el título que es mostrado en la nofitificacion.
- `body`: define el cuerpo del mensaje que es mostrado en la notificación.

4.3. Ejecución de las iteraciones

La siguiente sección describe como se aborda la implementación de cada una de las iteraciones planificadas en el proyecto, las cuales fueron especificadas en el capítulo tres sección 3.3.

4.3.1. Iteración 1 - Carga masiva de datos de estudiantes, cursos y docentes en aplicación web

En esta primera iteración la funcionalidad desarrollada para aplicación web fue la carga masiva de datos, la cual tiene la finalidad de alimentar la base de datos con la información de estudiantes, cursos y docentes. Esto es realizado por medio

de la lectura de un archivo excel, creando los componentes cursos, estudiantes y docentes dentro del proyecto Angular de la aplicación. Los cuales contienen las vistas, controladores dentro de cada componente. Para implementar la carga masiva de datos se hace uso del elemento Input de tipo file incluido en HTML5, el cual nos permite seleccionar un archivo de nuestro dispositivo, permitiéndonos obtener el nombre del archivo y su contenido. Estos son enviados al controlador o también conocido en Angular como component, dentro de este es convertido en binario y luego en formato JSON, para luego ser parseado y almacenado en el servicio Cloud Firestore bajo las colecciones docentes, estudiantes, cursos y instancia_cursos. El resultado de esta funcionalidad se muestra a continuación.

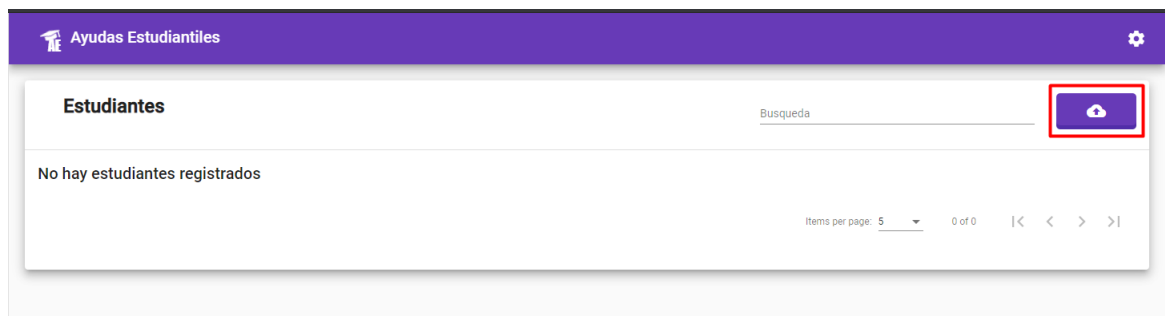


Figura 4.10: Input de tipo file para selección del archivo Excel de carga de datos masiva

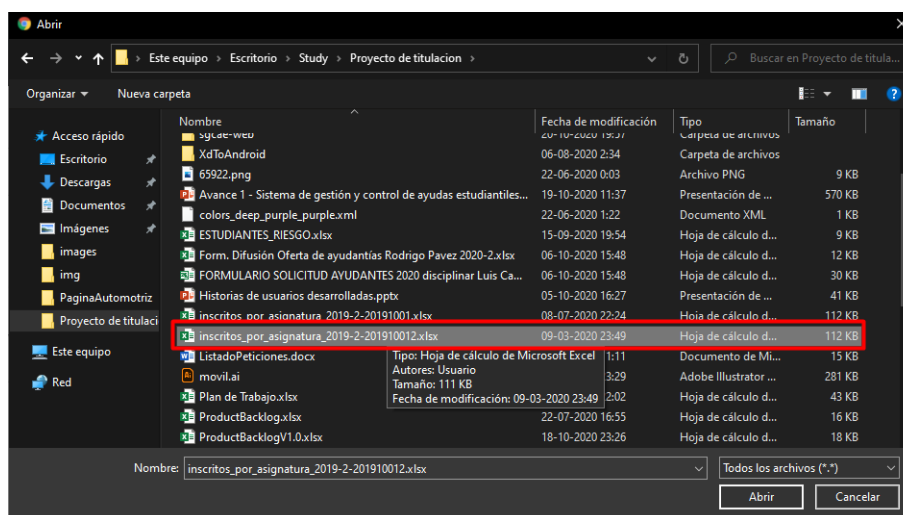


Figura 4.11: Selección del archivo Excel de carga de datos masiva

En las Figuras 4.11 y 4.12 podemos observar como se selecciona el archivo excel llamado `inscritos_por_asignaturas`, el cual contiene los datos de los estudiantes y los cursos inscritos de un semestre. Una vez que se ha seleccionado el archivo es procesado, parseado y almacenado en la base de datos, como se ve a continuación.

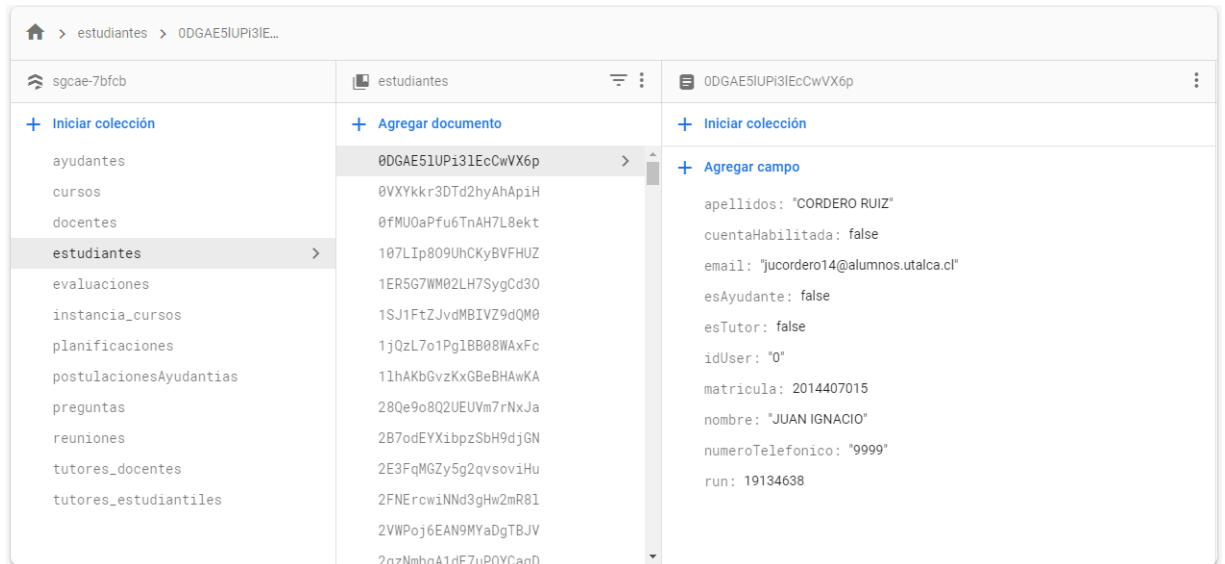
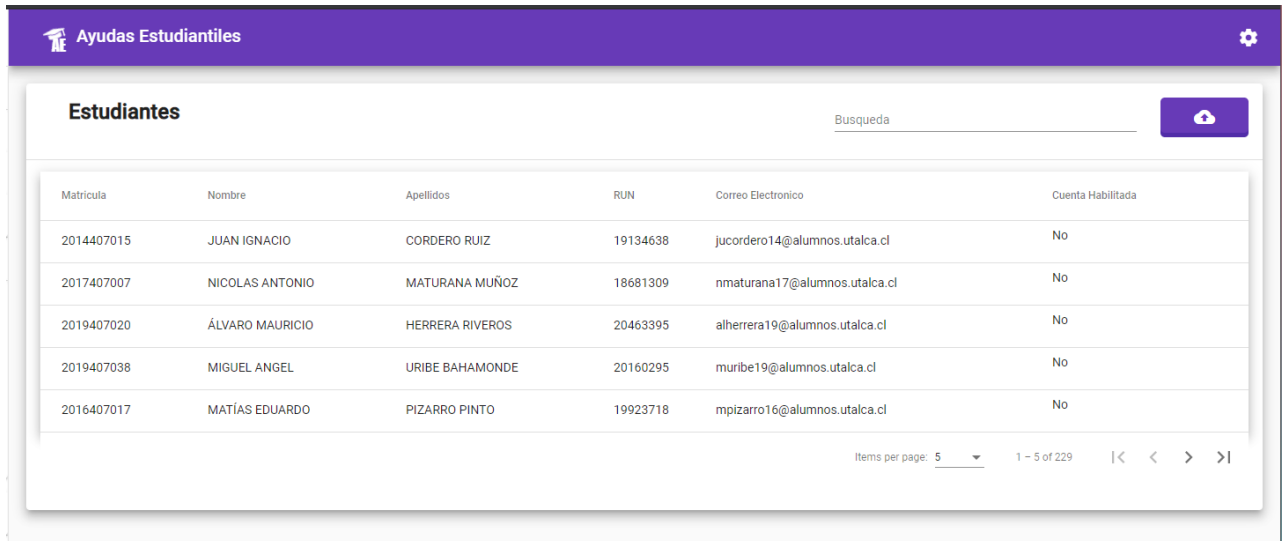


Figura 4.12: Almacenamiento de datos en Cloud Firestore de Estudiantes

La Figura 4.13 nos muestra como se almacenan los datos obtenidos desde el archivo Excel, lo cual crea las colecciones de documentos `estudiantes`, `cursos`, `docentes` y `instancia_cursos`, con sus respectivos campos cada uno.



Matricula	Nombre	Apellidos	RUN	Correo Electronico	Cuenta Habilitada
2014407015	JUAN IGNACIO	CORDERO RUIZ	19134638	juccordero14@alumnos.utalca.cl	No
2017407007	NICOLAS ANTONIO	MATURANA MUÑOZ	18681309	nmaturana17@alumnos.utalca.cl	No
2019407020	ÁLVARO MAURICIO	HERRERA RIVEROS	20463395	alherrera19@alumnos.utalca.cl	No
2019407038	MIGUEL ANGEL	URIBE BAHAMONDE	20160295	muribe19@alumnos.utalca.cl	No
2016407017	MATÍAS EDUARDO	PIZARRO PINTO	19923718	mpizarro16@alumnos.utalca.cl	No

Figura 4.13: Visualización de estudiantes cargados en la aplicación web

Por último en la Figura 4.14, se puede apreciar como son visualizados los datos en la plataforma web, en forma de lista en una tabla, además de incorporar un input, el cual permite filtrar por el nombre ya sea del curso, docente o estudiante dependiendo de la vista en que se encuentre el usuario.

4.3.2. Iteración 2 - Autenticación de usuarios y visualización de ayudantías por cursos en la aplicación web

En la iteración dos se implementan las funcionalidades de autenticación de usuarios, visualización de datos de ayudantías por cursos. En la primera funcionalidad se crea el componente login, el cual contiene la vista de inicio, el controlador donde se envían los datos al modelo para que valide los datos de usuarios a través del servicio Authentication de Firebase. Dentro de la vista encontramos un formulario, con dos input uno de tipo email y otro de tipo password, donde el usuario debe ingresar su correo y contraseña. Estos datos son obtenidos en el component y enviados al service llamado auth para que los procese y valide en Firebase Authentication, una vez validados, el usuario es redirigido al inicio de la aplicación. Para el proyecto solamente la dirección de escuela y asistente tendrá acceso a esta plataforma web.

Luego la segunda funcionalidad desarrollada es mostrar los datos de ayudantías por cada curso, para este cometido se divide la vista de ayudantías en pestañas o

llamadas tabs, teniendo una pestaña dedicada para los estudiantes inscritos de un curso, otra para el ayudante de ese curso y por último uno el docente asignado a ese curso.

4.3.3. Iteración 5 - Asignar estudiantes a docentes tutores

La tarea que se realiza en esta iteración fue la asignación de estudiantes a tutores docentes, todo esto para facilitar y reducir el tiempo que se usa en este cometido. Para la implementación de la asignación de estudiantes a docentes tutores, se implementa el CDK Drag&Drop de Angular Material, el cual permite que se arrastren ítems a un contenedor para facilitar la asignación a la dirección de escuela. Esta se puede ver en la Figura 4.15.

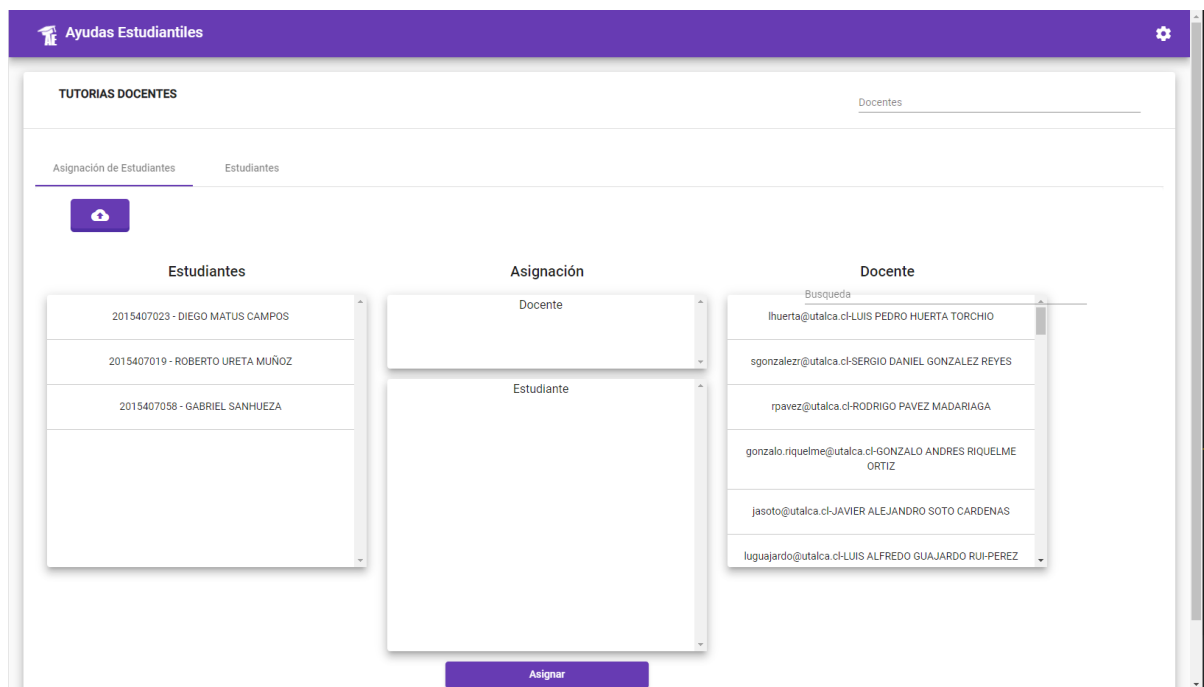


Figura 4.14: Vista de asignación de estudiantes a docentes tutores

La Figura 4.15 muestra los contenedores implementados, los cuales podemos arrastrar items a los dos contenedores centrales, es decir, el contenedor estudiantes, donde puede solamente ser arrastrado un estudiante al contenedor asignación estudiantes y del contenedor docentes solo puede ser arrastrado al contenedor asignación docentes, una vez que se realiza la asignación se debe presionar el botón

asignar el cual desencadena el envío de una notificación push, y almacenamiento del tutor y los estudiantes que tiene asignado. Como se muestra a continuación.

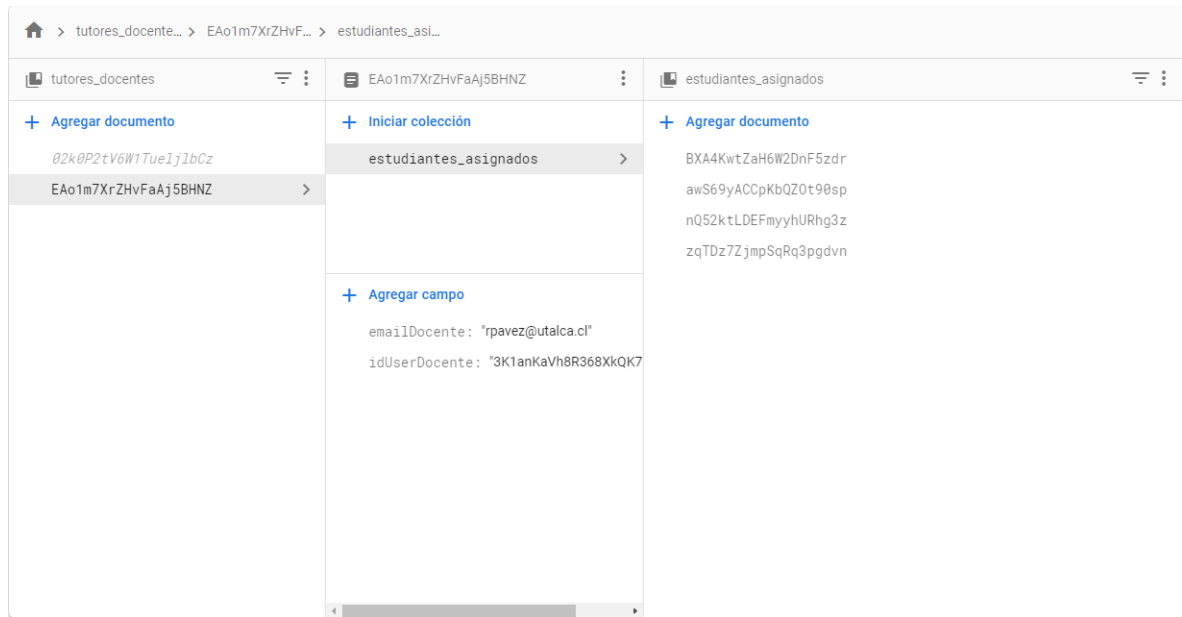


Figura 4.15: Almacenamiento de tutor docente en la base de datos

La Figura 4.34 muestra como se almacena el tutor docente en la base de datos, tras haber sido realizada la asignación en la plataforma web. En primera instancia se crea una colección llamada `tutores_docentes` la cual tiene como campos el `emailDocente` y `idUserDocente`, el cual es identificador de usuario del docente. A su vez el documento contiene una subcolección llamada `estudiantes_asignados` la cual contiene la matrícula y nombre del estudiante. Al haber finalizado correctamente la creación del docente tutor, se genera el envío de una notificación push, notificándole que se le han asignado estudiantes, como se muestra en la Figura 4.35.

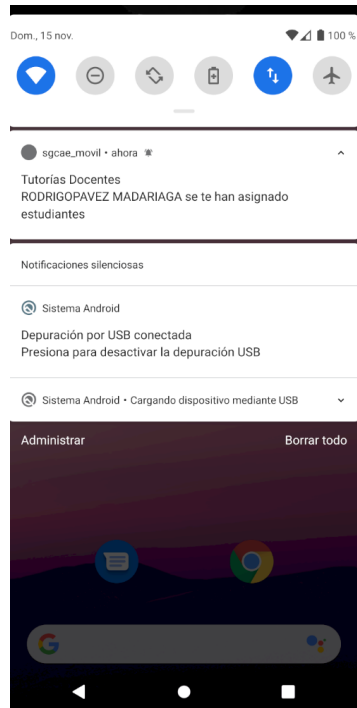
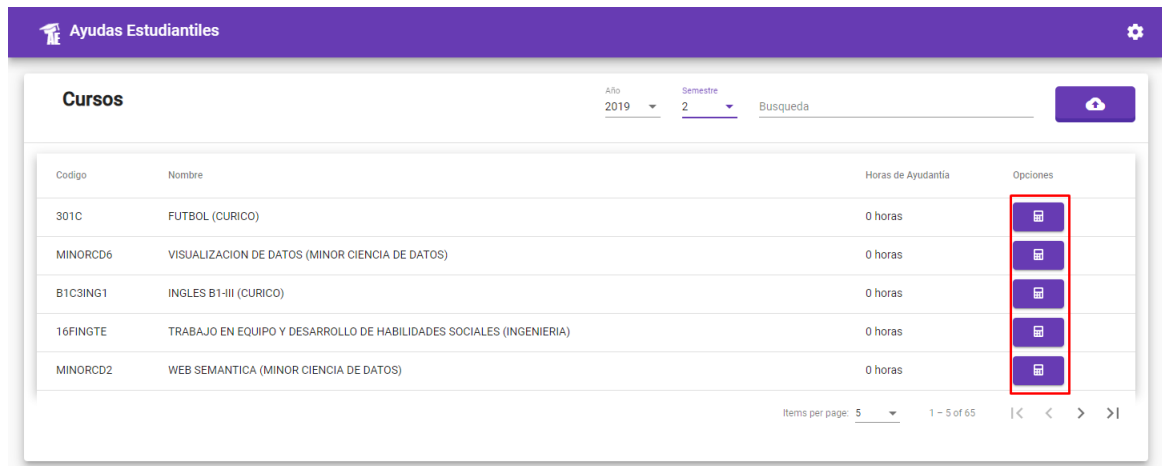


Figura 4.16: Notificación enviada al tutor docente

4.3.4. Iteración 6 - Cálculo de horas de ayudantías y resumen de ayudantes del semestre

En esta iteración se trabajó con las funcionalidades de cálculo de horas de ayudantías, el cual consiste en calcular el número de horas de ayudantías en base a la cantidad de estudiantes inscritos en un curso, además de las horas de ayudantía y/o laboratorio que tiene un curso. Por otra parte, tenemos la funcionalidad de mostrar el resumen de ayudantes del semestre una vez que se han seleccionado los ayudantes, se muestra la información de los ayudantes del semestre. Para la primera funcionalidad se implementa un botón en la vista de cursos, el cual despliega un modal o diálogo, que nos permite ingresar la cantidad de horas de laboratorio y/o ayudantías que tiene el curso. Posteriormente a esto si se confirma la operación se realiza el cálculo automático de horas propuestas en base a los alumnos inscritos. Esto se muestra en las siguientes figuras.

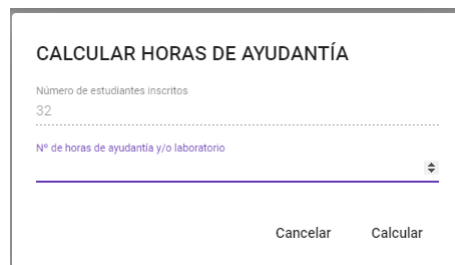


The screenshot shows a web interface for 'Ayudas Estudiantiles'. At the top, there is a purple header with the logo and a settings gear. Below the header, there is a section for 'Cursos' with filters for 'Año' (2019) and 'Semestre' (2). A search bar is also present. The main content is a table with the following data:

Codigo	Nombre	Horas de Ayudantia	Opciones
301C	FUTBOL (CURICO)	0 horas	[Icon]
MINORCD6	VISUALIZACION DE DATOS (MINOR CIENCIA DE DATOS)	0 horas	[Icon]
B1C3ING1	INGLES B1-III (CURICO)	0 horas	[Icon]
16FINGTE	TRABAJO EN EQUIPO Y DESARROLLO DE HABILIDADES SOCIALES (INGENIERIA)	0 horas	[Icon]
MINORCD2	WEB SEMANTICA (MINOR CIENCIA DE DATOS)	0 horas	[Icon]

A red box highlights the 'Opciones' column, which contains a document icon for each row. At the bottom right of the table, there is a pagination control showing 'Items per page: 5' and '1 - 5 of 65'.

Figura 4.17: Implementación de calcular propuesta de horas de ayudantías



The dialog box is titled 'CALCULAR HORAS DE AYUDANTÍA'. It contains two input fields: 'Número de estudiantes inscritos' with the value '32' and 'Nº de horas de ayudantía y/o laboratorio' with a dropdown arrow. At the bottom, there are two buttons: 'Cancelar' and 'Calcular'.

Figura 4.18: Diálogo para calcular la propuesta de horas de ayudantías

Las Figuras 4.18 y 4.19 se muestran como se implementa la funcionalidad de calcular horas de ayudantías, donde podemos apreciar que cada curso tiene un botón para dicho cálculo. Una vez que se a presionado el botón este despliega un diálogo el cual contiene dos inputs, uno que calcula la cantidad de estudiante inscritos de un curso. Luego, tenemos otro input el cual nos permite ingresar el número de horas que se imparten el curso de ayudantías y/o laboratorios. Un vez que se ha calculado, enviará una notificación al docente a cargo del curso, indicándole que se ha calculado su propuesta de horas de ayudantías, como se puede apreciar a continuación en la Figura 4.20.

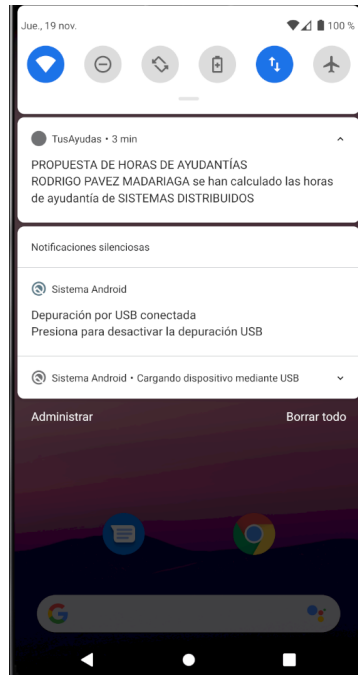


Figura 4.19: Notificación de propuesta de horas de ayudantías de un curso

Luego se realiza la funcionalidad de mostrar un resumen de ayudantes del semestre, el cual consiste en mostrar la información relevante del ayudante. Esto se muestra en la Figura 4.21

The screenshot shows a web application interface with a purple header 'Ayudas Estudiantiles'. The main content is titled 'RESUMEN DE AYUDANTÍAS' and includes a 'Descargar' button. There are two tabs: 'Resumen de ayudantes' (selected) and 'Información de montos de ayudantías'. A search bar labeled 'Busqueda' is present. Below is a table with the following data:

Módulo	Nota Aprobación Módulo	Nombres	Matricula	RUT	Número de horas mensuales	Correo	Profesor
TECNOLOGÍAS WEB (MINOR CIENCIAS DE LA COMPUTACION)	6.5	DIEGO ANTONIO MATUS CAMPOS	2015407023	19007996	8 hrs	NO	NO
SISTEMAS DISTRIBUIDOS	6.5	DIEGO ANTONIO MATUS CAMPOS	2015407023	19007996	16 hrs	NO	NO

At the bottom right, there is a pagination control: 'Items per page: 5', '0 of 0', and navigation arrows.

Figura 4.20: Resumen de ayudantes del semestre

En la Figura 4.22 podemos observar la información que contiene el resumen de ayudantes del semestre, los cuales son, nombre del curso, nota de aprobación del

curso, nombre del ayudante, matricula, RUT, numero de horas de ayudantías, correo del estudiante y el profesor a cargo del curso.

4.3.5. Iteración 8 - Mostrar información de ayudantías, tutorías y generación y descarga de información del proceso en formato XLS

Dentro de la iteración ocho se implementa la visualización de la información generada por la aplicación móvil de ayudantías, tutorías docentes y tutorías estudiantiles, además se implementa la generación y descarga de información resumen del proceso de ayudantías y tutorías, la cual descarga un archivo excel con los datos correspondiente al apartado seleccionado.

4.4. Pruebas

En esta sección se expondrá el plan de pruebas que se hicieron para verificar el correcto funcionamiento de la plataforma web. Cabe destacar que se hicieron pruebas de caja negra y se muestran solo las pruebas de las funcionalidades más importantes. Las evidencias de estas pruebas están disponibles en el Anexo F.

Cuadro 4.1: Prueba de carga masiva de datos de estudiantes, cursos y docentes.

Prueba de carga masiva de datos	
Propósito	Verificar que se carguen correctamente los datos de estudiantes, cursos y docentes en la plataforma web
Prerrequisitos	El usuario debe haber ingresado al sistema.
Entradas	Archivo Excel con la nomina de estudiantes inscritos del semestre.
Salidas	Mostar los datos en la plataforma web y almacenarlos en base de datos.
Resultado de la prueba	Medianamente Correcto.

Cuadro 4.2: Prueba de asignación de estudiantes a un docente tutor

Prueba de asignación de estudiantes a docentes tutores	
Propósito	Verificar la correcta asignación de estudiantes a un docente tutor
Prerrequisitos	El usuario debe haber ingresado al sistema.
Entradas	Archivo Excel con la lista de estudiantes en riesgo derivados a tutorías docentes
Salidas	Mostrar mensaje de éxito y almacenar los datos en base de datos.
Resultado de la prueba	Correcto.

Cuadro 4.3: Prueba de asignación de estudiantes a un docente estudiantiles

Prueba de asignación de estudiantes a tutores estudiantiles	
Proposito	Verificar la correcta asignación de estudiantes a un tutor estudiantil
Prerrequisitos	El usuario debe haber ingresado al sistema.
Entradas	Archivo excel con la lista de estudiantes en riesgo derivados a tutorías estudiantiles, y archivo excel con la lista de tutores estudiantiles
Salidas	Mostrar mensaje de éxito y almacenar los datos en base de datos.
Resultado de la prueba	Correcto.

Cuadro 4.4: Prueba de cálculo de horas de ayudantías.

Prueba de calculo de propuesta de horas de ayudantías.	
Proposito	Verificar la correcta actualización de las horas ayudantías propuestas para un curso.
Prerrequisitos	El usuario debe ingresado a la plataforma web, cargado los datos de estudiantes, cursos y docentes.
Entradas	Seleccionar un curso, número de horas de ayudantías y/o laboratorios, número de estudiantes inscritos.
Salidas	Actualización de horas de ayudantías en el registro del curso en la base de datos, envió de notificación al docente el curso.
Resultado de la prueba	Correcto.

Cuadro 4.5: Prueba de generación de información de resumen del proceso ayudantías y tutorías.

Prueba de generación de información de resumen del proceso ayudantías y tutorías.	
Proposito	Verificar la correcta generación de información de ayudantías y tutorías, además de la correcta generación de archivos excel con la información.
Prerrequisitos	El usuario debe haber ingresado al sistema.
Entradas	Seleccionar el apartado, seleccionar el año, semestre, curso y descarga de la información.
Salidas	Creación del archivo excel y descarga de este.
Resultado de la prueba	Mediamente Correcto.

5. Desarrollo aplicación móvil

En este capítulo se presenta como se aplica la metodología en el desarrollo de la aplicación móvil, detallando las iteraciones que se ejecutaron para construirla. En primera instancia se muestra el flujo de trabajo en que se rige la aplicación móvil, a través de diagramas de actividades. Para posteriormente exponer el diseño de la aplicación mostrando la arquitectura física y lógica utilizado. A continuación de esto se presenta la ejecución de cada una de las iteraciones y las funcionalidades que se implementaron en cada una de ellas, para finalizar mostrando las pruebas de caja negra que se llevaron a cabo.

5.1. Flujo de trabajo de la aplicación móvil

En esta sección se representara el flujo que sigue la aplicación móvil para realizar las actividades que corresponde al proceso de ayudantías, tutorías docentes y tutorías estudiantiles.

5.1.1. Flujo para el proceso de ayudantías

El flujo que siguen las actividades para interactuar con la aplicación móvil son las siguientes:

1. **Activación de cuenta de usuario para los estudiantes:** este es la primera actividad que deben hacer los estudiantes para poder acceder a la aplicación, para este cometido deben ingresar sus matrícula y rut sin dígito verificador, luego serán validados para seguir con el siguiente paso que corresponde a la verificación de correo electrónico. Este flujo esta representado en un diagrama de actividades, el cual está disponible en el Anexo B.1.

2. **Postulación a una oferta de ayudantía:** en esta actividad los estudiantes pueden postular a una oferta de ayudantía, para esto deben seleccionar una oferta, rellenar el formulario de postulación y confirmar su postulación. Para cerrar el proceso se envía una notificación al docente de que han postulado a una oferta de su curso. El diagrama de este subproceso está disponible en el Anexo B.2
3. **Selección de ayudantes:** en esta actividad están involucrados los docentes, el cual debe seleccionar una postulación, a continuación en el detalle de la postulación puede aceptarla o rechazarla. Al igual que los subprocesos anteriores el diagrama de actividades correspondiente está disponible en el Anexo B.3.
4. **Creación de reuniones de ayudantías:** una vez que se han seleccionado a los estudiantes que serán ayudantes del semestre, estos pueden crear reuniones de ayudantías con el fin de registrar la asistencia de los participantes. Para eso debe seleccionar el curso en el que quieren realizar la reunión, el horario y una observación. El diagrama de actividades correspondiente se encuentra en el Anexo B.5.
5. **Registro de asistencia:** luego de haber creado una reunión de ayudantía el ayudante puede generar un código único para la reunión con el fin de que los estudiantes puedan ingresar esta código y marcar asistencia para esa reunión. Esto es representado en el diagrama de actividades que esta disponible en el Anexo B.6.
6. **Evaluar a un ayudante:** para esto docentes y estudiantes, deben seleccionar el ayudante en el apartado correspondiente, contestar el cuestionario de evaluación y para finalizar dejar un comentario para realizar dicha evaluación. Este subproceso se encuentra detallado en el Anexo B.4.

5.1.2. Flujo para el proceso de tutorías docentes en la aplicación

El flujo que se realiza para este proceso donde interactúan el docente y los estudiantes es el siguiente:

1. **Agenda de reuniones con los estudiantes:** el docente puede agendar reuniones con los estudiantes, para esto debe seleccionar al estudiantes con

el cual quiere reunirse, el horario y una observación que el estime pertinente. Un vez creada la reunión le llegará una notificación al estudiante de la reunión que ha solicitado el docente, la cual puede ser rechazada y aceptada. este subproceso se detalla en el Anexo B.7.

2. **Registrar anotaciones de sesiones:** el docente puede registrar anotaciones en la sesiones, ingresando un título y el contenido de la anotación. Este subproceso esta representado en un diagrama de actividades en el Anexo B.8

5.1.3. Flujo para el proceso de tutorías estudiantiles en la aplicación

El flujo que se realiza para este proceso donde interactúan el tutor estudiantil y los estudiantes asignados a este, es el siguiente:

1. **Planificación de tutorías estudiantiles:** donde los tutores pueden agregar una planificación ingresando el horario, el título, objetivos, actividades y alguna observación, lo cual genera la planificación y reunión correspondiente. Este proceso es detallado en el Anexo B.9
2. **Registrar asistencia de una tutoría estudiantil:** este subproceso sigue el mismo flujo que el registro de asistencia de ayudantías. Donde el tutor genera un código único para la sesión de tutoría, el cual es entregado a los estudiantes, para que ingresen ese código y marquen asistencia. El diagrama correspondiente está disponible en el Anexo B.10.

5.2. Diseño de la aplicación móvil

En esta sección se describe el diseño que se realiza para la aplicación móvil, mostrando el diseño de la arquitectura física, lógica. Posteriormente se expone el diagrama de clases, el cual es subdividido por roles de usuario.

5.2.1. Arquitectura Física

La arquitectura física intenta modelar la interacción y comunicación entre los componentes físicos que están involucrados en el proyecto, siendo estos los clientes y servidores como servidor web y servidor de base de datos, entre otros. Para este

proyecto se eligió una arquitectura Cliente-Servidor, donde los clientes son dispositivos móviles que serán usados por los estudiantes y docentes. Esto este representado a continuación en la Figura 5.1.

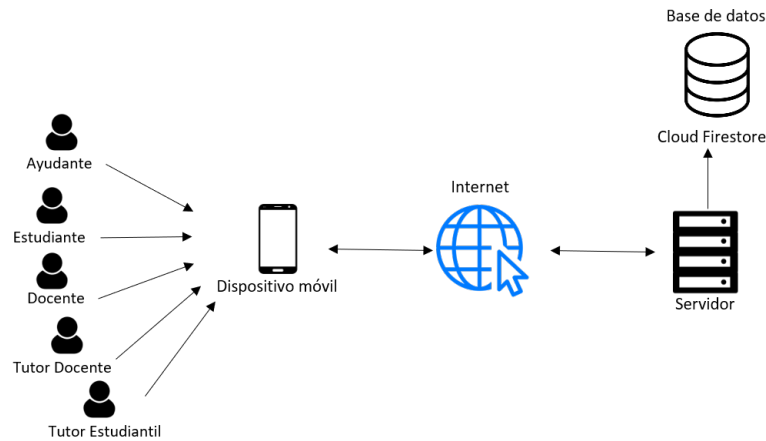


Figura 5.1: Arquitectura física - Cliente-Servidor aplicación móvil

5.2.2. Arquitectura lógica

Una arquitectura física nos provee la información de como se interrelacionan los distintos módulos y/o submódulos del proyecto, al igual que en la arquitectura lógica de la aplicación web primero se muestra la arquitectura lógica a alto nivel, para luego entrar en detalle con los módulos y submódulos de la aplicación móvil, subdivididos en vista, controlador y modelo, esto se aprecia en la Figura 5.2

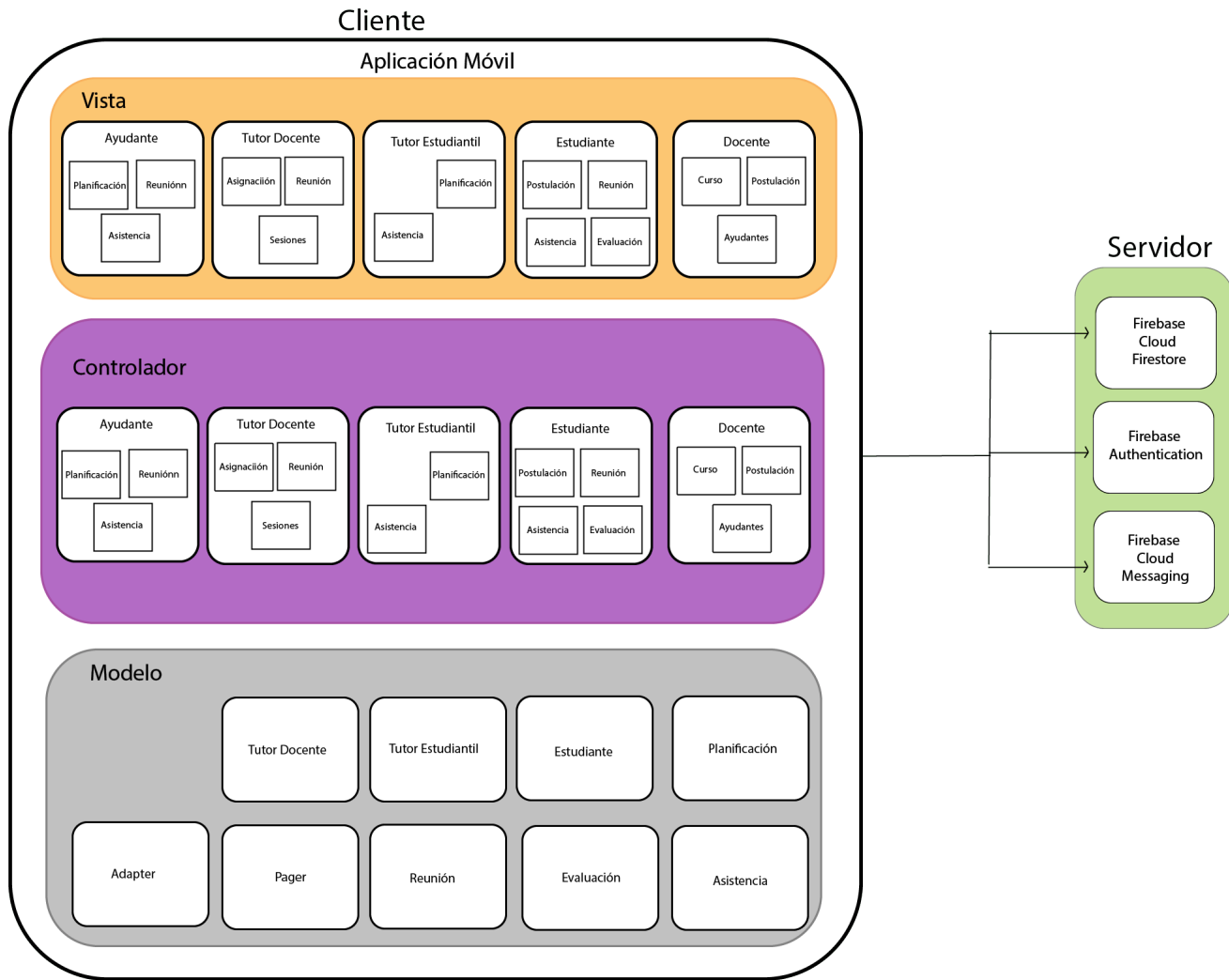


Figura 5.2: Arquitectura lógica - MVC aplicación móvil

La Figura 5.2 muestra la interacciones de los subsistemas de la aplicación móvil que al igual que la aplicación web utilizado el patrón MVC. Dividiendo las vistas por cada uno de los usuarios que interactúa en la aplicación, como el ayudante el cual tiene las vista de planificación, reunión y asistencia. El docente que tiene interacción con las vista de curso, postulación y ayudantes, además de los tutores docentes y estudiantiles que utilizan las vista de asignaciones, reunión, sesiones para el primero y el segundo planificación y asistencia. Por último las vistas con que interactúan los estudiantes, teniendo la postulación , reunión, asistencia y evaluación. También tenemos los controladores, uno por cada vista antes mencionada y para

finalizar tenemos los modelos los cuales contienen la lógica necesaria para procesar los datos por los controladores y ser mostrados por las vistas en la aplicación móvil. Adicionalmente la aplicación hace peticiones a Firebase en conjunto con la aplicación web las cuales intercambian información a través de este servicio en la nube.

5.2.3. Diagrama de clases

Al igual que la aplicación web, la aplicación móvil esta construida utilizando el paradigma de programación orientado a objetos (POO), utilizando el lenguaje de programación Kotlin que es utilizado en el desarrollo de aplicaciones para el sistema operativo Android. Por este motivo la mejor representación para este tipo de paradigma es diseñar un diagrama de clases. Las partes que componen un diagrama de clases son: clases, atributos, operaciones y relaciones entre clases. Dentro de las relaciones entre clases encontramos diferentes tipos como dependencia, composición y agregación. Donde la relación de dependencia hace referencia que una clase A usa a la clase B, si la clase B es modificada también se vera afectada la clase A. Luego tenemos la composición y agregación que se explicaron con anterioridad en el capítulo cuatro en la sección 4.1. Para el caso del desarrollo de la aplicación móvil se diseño un diagrama de clases para cada uno de roles existentes siendo los roles de estudiante, ayudante, tutor docente, docente y tutor estudiantil, estos son presentados a continuación.

Diagrama de clases para el rol de estudiante

El rol de estudiante contempla los módulos de reuniones, asistencia y evaluaciones los cuales están subdivididos en clases que se destinadas para la implementación de actividades y fragmentos, clases adaptadoras y clases de modelo de datos. Este diagrama se separó en diferentes partes, las cuales serán mostradas con más detalle en las figuras siguientes.

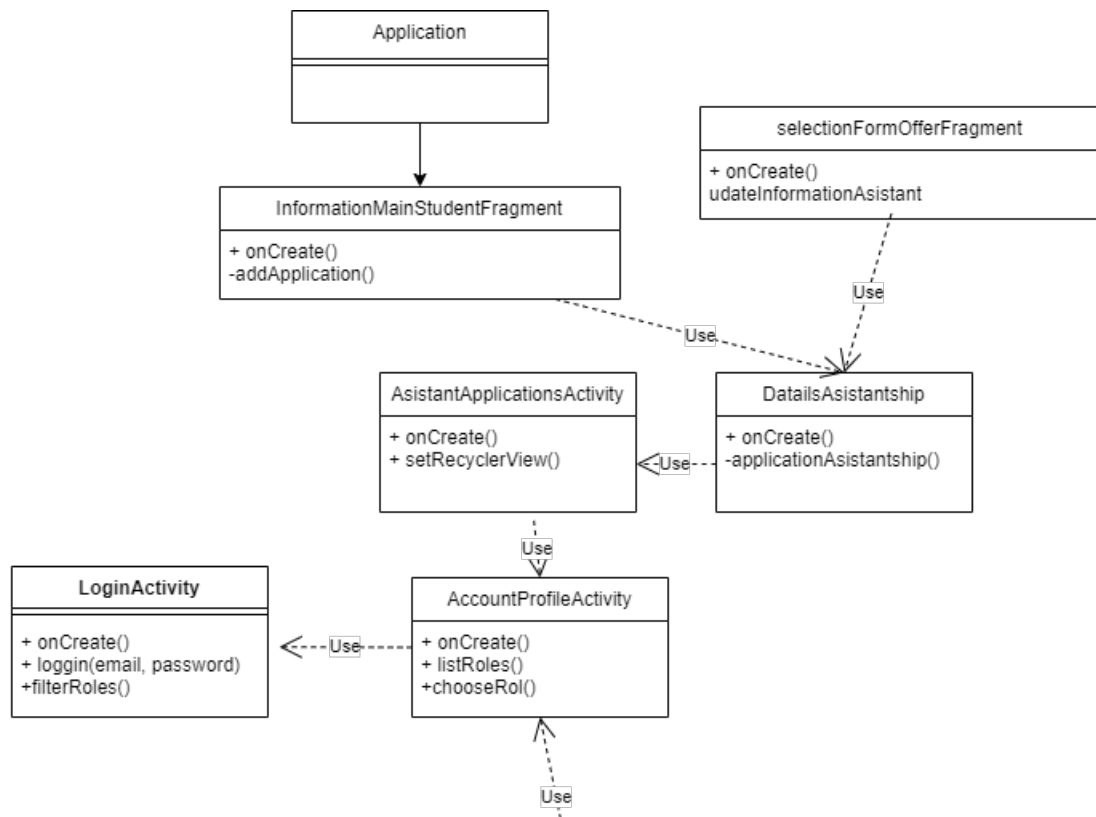


Figura 5.3: Parte del diagrama de clases del rol de estudiante para ingresar a la aplicación y realizar una postulación a ayudantías

La Figura 5.3 representa la parte número 1 del diagrama de clases del rol de estudiante el cual contempla, las clases necesarias para iniciar sesión, seleccionar el perfil a utilizar y la postulación a ayudantías. Donde encontramos varias relaciones de dependencias, como por ejemplo la clases AccountProfileActivity es usada por la clase LoginActivity y a su vez la clases AsistantApplicationsActivity usa la clase AccountPorofileActivity. Otro ejemplo que tenemos es que la clase DetailsAsistantshipFragment usa la clase AsistantApplicationActivity, ya que si se genera un cambio dentro de la actividad afectará el funcionamiento del fragmento.

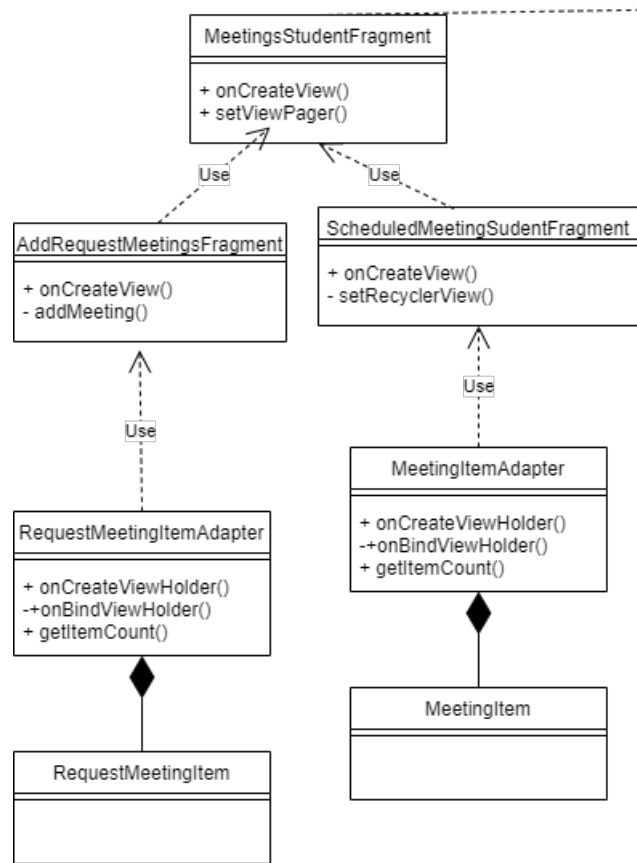


Figura 5.4: Parte del diagrama de clases del rol de estudiante para visualizar las reuniones de ayudantías y tutorías

En la Figura 5.4 encontramos la segunda parte del diagrama del rol de estudiantes, el cual contempla las clases asociadas al apartado de reuniones del estudiante, donde tenemos varias relaciones de dependencia como la clase `MeetingsStudentFragment` el cual usa a la clase `MainMenuNavigationStudentActivity`. También podemos observar relaciones de composición entre la clase `RequestMeetingItem` y `RequestMeetingItemAdapter`, ya que si deja de existir el adaptador que esta siendo usado por el fragmento, los items que están siendo utilizados dejan de existir, es decir los `RequestMeetingItem` no pueden existir por si solos, otro ejemplo de esto es la relación entre `MeetingItem` y `MeetingItemAdapter`.

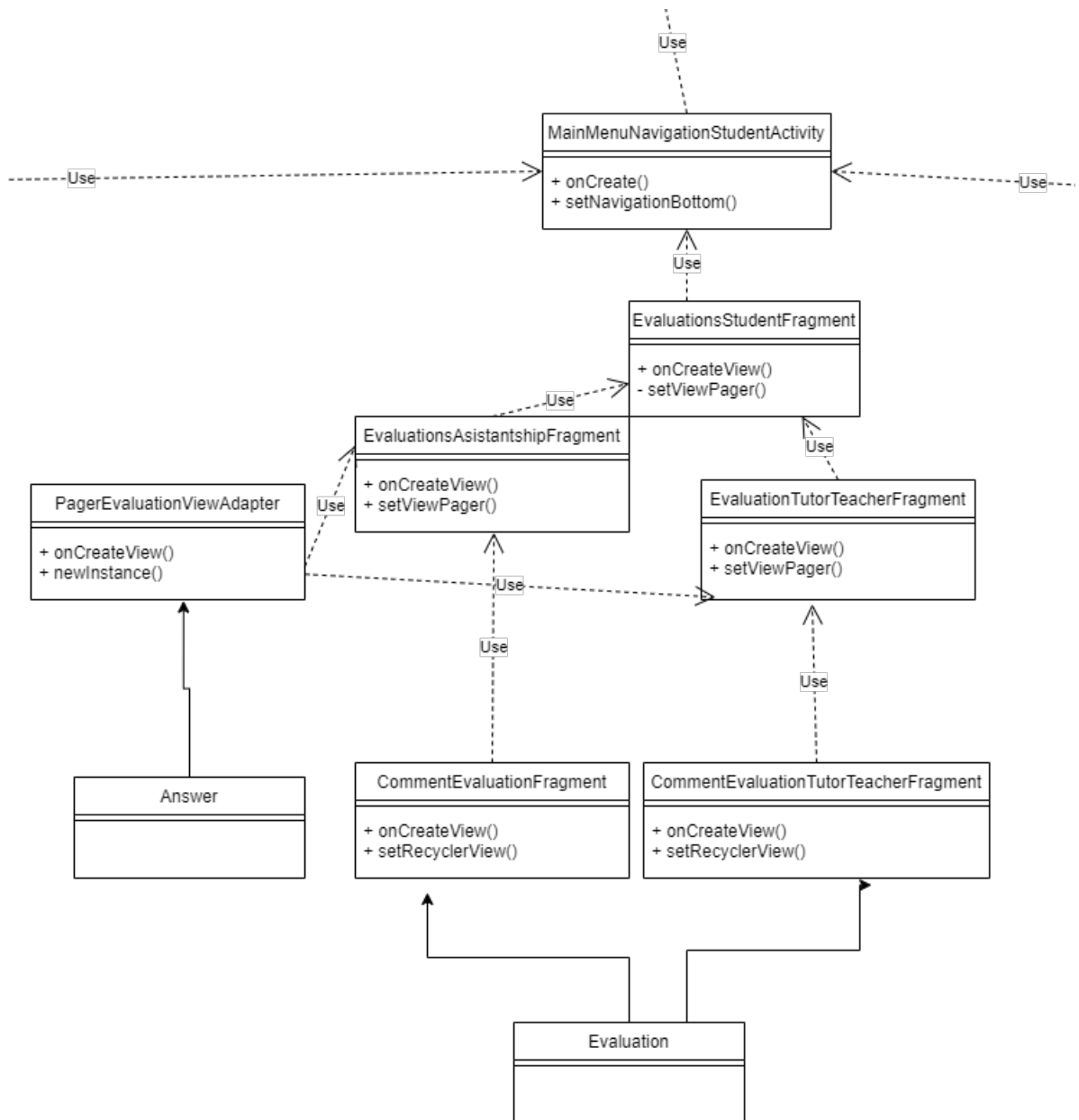


Figura 5.5: Parte del diagrama de clases del rol de estudiante para las evaluaciones de ayudantes y tutor docente

La Figura 5.5 representa la parte tres del diagrama de clases del rol estudiante, el cual contienen las clases destinadas para el apartado de evaluación de ayudantes y tutores estudiante, en este nos encontramos con la clase `MainMenuNavigationStudentActivity`, la cual se relaciona con la clase `EvaluationStudentFragment`, la cual

esa uso de esta. También nos encontramos con la relación entre Answer y PageEvaluationViewAdapter, la cual tiene una relación de asociación ya que PageEvaluationViewAdppter usa la clase Answer para almacenar datos en base de datos, al igual que las clases CommentEvaluationFragment y CommentEvaluationTutorTeacherFragment, los cuales usan la clase Evaluation.

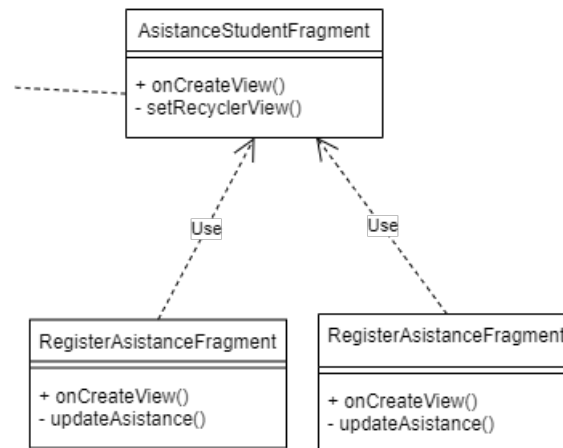


Figura 5.6: Parte del diagrama de clases del rol de estudiante para el registro de asistencia para ayudantías y tutorías estudiantiles

Por último en la Figura 5.6 se muestra la parte del diagrama que esta asociada al registro de asistencia por parte del estudiante.

Diagrama de clases para el rol de ayudante

El rol de ayudante contempla los módulos de reuniones de ayudantías, registro de asistencia, y planificación de ayudantías al igual que el diagrama anterior es subdivido en clases destinadas a actividades como la actividad MainMenuNavigationActivity, fragmentos como el fragmento MeetingsFragment, PlanificationFragment, AsistanceFragment, adaptadores siendo MeetingAdapter, PPlanificationItemAdapter, AsistantshipItemAdapter y modelos MeetingItem, Planning, PlanificationItem y ItemAsisstantship, los cuales se encuentran en el Anexo C.1 y Anexo C.2.

Diagrama de clases para el rol de docente

Para el rol de docente los módulos utilizados son cursos, postulaciones a ayudantías, ayudantes y evaluación, de la misma forma que los diagramas anteriores las clases se dividen en actividades como `MainMenuNavigationTeacherToAsistantActivity`, fragmentos donde encontramos el frgamento `CoursesFragment`, `ApplicationsFragment` y `AsistansFragment`, adaptadores como `MeetingItemAdppter`, `ItemOfferAsistantshipAdapter`, `PagerEvaluationTeacherViewAdapter` y por último modelos como `DifusionAsistanship`, `AvailableHpooursMeetingItem`, `ItemOfferAsistantship` y `EvaluationTeacher`, los cuales están dentro del Anexo C.3, Anexo C.4 y Anexo C.5.

Diagrama de clases para el rol de tutor docente

El rol de tutor docente involucra los módulos de asignaciones de estudiantes, reuniones, anotaciones de sesiones y reporte de evolución al estudiante, estos son representados en un diargama de clases el cual contine la actividad `MainMenuNavigationTutorTeacherActivity`, los fragmentos `MeetingsTutorTeacherFragment`, `AssignFragment`, `SessionsFragment`, `AnotationsFragment`, los adaptadores `MeetingAdapter`, `AssignIteamAdapter`, `SessionsItemApdapter` y `AnotationItemAdapter` y por último los modelos `Meeting`, `MeetingItem`, `AssignItem`, `SessionsItem`, `AnotationItem`, `Anotations`, los cuales se encuentran dentro del Anexo C.6, Anexo C.7 y Anexo C.8.

Diagrama de clases para el rol de tutor estudiantil

Para el rol de tutor estudiantil los módulos utilizados son planificación y registro de asistencia, de la misma forma se diseña un diagrama de clases, el cual contiene la actividad `MainMenuNavigationTutorStudentActivity`, los fragmentos `PLanificationStudentTutorshipFragment` y `AsistanceStudentTutorshipFragment`, los adaptadores `ItemPlanificationTutorshipStudentAdapter` y `ItemAsistantsshipAdapter` y modelos `Objective`, `Plnification`, `ItemPlanificationTutorshipStudent` y `ItemAsistantship`, estos se encuentran en el Anexo C.9 y Anexo C.10.

5.2.4. Implementación Cloud Firestore

Este servicio nos provee la posibilidad de enviar y recibir notificaciones push, ya sea para una aplicación móvil y/o web. Al igual que los servicios anteriores de firebase se debe incluir las librerías en el archivo build.gradle en el caso de Android.

```
implementation 'com.google.firebase:firebase-firestore:21.2.1'
```

Figura 5.7: Implementación del servicio Cloud Firestore en Android Studio

5.2.5. Implementación de Firebase Authentication

Firebase Authentication es el servicio que nos provee Firebase para autenticar usuarios en la aplicación ya sea web o móvil. Para esto se debe implementar las librerías necesarias en el archivo build.gradle en el caso de Android Studio.

```
implementation 'com.google.firebase:firebase-auth:18.0.0'
```

Figura 5.8: Implementación del servicio Authentication en Android Studio

5.2.6. Implementación de Firebase Cloud Messaging

Este servicio nos provee la posibilidad de enviar y recibir notificaciones push, ya sea para una aplicación móvil y/o web. Al igual que los servicios anteriores de firebase se debe incluir las librerías en el archivo build.gradle en el caso de Android

```
implementation 'com.google.firebase:firebase-messaging:20.2.4'
```

Figura 5.9: Implementación del servicio Firebase Cloud Messaging en Android

5.3. Ejecución de iteraciones

5.3.1. Iteración 1 - Autenticación de usuarios en aplicación móvil

Dentro de esta iteración se trabajó la configuración e implementación de la autenticación de usuarios en correspondiente a la aplicación móvil. Para esto se crea la actividad de inicio de sesión LoginActivity, la cual le permitirá a los docentes y estudiantes acceder a la aplicación con su cuenta. El mecanismo utilizado es mediante correo electrónico y contraseña, donde el correo corresponde al correo institucional del estudiante y/o del docente. Para lo cual se utiliza un formulario que contiene dos campos de texto o EditText. Una vez que el usuario ingresa los datos estos son obtenidos de los campos de texto, posteriormente son enviados al servicio Firebase Authentication, el cual valida que el usuario ingresado existe, en caso de ser exitoso el usuario puede ingresar a la aplicación, en caso contrario se muestra una alerta de error.

5.3.2. Iteración 2 - Activación de cuentas de usuarios estudiantes

En la iteración dos se realiza la funcionalidad de activación de cuenta de usuario para los estudiantes en la aplicación móvil, en donde una vez cargados los datos de masivos de los estudiantes inscritos, cursos y docentes del semestre. Es necesario que cada estudiante obtenga su cuenta de usuario, para aquello se implementa una actividad llamada AccountActivationActivity. En la cual por medio de dos campos de textos, uno para obtener la matrícula del estudiante y otro para obtener su RUN, siendo estos utilizados para validar que efectivamente el usuario es un estudiante que está inscrito a algunos de los cursos que se dictan en el semestre. Si los datos coinciden con los datos registrados en la base de datos entonces se redirige al usuario a la actividad de verificación de correo electrónico, donde el estudiante debe dirigirse a su correo institucional e ingresar al enlace de verificación que le fue enviado. Una vez verificado el usuario es redirigido al menú principal de la aplicación para el usuario de tipo estudiante. El cual es implementado por la actividad MainMenuNavegationStudent, mostrando los apartados disponibles para el estudiante. Estos apartados son: Reuniones, Asistencia, Evaluaciones.

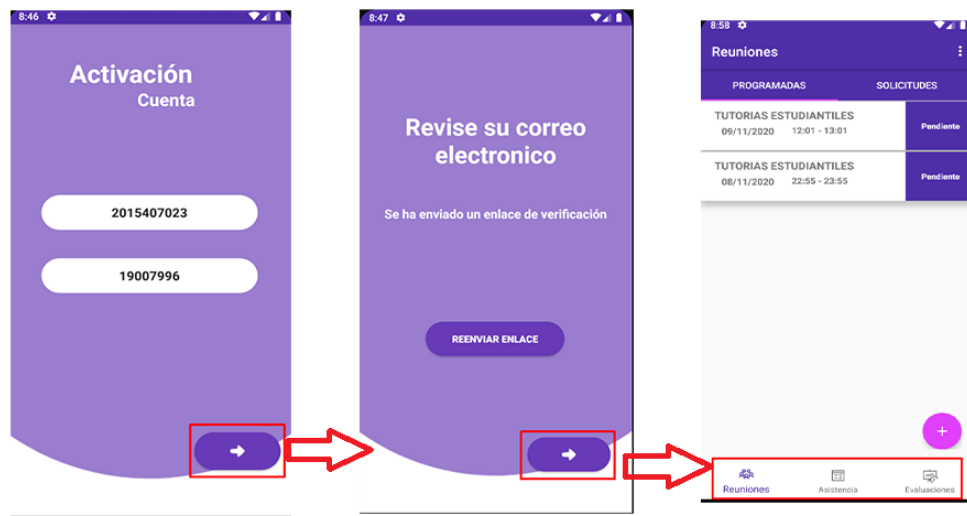


Figura 5.10: Proceso de activación de cuentas de estudiantes

La Figura 5.10 muestra las vistas implementadas para la funcionalidad de activación de cuentas de estudiantes. Además de mostrar los pasos que se debe seguir para este cometido, en primera instancia el usuario debe ingresar su matrícula y RUN, para luego presionar el botón continuar para pasar al paso de verificación de correo como se muestra en la imagen dos de izquierda a derecha, en ese momento al usuario le es enviado un enlace de verificación como el que se muestra a continuación.

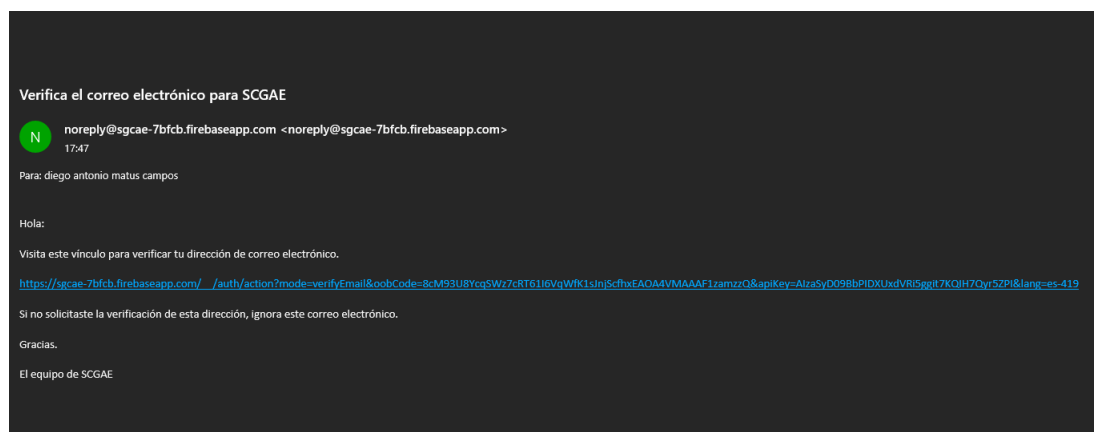


Figura 5.11: Enlace de verificación de correo electrónico

Como se muestra en la Figura 5.11 el enlace se envía al correo, donde al momento de ingresar se valida el correo del usuario indicándole en el navegador al usuario que

ha sido verificado correctamente. Tras a haber verificado el correo, el usuario puede presionar el botón continuar indicado en la Figura 5.10 imagen tres de izquierda a derecha, donde observa los tres apartados que están disponibles para el estudiante.

5.3.3. Iteración 3 - Registrar reuniones de ayudantías en la aplicación móvil

En la iteración tres se desarrollo la funcionalidad de registro de reuniones de ayudantías, el que contempla la implementación del formulario para registra la reunión, además de las vista para mostrar las reuniones pendientes del usuario, para esto se implementó la siguientes vistas.

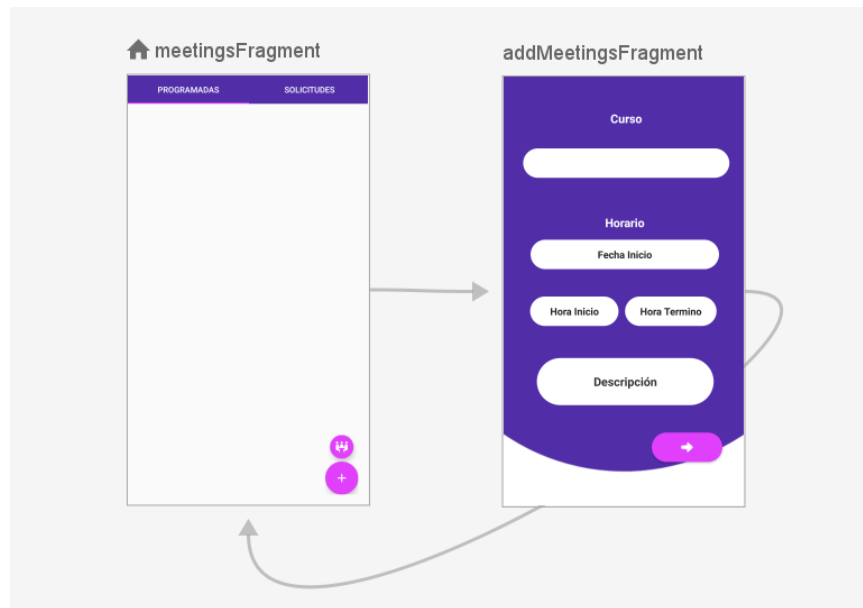


Figura 5.12: Vistas implementadas para registrar una reunión de ayudantía

Como se puede apreciar en la Figura 5.8, tenemos dos fragmentos uno llamado MeetingsFragment y addMeetingsFragment, el primero esta compuesto por :

- **TabLayout:** este se encuentra en la parte superior del fragmento MeetingFrgament, el cual nos permite dividir la vista en pestañas, para este caso divide la vista en la pestaña Programadas y Solicitudes, las cuales hacen referencia a las reuniones programadas y las solicitudes de reuniones pendientes. Ambas

pestañas contienen un `ViewPager`, que es un elemento que nos permite desplazarnos entre pestañas, actualizando dinámicamente la vista según en la pestaña en que nos encontremos.

- `FloatingButtonAction`: en la parte inferior derecha encontramos un botón flotante, el cual despliega un botón destinado a redirigir al fragmento `AddMeetingsFragment`.

Luego tenemos el fragmento `AddMeetingsFragment`, este es el encargado de implementar el formulario necesario para registrar una reunión en el sistema, el cual contiene los siguientes elementos:

- `AutoComplete`: el cual es el encargado de listar y permite filtrar por los cursos que un ayudante tiene asignado, mas abajo nos encontramos con el apartado de `Horario`, el cual incluye tres `TextView` que nos permiten seleccionar una fecha, hora de inicio y termino de la reunión.
- `EditTextMultiLine`: este componente es similar a campo de texto o `EditText`, con la diferencia que nos permite escribir en múltiples líneas, este es utilizado para ingresar una descripción a la reunión.

Para terminar de registrar una reunión se deben completar todos los campos, en caso contrario nos aparece una alerta que nos falta algún campo por rellenar. En caso de éxito, la reunión es almacenada en la base de datos, con el nombre de la colección `reuniones`, la cual internamente tiene un tipo que para este caso corresponde a una reunión de ayudantía. Una vez que ha sido almacenada, se muestra en la vista de reuniones programadas. El detalle de la navegación de estas funcionalidades se encuentra en el Anexo E.

5.3.4. Iteración 4 - Registro de planificación, asistencia de ayudantías y evaluaciones de ayudantes por parte del estudiante en aplicación móvil

En esta iteración se construyeron tres funcionalidades importantes, la cual es la planificación de un ayudantía que genera un ayudante, el registro de asistencia en

ayudantías que están programadas y las evaluaciones a ayudantes por parte del estudiante. Para la primera funcionalidad se implementa el fragmento `PlanificationFragment`, `addPlanificationStepOne` y `addPlanificationStepTwo`, estos se ven de forma detallada a continuación en la Figura 5.13.

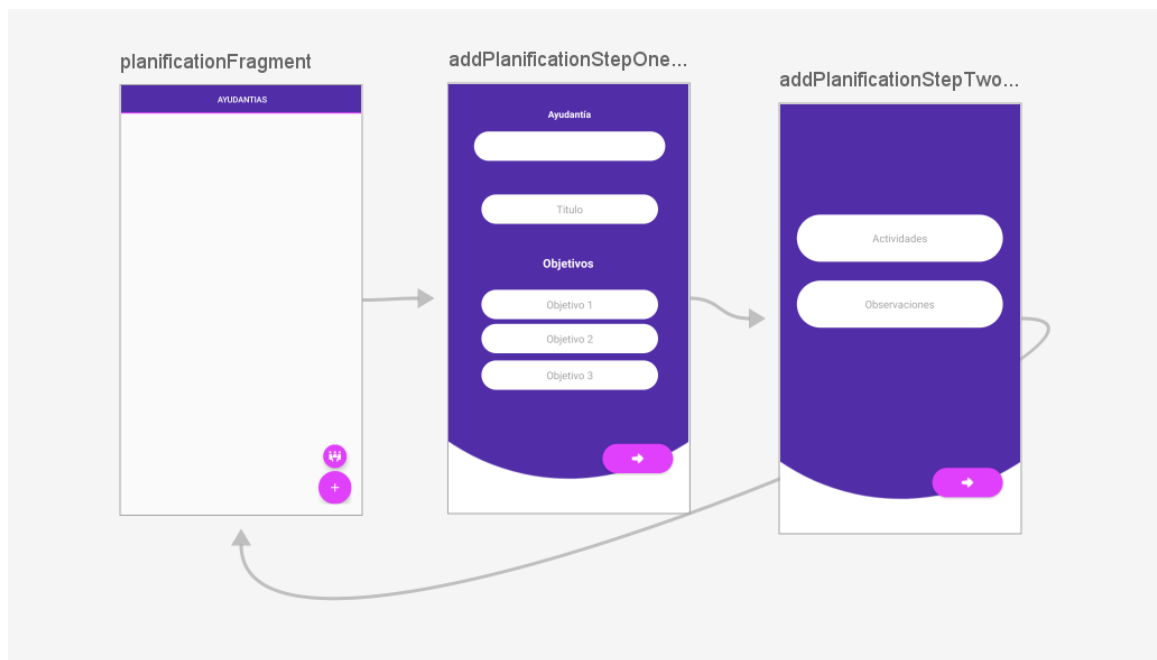


Figura 5.13: Vistas de planificación de ayudantías

La Figura 5.13 nos muestra los fragmentos implementados para la funcionalidad de planificación de ayudantías, partiendo por el fragmento `PlanificationFragment`, el cual contiene:

- `RecyclerView`: el cual nos permite visualizar elementos dinámicamente en forma de una lista, ya sea de forma horizontal o vertical. Este es utilizado para visualizar las planificaciones que se han realizado de una ayudantía.
- `FloatingButtonAction`: en la parte inferior derecha tenemos un botón flotante, el cual despliega un botón que nos permite ir al formulario para agregar una planificación al sistema, el cual se divide en dos fragmentos.

La segunda imagen nos muestra el fragmento `addPlanificationStepOne`, el cual contiene los siguiente elementos:

- **AutoComplete:** es el encargado de listar las reuniones de ayudantías disponibles, para realizar una planificación con el nombre del curso y el horario.
- **EditText:** se encuentran 4 campos de texto, uno dedicado para introducir el título de la planificación, seguido de tres más para ingresar los objetivos de la ejecución de esa ayudantía en específico.

Por último se nos muestra la imagen del fragmento `addPlanficationStepTwo` el cual contiene los siguientes elementos:

- **EditTextMultiline:** en este fragmento se implementan dos campos de texto que permite escribir múltiples líneas de texto, para ingresar las actividades que se van a realizar en ese ayudantía y alguna observación que estime pertinente el ayudante. Una vez que el ayudante presione el botón de continuar se valida los campos anteriores, si los valores no están vacíos o incorrectos procede a almacenar la planificación en la base de datos y redirigir al fragmento `PlanficationFragment` para visualizar las planificaciones, en caso contrario nos alertara de que faltan campos por rellenar.

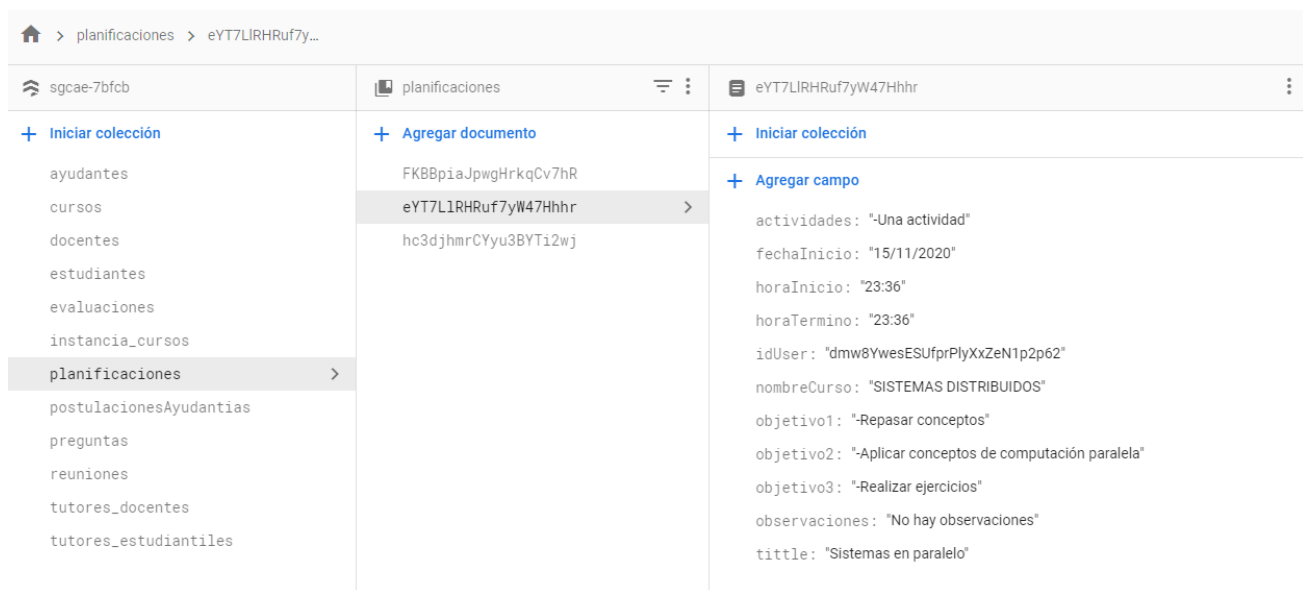


Figura 5.14: Almacenamiento de una planificación de ayudantías en la base de datos

La Figura 5.14 muestra como se almacena una planificación en la base de datos tras haberla ingresado por del formulario, expuesto anteriormente. En primera

instancia se crea la colección planificaciones, con su respectivo documento, el cual contiene los campos:

- `idUser`: es el identificador del usuario que realiza un registro de la planificación.
- `fechaInicio`: es la fecha en que se realizará la ayudantía de la planificación ingresada.
- `horaInicio`: hora en que se inicia una ayudantía planificada.
- `horaTermino`: hora que termina una ayudantía planificada.
- `nombreCurso`: es el nombre del curso de la ayudantía a realizar.
- `objetivos`: son los objetivos que se quieren cumplir con la realización de la ayudantía.
- `tittle`: el título que se dio a la planificación, generalmente será el tema a ver en la ayudantía.
- `observaciones`: alguna observación que el ayudante quiere realizar de la planificación.

Por otro lado, se realiza la implementación de la funcionalidad de registro de asistencia de una ayudantía, que tiene como fin registrar que el ayudante efectivamente está realizando las ayudantías. Para lo cual la funcionalidad se dividió en dos partes, la toma de asistencia por parte del ayudante y el registro de asistencia por parte del estudiante. Para el caso del ayudante, el mecanismo utilizado es la generación de un código único de la reunión de ayudantía, el cual es entregado a los estudiantes para que los ingresen y marquen asistencia. Para esto tenemos las siguientes vistas.

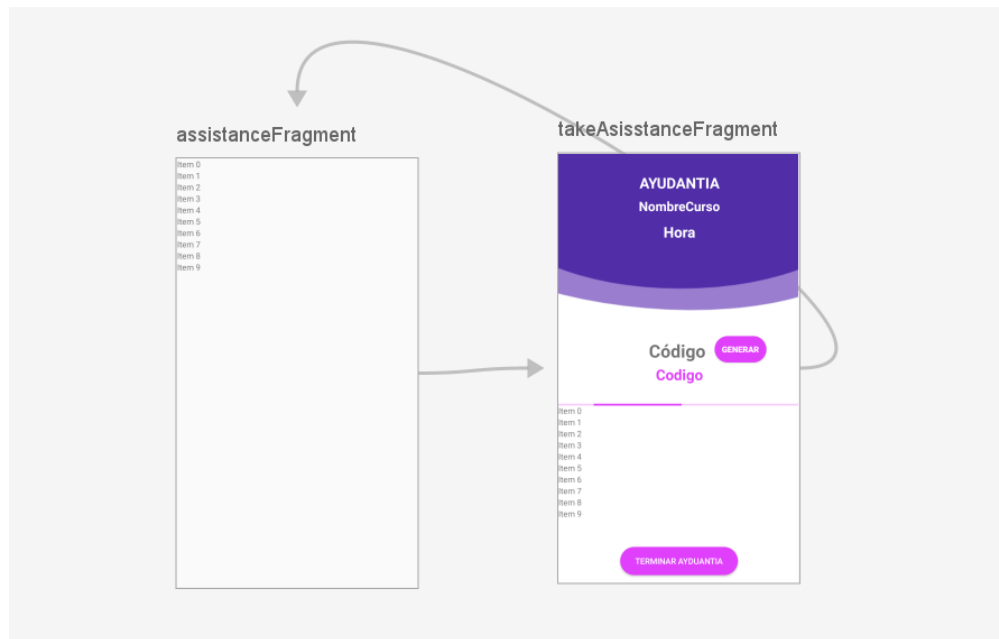


Figura 5.15: Vistas implementadas para tomar asistencia por parte del ayudante

La Figura 5.15 nos muestra el fragmento AssistanceFragment y TakeAsisstanceFragment, donde el primero implementa un RecyclerView para visualizar las reuniones de ayudantías que esta disponibles en ese momento para registrar asistencia, mientras que el fragmento TakeAsisstanceFragment contiene tres TextView que se actualizan dinámicamente según el nombre del curso y hora en que esta prevista esa reunión. Luego podemos ver un botón generar el cual le permite al ayudante generar el código único de la reunión, posteriormente se encuentra un TextView que actualizado con el código generado. Para finalizar con RecyclerView que muestra la lista de participantes de esa reunión de ayudantía. Tras haber generado el código de la reunión este es actualizado en la base de datos en la reunión creada anteriormente.

Por otra lado, el estudiante debe ingresar el código que le es facilitado por el ayudante al momento de la ayudantía. Para esto se construyen las vistas que se muestran a continuación.

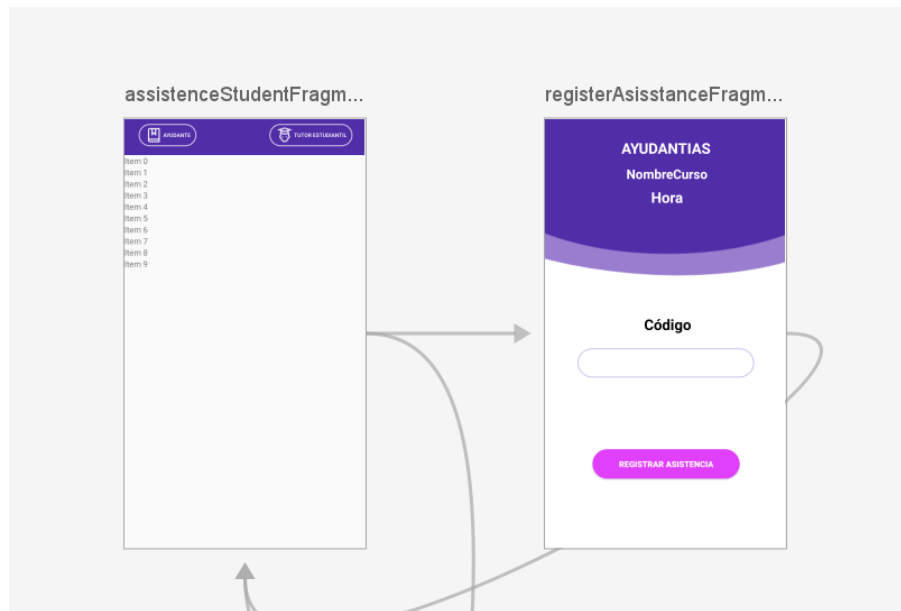


Figura 5.16: Vistas de registro de asistencia de ayudantías por parte del estudiante

La Figura 5.16 nos muestra los fragmentos utilizados para que el estudiante marque asistencia en una ayudantía, los cuales son `assistanceStudentFragment` y `registerAsisstanceFragment`. El primero muestra la lista de reuniones disponibles en ese momento para marcar asistencia, esto se hace utilizando un `RecyclerView`. El estudiante selecciona la reunión en la que quiere marcar asistencia haciendo clic en una de los ítems mostrados, el cual lo redirecciona al fragmento `RegisterAsisstanceFragment`. Una vez en el fragmento para registrar asistencia, nos encontramos con dos `TextView` que se actualizan con la información del curso y hora de la reunión. A continuación tenemos un `EditText` donde el estudiante debe ingresar el código suministrado por el ayudante. Por último el botón registrar asistencia, el cual desencadena la validación del código para verificar que sea correcto y marcar como presente al estudiante.

La última funcionalidad implementada en esta iteración es la evaluación de ayudantes por parte de los estudiantes, con la finalidad de que haya una retroalimentación del desempeño del ayudante. Para este cometido se implementaron los siguientes fragmentos para cumplir con esta funcionalidad.

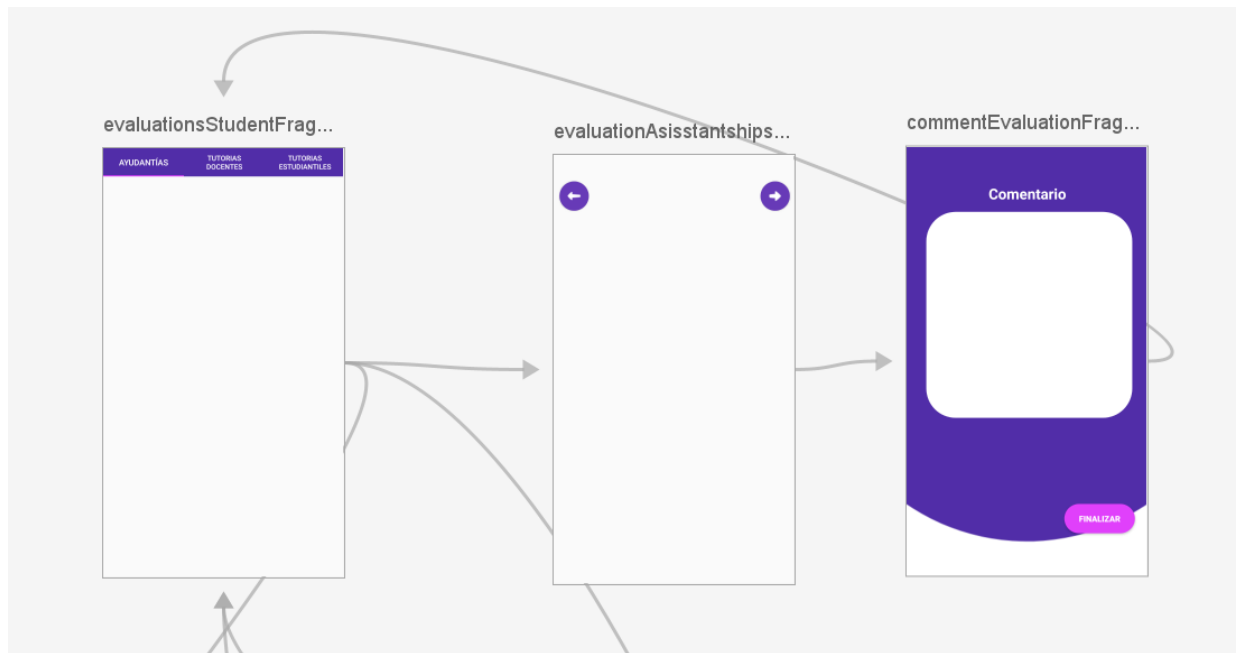


Figura 5.17: Vistas implementadas para la evaluación de ayudantes

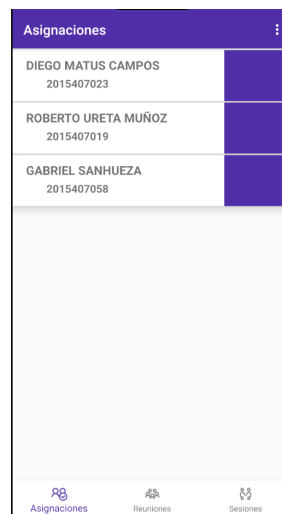
La Figura 5.17 muestra los fragmentos desarrollados para realizar las evaluaciones de ayudantes, partiendo por el fragmento `EvaluationsStundentFragment` el cual es el encargado de mostrar todas las evaluaciones que tiene disponible un estudiante. Este se compone de un `TabLayout` con tres pestañas Ayudantías, Tutorías Docentes y Tutorías Estudiantiles, pero para esta funcionalidad nos enfocaremos en la pestaña ayudantía. Esta pestaña contiene un `ViewPager` el cual es el encargado de listar las evaluaciones disponibles, mediante el uso de un `RecyclerView`. Para evaluar a un ayudante el estudiante tiene que hacer clic en uno de los items listados, el cual lo dirigirá al fragmento `EvaluationAsisstantships`, el cual es el encargado de cargar las preguntas del cuestionario de evaluación, mediante un `ViewPager` dinámico. Una vez que el estudiante haya completado el cuestionario de evaluación es redirigido al fragmento `CommentEvaluationFragment`. Este contiene un `EditTextMultiline` donde el estudiante puede dejar un comentario al ayudante.

Además el documento de evaluación contiene una subcolección llamada `respuestas` las cuales son las repuestas de cada una de la preguntas del cuestionario. Esta subcolección se muestra a continuación.

5.3.5. Iteración 5 - Recepcionar, agendar reuniones y registrar anotaciones de tutorías docentes

Para esta iteración se trabaja las funcionalidades de recepcionar la asignación de los estudiantes, agendar reuniones con ellos y por último poder registrar anotaciones durante una sesión con un estudiante, todo esto con la finalidad de gestionar de mejor forma las tutorías docentes. Para realizar la primera funcionalidad se implementa el fragmento AssignnFragment, el cual contiene una lista de los estudiantes que un docente tiene asignado, haciendo uso de un RecyclerView.

Posteriormente que ha sido notificado, el docente puede ver los estudiantes asignados en el apartado de asignaciones dentro del menú dedicado para el tutor docente que se puede ver a continuación.



The screenshot shows a mobile application interface with a purple header bar containing the text 'Asignaciones' and a three-dot menu icon. Below the header is a list of three students, each with their name and ID number. The list is displayed in a white box with a light gray background. At the bottom of the screen is a navigation bar with three icons and labels: 'Asignaciones', 'Reuniones', and 'Sesiones'.

Asignaciones	
DIEGO MATUS CAMPOS 2015407023	
ROBERTO URETA MUÑOZ 2015407019	
GABRIEL SANHUEZA 2015407058	

Figura 5.18: Almacenamiento de tutor docente en la base de datos

Luego tenemos la implementación de la funcionalidad de agenda de reuniones para el tutor docente, el cual permite agendar una reunión con los estudiantes que tiene asignado, utilizando los fragmentos MeetingsTeacherFragment y AddMeetingsTeacherFragment que se construyen para esta funcionalidad.

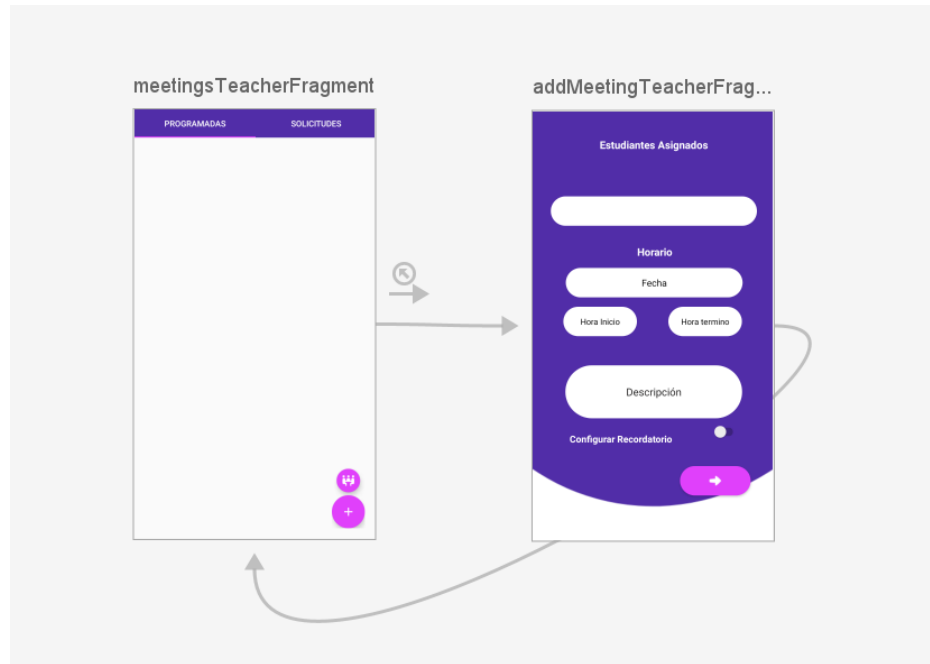


Figura 5.19: Vistas implementadas en la agenda de reuniones de tutores docentes

En la Figura 5.19 se ven los fragmentos implementados para la funcionalidad de reuniones de docentes tutores, por un lado tenemos los fragmentos `meetingsTeacherFragment` y por otro `AddMeetingTeacherFragment`. El primer fragmento es el encargado de mostrar las reuniones programadas y las solicitudes de reuniones de un docente tutor, para esto se utiliza un `TabLayout`, el cual divide la vista en dos pestañas, programadas y solicitudes. En la pestaña de programadas poder observar un `FloatingActionButton`, el cual despliega un segundo botón, que permite ingresar al fragmento `AddMeetingTeacherFragment` encargado de mostrar el formulario para registrar una reunión, el cual contiene los siguientes elementos:

- `AutoComplete`: lista y filtra los estudiantes que tiene asignado un docente, permitiendo seleccionarlo.
- `TextView`: tenemos tres `TextView` los cuales nos permite seleccionar la fecha, hora inicio y termino de la reunión.
- `EditTextMultiline`: este es utilizado para ingresar una descripción a la reunión.
- `Button`: nos permite navegar a la siguiente pantalla, siempre y cuando hayamos rellenado todos los campos.

5.3.6. Iteración 6 - Postular a una ayudantía y seleccionar un ayudante

La iteración seis contempla la implementación de las funcionalidades de postulación de estudiantes a una oferta de ayudantía previamente difundida, luego tenemos la funcionalidad de selección de un ayudante por parte del docente a cargo de un curso, el donde podrá seleccionar de entre todas las postulaciones hechas un estudiante para que sea su ayudante. En primera instancia tenemos la postulación a ayudantías por parte del estudiante, para esto se crearon los siguientes fragmentos.

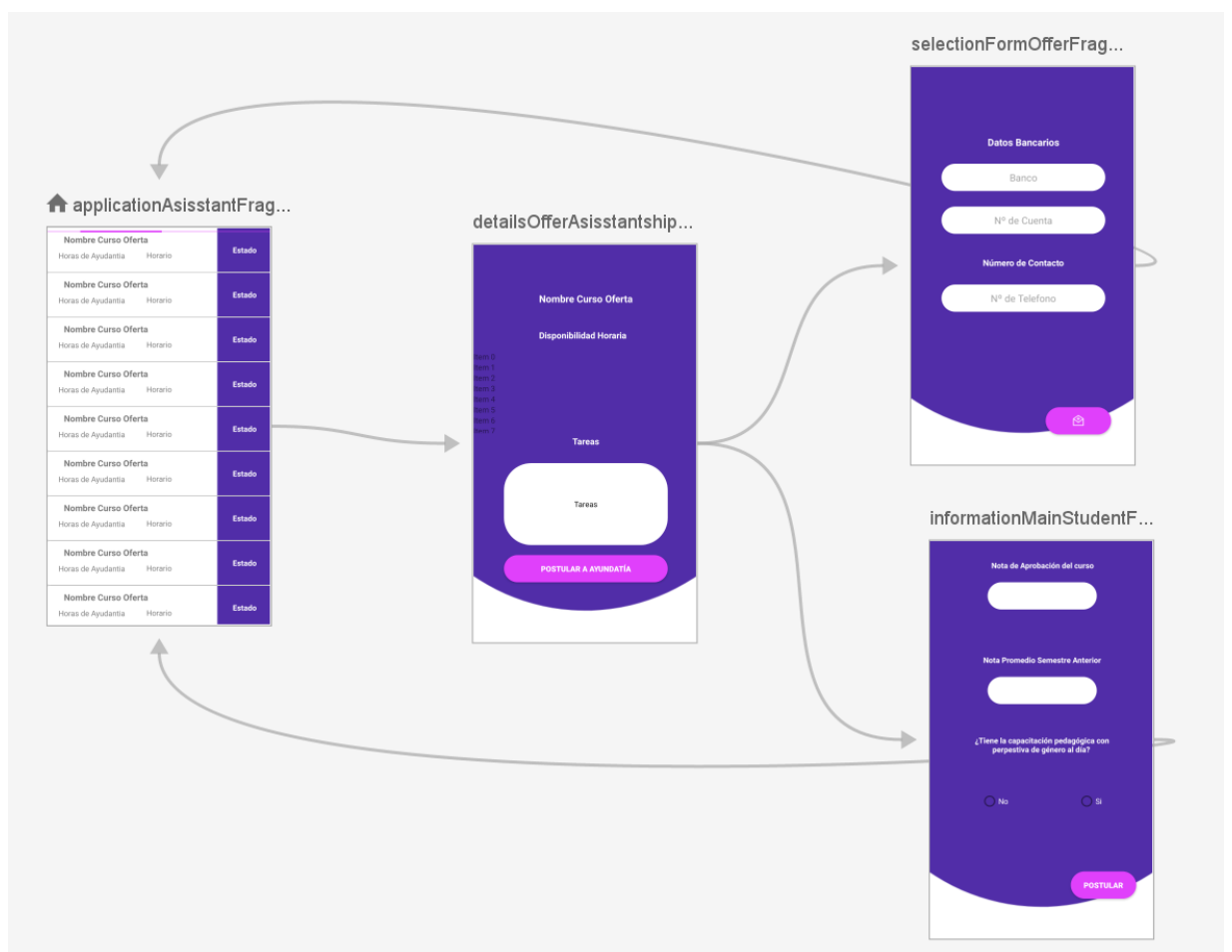


Figura 5.20: Fragmentos implementados para postular a una ayudantía

La Figura 5.20 muestra los fragmentos que se usaron en la funcionalidad de postulación a una ayudantía, siendo estos los siguientes ApplicationAsisstantFragment, donde se lista las ofertas disponibles a los estudiantes, los cuales pueden postular a

una ayudantías, teniendo la posibilidad seleccionar una para realizar la postulación. Luego tenemos el fragmento DetailsOfferAsisstantshipFragment, el cual muestra el detalle de la ayudantía que ha seleccionado el estudiante anteriormente, posteriormente tenemos el fragmento InformationMainStudentFragment siendo el formulario inicial que debe llenar el estudiante con la información de la nota de aprobación del curso, la nota promedio del semestre anterior y si tiene completada la capacitación practica de perspectiva de genero. Una vez que un estudiante a ha sido seleccionado debe volver ingresar a la postulación de ese ayudantía para completar el proceso utilizando el formulario de selección implementado en el fragmento selecctionFormOfferFragment.

La segunda funcionalidad implementada fue la selección de ayudantes por parte del docente, para esto se crearon los fragmentos que se ven a continuación.

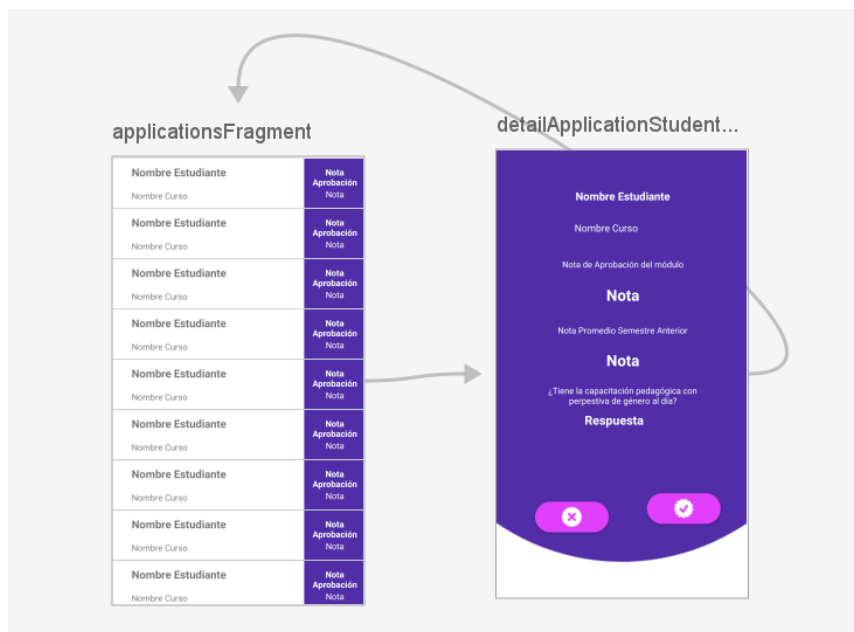


Figura 5.21: Fragmentos implementados para la selección de ayudantes

En la Figura 5.21 podemos observar el fragmento ApplicationsFragment el cual lista las postulaciones de los estudiantes a alguna oferta de ayudantía, donde el docente puede seleccionar una para ver el detalle de su postulación, además de poder aceptar o rechazar su postulación, todo esto dentro del fragmento DetailApplicationStudentFragment.

5.3.7. Iteración 7 - Registrar planificación, asistencia de tutorías estudiantiles y evaluación de tutores estudiantiles

Dentro de la iteración siete se desarrollaron las funcionalidades de registro de planificación de tutorías estudiantiles, lo cual permite al tutor llevar cuenta de los temas que se han tratado en las sesiones. Luego tenemos la funcionalidad de registro de asistencia de en las sesiones, permitiéndole al tutor contar a los estudiantes que asisten a sus sesiones. Por último tenemos, la evaluación de tutores estudiantiles. Para la primera funcionalidad se implementaron los siguientes fragmentos:

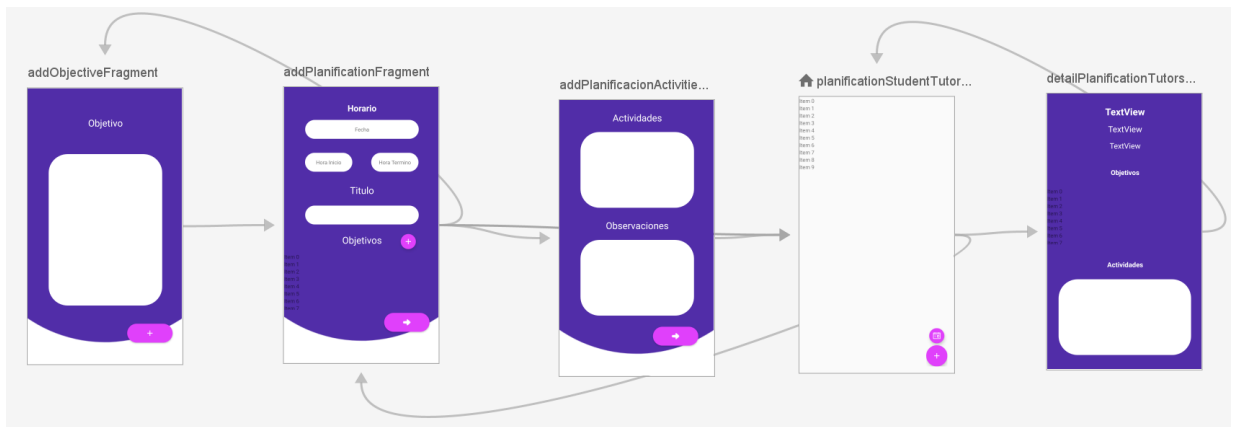


Figura 5.22: Fragmentos implementados para la funcionalidad de planificación de tutorías estudiantiles

La Figura 5.22 muestra los fragmentos que fueron implementados, siendo estos `PlanificationStudentTutorFragment` el cual es el encargado de listar las planificaciones que se han hecho dentro de una semana. Luego nos encontramos los fragmentos `AddObjectiveFragment`, `AddPlanificationFragment` y `AddPlanificacionActivitiesFragment`, todos estos nos permiten agregar la información de una planificación siendo su título, horario, objetivos y actividades a realizar en una sesión de tutorías docentes. Por último tenemos el fragmento `DetailPlanificationTutorStudentFragment` el cual nos permite ver el detalle de una planificación listada.

La segunda funcionalidad implementada en esta iteración fue el registro de asistencia a una sesión de tutorías estudiantiles, tanto para tomar asistencia por parte del tutor estudiantil y para marcar asistencia por parte del estudiante. Para esto se crearon los siguiente fragmentos, los cuales son mostrados a continuación.

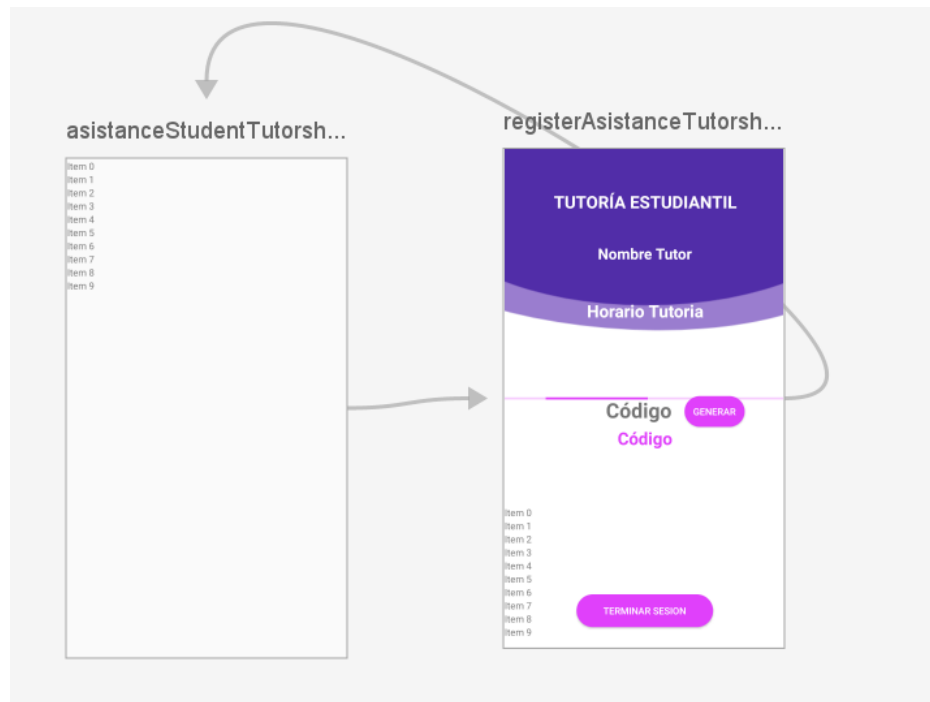


Figura 5.23: Fragmentos implementados para la funcionalidad de tomar asistencia de sesiones de tutorías estudiantiles

La Figura 5.23 muestra los fragmentos implementados para construir la funcionalidad de registrar asistencia en las sesiones de tutorías, la cual incluye la creación del fragmento `asistenciaStudentTutorshipFragment`, el cual es el encargado de listar las sesiones disponibles en esa fecha para tomar asistencia a los estudiantes. Una vez que el tutor ha seleccionado la sesión de la lista, será redireccionado al fragmento `registerAsistenciaTutorship`, donde se muestra la información de la sesión, además de poder generar el código de la sesión, el cual debe ser entregado a los estudiantes asistentes para que marquen asistencia. Por último tenemos un botón para terminar una sesión de tutorías. Ahora bien para el caso de marcar asistencia para el estudiante, se construyeron los siguientes fragmentos.

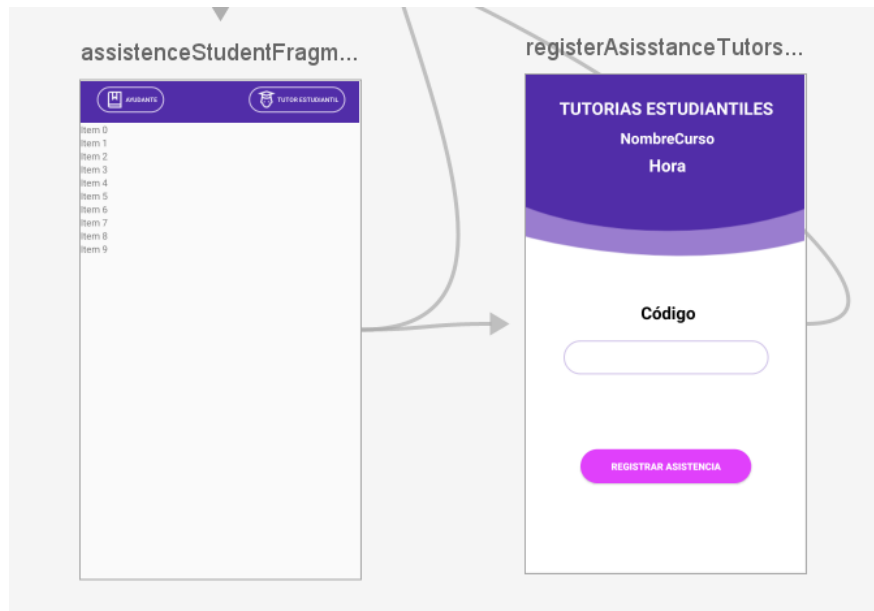


Figura 5.24: Fragmentos implementados para la funcionalidad de marcar asistencia de sesiones de tutorías estudiantiles para el estudiante

En la Figura 5.24 encontramos los fragmentos implementados para marcar asistencia en un sesión de tutorías estudiantiles, creando el fragmento `asistenceStudentFragment` el cual se encarga de listar las reuniones disponibles para marcar asistencia, donde el estudiante podrá seleccionar una rediriéndo al fragmento `registerAsisstanceTutorshipFragment`, el cual contiene la información del horario, además de un campo de texto (`EditText`) el cual le permitirá al estudiante ingresar el código de la sesión con el fin de quedar presente.

5.4. Pruebas

En esta sección se aborda el plan de pruebas realizado para probar las funcionalidades de la aplicación móvil. Cabe destacar que las pruebas que se tratan en esta sección son pruebas de caja negra. Además es importante mencionar que solo se muestran las pruebas a las funcionalidades más importantes de la aplicación móvil, el detalle de las demás pruebas y de las evidencias serán encontrados en el Anexo G.

Cuadro 5.1: Prueba de registro de reuniones de ayudantías.

Prueba de registro de reuniones de ayudantías	
Propósito	Verificar la correcta creación de reuniones cuando un ayudante completa el formulario de registro de reuniones de ayudantías.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado a la aplicación móvil, y ser un ayudante.
Entradas	Curso a realizar la ayudantía, fecha, hora de inicio y término de la reunión, observación.
Salidas	Mostrar mensaje de éxito, creación del registro en la base de datos y mostrar la reunión en el apartado de reuniones de ayudantías pendientes.
Resultado de la prueba	Correcto.

Cuadro 5.2: Prueba de registro de reuniones de ayudantías.

Prueba de registro de planificaciones de ayudantías	
Propósito	Verificar la correcta creación de una planificación cuando un ayudante completa el formulario de registro de planificación de ayudantías
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado a la aplicación móvil, y ser un ayudante.
Entradas	Reunión de ayudantía, título, objetivos, actividades, y observación.
Salidas	Mostrar mensaje de éxito, creación del registro en la base de datos y mostrar la planificación en el apartado de planificaciones ayudantía.
Resultado de la prueba	Correcto.

Cuadro 5.3: Prueba de registro de asistencia de ayudantías ayudante.

Prueba de registro de asistencia de ayudantías ayudante	
Propósito	Verificar la correcta actualización del código de reunión de ayudantía.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado a la aplicación móvil y ser un ayudante.
Entradas	Seleccionar una reunión del listado, generar código de reunión de ayudantía.
Salidas	Actualizar código de reunión del registro en la base de datos, mostrar código al ayudante.
Resultado de la prueba	Correcto.

Cuadro 5.4: Prueba de registro de asistencia de ayudantías estudiante.

Prueba de registro de asistencia de ayudantías estudiante	
Propósito	Verificar la correcta actualización del estado de presente del estudiante en una reunión de ayudantía.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado a la aplicación móvil.
Entradas	Seleccionar una reunión del listado, código de reunión de ayudantías.
Salidas	Actualizar campo presente de dentro de los participantes de la reunión, actualizar texto del botón presente.
Resultado de la prueba	Correcto.

Cuadro 5.5: Prueba de selección de ayudante.

Prueba de selección de ayudante	
Propósito	Verificar la correcta creación de ayudante, luego de aceptar la solicitud de un estudiante a ayudante.
Prerrequisitos	El usuario debe haber ingresado a la aplicación móvil, y ser un docente.
Entradas	Seleccionar una postulación de ayudante, rechazar o aceptar solicitud.
Salidas	Creación de registro de ayudante en base de datos, envío de notificación al estudiante, mostrar en el apartado de ayudantes en el menú del docente.
Resultado de la prueba	Correcto.

Cuadro 5.6: Prueba de postulación a oferta de ayudantía.

Prueba de postulación a ayudantía.	
Propósito	Verificar la correcta creación solicitud de postulación a una oferta de ayudantía.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado en la aplicación.
Entradas	Seleccionar una oferta de ayudantía, nota de aprobación del curso, nota promedio semestre anterior, capacitación al día.
Salidas	Creación de registro de postulación a ayudantía en la base de datos, envío de notificación a docente de la postulación, mostrar en el apartado de postulaciones en el menú del docente.
Resultado de la prueba	Correcto.

Cuadro 5.7: Prueba de evaluación de ayudantes.

Prueba de evaluación de ayudantes	
Proposito	Verificar la correcta creación de la evaluación, respuestas y el puntaje calculado.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado en la aplicación.
Entradas	Seleccionar un ayudante en el apartado de evaluaciones, completar el cuestionario de evaluación del ayudante, dejar un comentario al ayudante.
Salidas	Creación de registro de evaluación de ayudante, además del registro de respuestas asociados a una evaluación.
Resultado de la prueba	Correcto.

Cuadro 5.8: Prueba de evaluación de ayudantes por parte del docente.

Prueba de evaluación de ayudantes por parte del docente	
Proposito	Verificar la correcta creación de la evaluación, respuestas y el puntaje calculado.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado en la aplicación.
Entradas	Seleccionar un ayudante en el apartado de ayudantes, completar el cuestionario de evaluación del ayudante, dejar un comentario al ayudante.
Salidas	Creación de registro de evaluación de ayudante, además del registro de respuestas asociados a una evaluación.
Resultado de la prueba	Medianamente Correcto.

Cuadro 5.9: Prueba de evaluación de tutores docentes.

Prueba de evaluación de tutores docentes	
Proposito	Verificar la correcta creación de la evaluación, respuestas y el puntaje calculado.
Prerrequisitos	El usuario debe haber activado su cuenta, haber ingresado en la aplicación.
Entradas	Seleccionar un tutor docente en el apartado de evaluaciones, completar el cuestionario de evaluación del tutor docente, dejar un comentario al tutor docente.
Salidas	Creación de registro de evaluación de tutor docente, además del registro de respuestas asociados a una evaluación.
Resultado de la prueba	Medianamente Correcto.

5.5. Resumen del capítulo

El capítulo muestra la aplicación de la metodología elegida, presentando el detalle del flujo de trabajo que sigue la aplicación móvil, para luego detallar el diseño mostrando la arquitectura física, lógica y diagrama de clases correspondientes. Posteriormente se habla sobre la ejecuciones de las iteraciones planificadas en el proyecto, además sobre las funcionalidades que se implementaron en cada una de ellas. Para finalizar presentando el plan de pruebas ejecutado para la implementación de la aplicación móvil.

6. Resultados

En este capítulo se analizan los resultados obtenidos en las encuestas aplicada a estudiantes de diferentes cursos los cuales probaron diferentes perfiles de la aplicación móvil, siendo estos perfil de estudiante, ayudante y tutor estudiantil. Para el caso de los perfiles de docente y tutor docente fueron probados por algunos docentes. Posteriormente se aplicaron los cuestionarios para la aplicación web la cual solo fue probado por la directora de escuela y su asistente. Para evaluar la usabilidad de las dos aplicaciones se utiliza un cuestionario de usabilidad de escalas, además para el cumplimiento de los objetivos se realizaron encuestas específicas que apuntan a cada uno de ellos.

6.1. Prueba de usabilidad de la aplicación móvil

Para aplicar las pruebas de usabilidad de la aplicación móvil se utiliza un ambiente controlado, donde un total de doce estudiantes hicieron uso de la aplicación realizando las actividades definidas para el perfil de estudiantes y luego el perfil de de ayudante. Las actividades realizadas fueron las siguientes:

- Activación de cuenta de estudiante: cada estudiante debía activar su cuenta ingresando a sus matrícula y rut sin dígito verificador.
- Postulación a ayudantías: a cada estudiante se le pidió que realizará una postulación a alguna oferta de ayudantía disponible.
- Creación de reuniones y registro de asistencia de ayudantías: a cada estudiante se le pidió que creara reuniones de ayudantías y generar el código de la reunión el cual es utilizado luego para registrar asistencia. Luego se le pidió al estudiante

que registrara asistencia con el código con fin de probar el apartado de registro de asistencia.

- Evaluación de ayudantes: por último se le pedía al estudiante que evaluara al un ayudante.

Luego se le pidió a tres docentes que probaran el perfil de docente y a dos docentes que probaron el perfil de docente tutor, para este cometido yo actué como estudiante el cual postulaba a una oferta de ayudantía de algunos de sus cursos, con el fin que ellos aceptaran mi solicitud de postulación ayudantías. Luego de haberme aceptado, que me evaluaran como ayudante.

Una vez que realizaron estas actividades se le pidió que respondieran el cuestionario SUS, el cual cuenta con diez preguntas las cuales son respondidas según una escala Likert de uno a cinco, siendo uno totalmente en desacuerdo y cinco totalmente de acuerdo. Este cuestionario y sus respuestas están disponible en el Anexo H.1

Una vez analizado y tabulados los datos de la prueba de usabilidad de la aplicación móvil, haciendo uso del algoritmo definido por la prueba SUS, el cual nos dice que restemos uno al valor de las preguntas impares y restemos el valor de la respuesta menos uno para las preguntas pares.

Realizando este algoritmo el valor obtenido para la aplicación móvil es de 75,16 puntos, el cual si revisamos la escala de puntajes nos dice que la usabilidad es buena pero podría ser mejor. Esta escala se encuentra en el Anexo H.3

6.2. Prueba de usabilidad plataforma web

Para probar la aplicación web, se pidió a la directora de escuela y su asistente que probaran la aplicación realizando las actividades de carga masiva de datos, asignación de estudiantes a tutores docentes y asignación a tutores estudiantiles. Para evaluar la usabilidad de la plataforma web, se aplicó el mismo cuestionario que la aplicación móvil solo que esta vez las preguntan apuntaba a la aplicación web.

Una vez aplicado el cuestionario se procedió a calcular el puntaje de la prueba SUS dando como resultado un puntaje de 66,25 puntos, el cual sitúa la aplicación web con una usabilidad que esta bien, pero tiene amplio rango de mejora.

6.3. Verificación del cumplimiento de los objetivos

6.3.1. Comprobación de objetivo específico N° 1

El primer objetivo específico del proyecto esta definido como:

- Construir una aplicación móvil y web que permita gestionar el proceso de ayudantías y tutorías.

Este objetivo se comprueba con la implementación las funcionalidades de la aplicación móvil y aplicación web explicada en la ejecución de iteraciones de los capítulos cuatro y cinco respectivamente. Además de las pruebas unitarias y pruebas de usabilidad, las cuales están últimas están documentadas en el Anexo H.

6.3.2. Comprobación de objetivo específico N° 2

El segundo objetivo del proyecto es definido como:

- Disminuir el tiempo invertido en la gestión y obtención de información del proceso de ayudantías.

Para verificar este objetivo se hicieron encuestas del estado actual de cuanto lleva cada uno de los subprocessos realizados en el proceso de ayudantías, para luego de haber utilizado las aplicaciones aplicar la encuesta de después para verificar la disminución del tiempo. Estas preguntas serán presentadas a continuación.

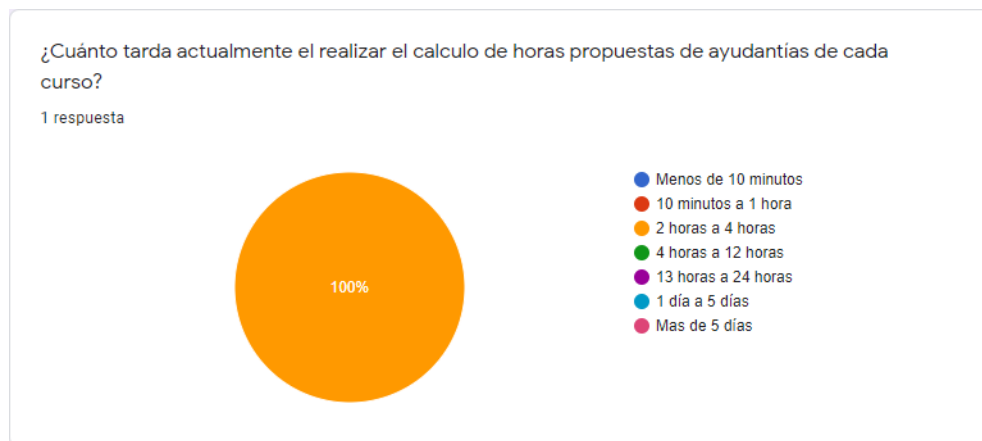


Figura 6.1: Tiempo de cálculo de horas propuestas de horas de ayudantía antes de usar la aplicación web

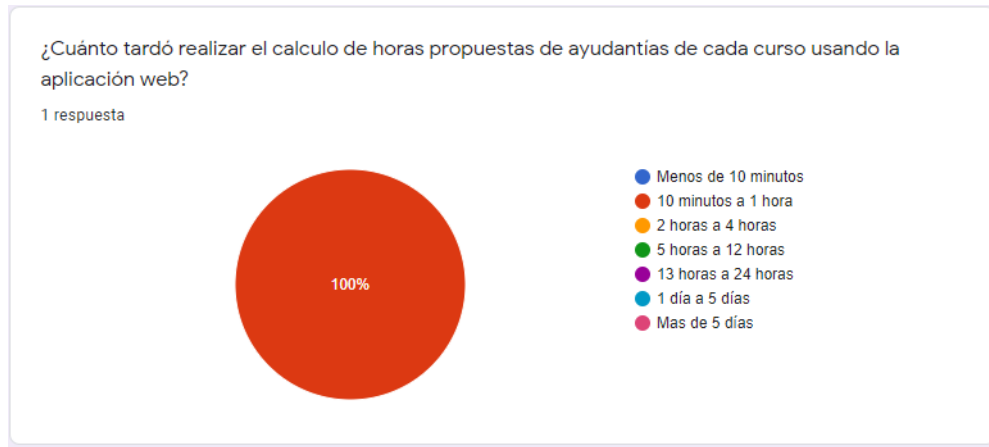


Figura 6.2: Tiempo de cálculo de horas de propuestas de horas al usar la aplicación web

En la Figura 6.1 y Figura 6.2 se nos muestra el tiempo antes de haber usado la aplicación móvil, donde la asistente nos indica que el tiempo que se demora en realizar el cálculo de horas de ayudantías propuestas de ayudantías de cada curso es entre dos a cuatro horas. En cambio al usar la aplicación web indica que se demoró entre diez minutos a una hora por lo que si hay una disminución en este subproceso.

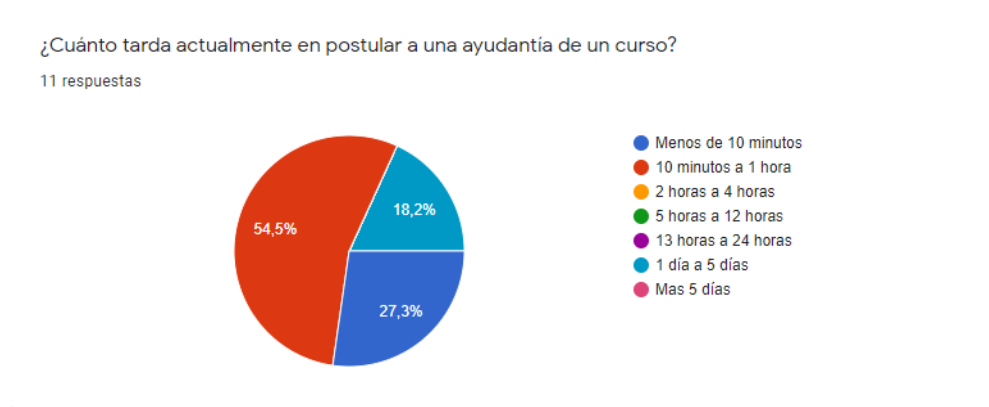


Figura 6.3: Tiempo de postulación a ayudantías antes de usar la aplicación móvil

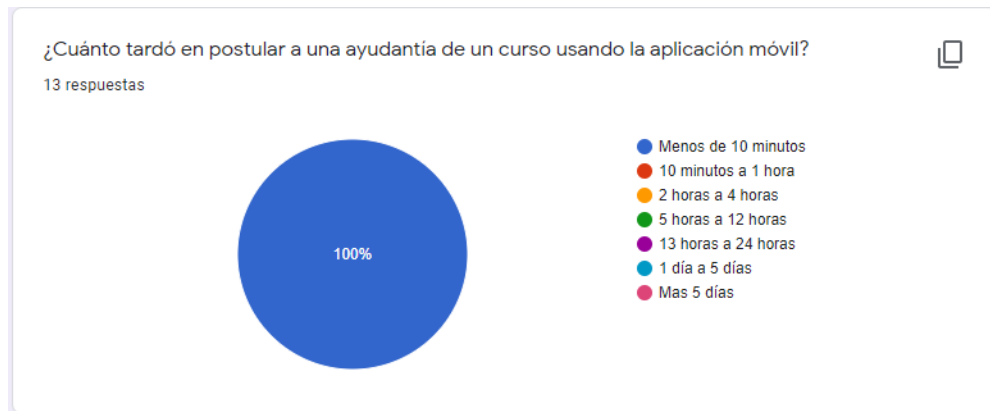


Figura 6.4: Tiempo de postulación a ayudantías al usar la aplicación móvil

Como se puede observar en la Figura 6.3 y 6.4 el tiempo que se tarda actualmente en realizar una postulación a ayudantías según 54,5% de las estudiantes va desde los diez minutos a una hora, un 27,3% demora menos de diez minutos. Luego si comparamos cuanto se tardaron usando la aplicación móvil tenemos que el 100% dijo que menos de diez minutos. Como podemos observar si hay una disminución en el tiempo de postulación a ayudantías.

Ahora bien para el caso de la selección de ayudantes, tenemos las siguientes preguntas que se muestran en la Figura 6.3 y 6.4.

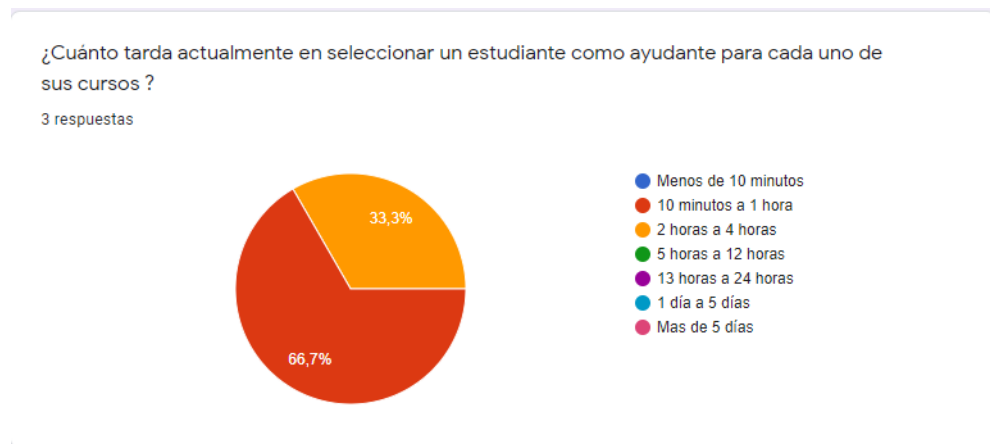


Figura 6.5: Tiempo de selección de ayudantes después de usar la aplicación móvil

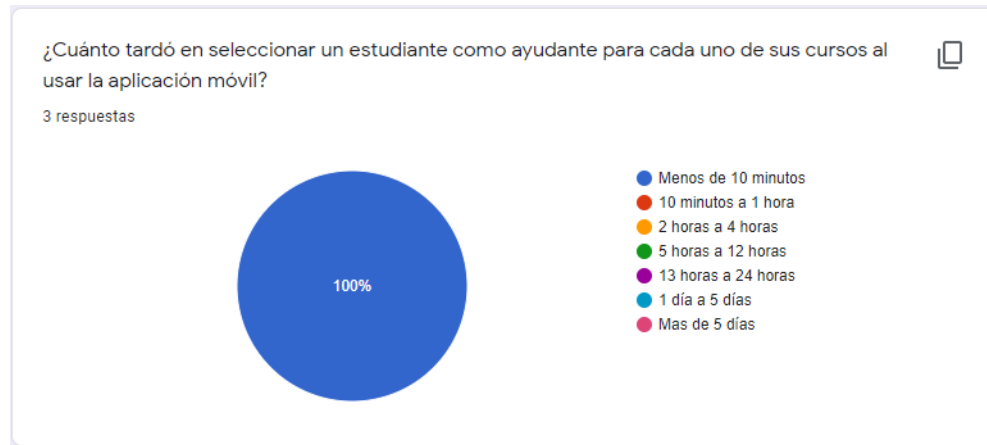


Figura 6.6: Tiempo de selección de un ayudante después de usar la aplicación móvil

Se puede apreciar la Figura 6.5 y Figura 6.6 que el tiempo de selección de ayudantes actualmente un 66,7% concuerda que tarda entre diez minutos y una hora, el 33% restante indica que tarda entre dos a cuatro horas. Ahora bien después de utilizar la aplicación móvil un 100% dice que demora menos de diez minutos, por consiguiente si hay una disminución de tiempo en este apartado.



Figura 6.7: Tiempo de obtención de resumen de ayudantes del semestre antes de usar la aplicación web

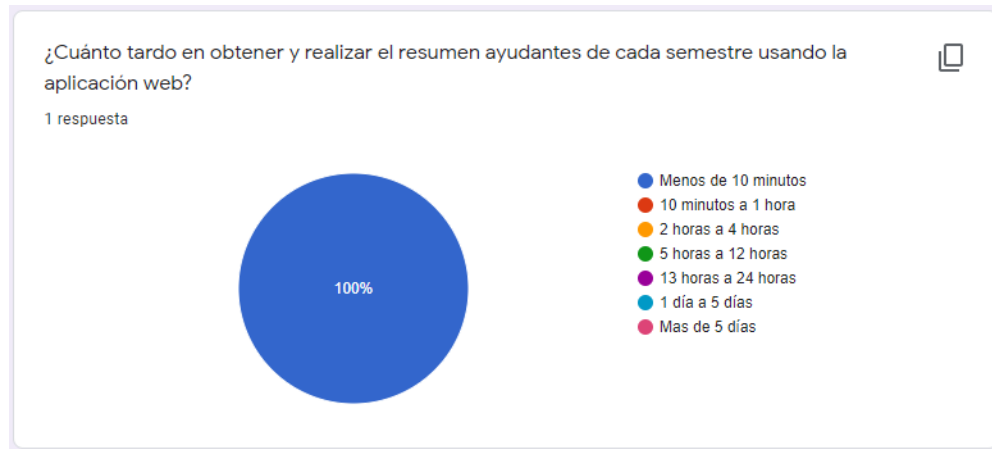


Figura 6.8: Tiempo de obtención de resumen de ayudantes del semestre al usar la aplicación web

Como muestran las Figura 6.7 y Figura 6.8 tenemos que el tiempo que tarda la asistente en obtener el resumen de ayudantes del semestre esta entre un día a cinco días, ahora bien usando la aplicación demoró menos de diez minutos. Demostrando una diferencia considerable usando la aplicación web a favor de esta.

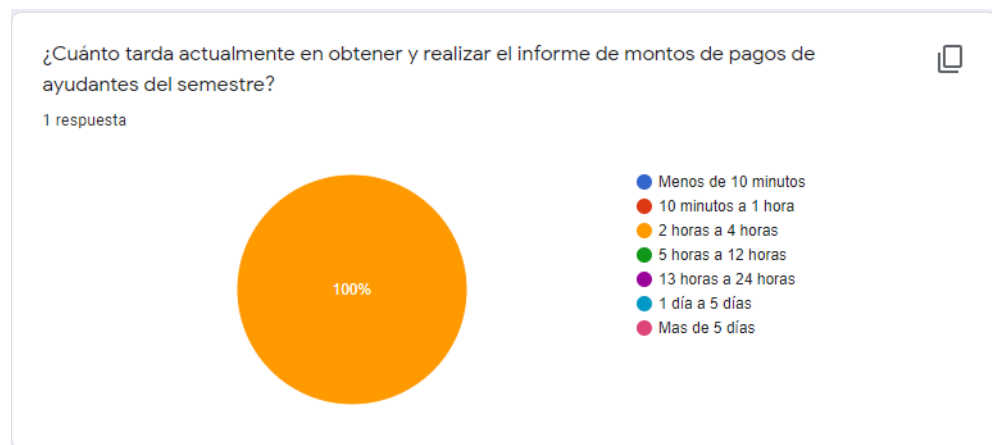


Figura 6.9: Tiempo de obtención de informe de pagos de ayudantes del semestre antes de usar la aplicación web

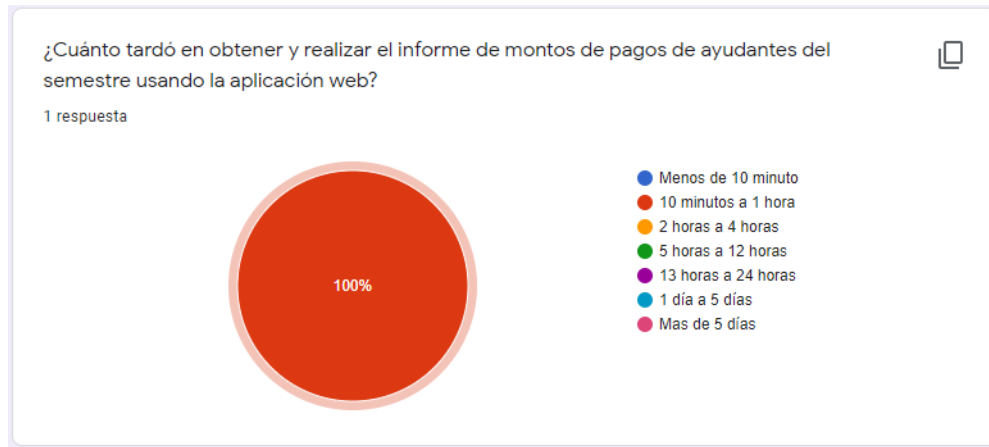


Figura 6.10: Tiempo de obtención de resumen de ayudantes del semestre al usar la aplicación web

En la Figura 6.9 y Figura 6.10 se puede notar que la asistente se demora de dos a cuatro horas para realizar el informe de pagos de ayudantes, en cambio usando la aplicación web se demora entre diez minutos a una hora, disminuyendo el tiempo invertido en este subproceso.

Observando los resultados de cada una de las preguntas que se hicieron con el fin de saber si el sistema cumple con el objetivo de disminuir el tiempo del proceso. Dándonos cuenta que en cada uno de sus subprocesos se ha disminuido el tiempo por consecuencia si se ha disminuido el tiempo del proceso de ayudantías y si se cumplió este objetivo.

6.3.3. Comprobación de objetivo específico N° 3

El tercer objetivo específico esta definido como:

- Proveer un entorno de monitoreo y control para el docente tutor de los estudiantes que tiene asignados.

Este objetivo se comprueba con la implementación de las funcionalidades para el apartado de tutorías docentes siendo las asignación de un tutor docente, la recepción de estudiantes asignados, la posibilidad de generar una reunión con el estudiante y finalmente generar anotaciones de las sesiones que ha tenido con el estudiante. Estas están detalladas en el apartado de ejecuciones de iteraciones los capítulos cuatro y cinco.

Además de las preguntas realizadas a dos docentes que probaron el apartado de tutorías docentes. Estas preguntas se presentan a continuación.

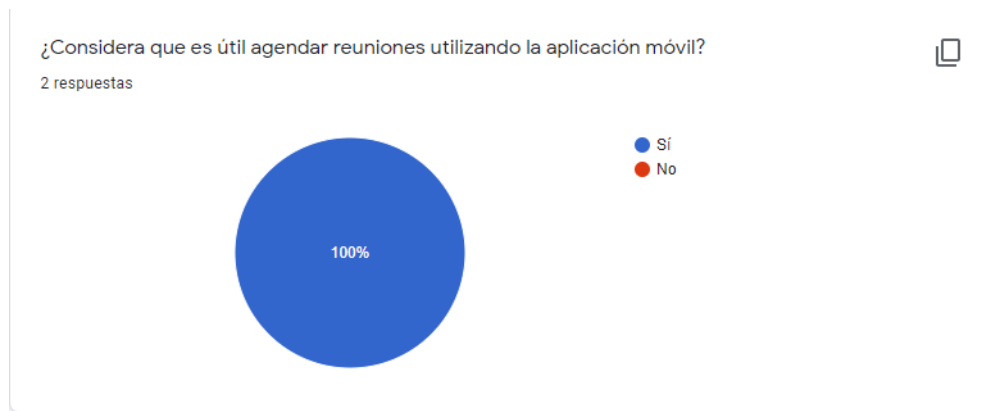


Figura 6.11: Apartado de agenda de reuniones de tutor docente a estudiante



Figura 6.12: Apartado de generación de anotaciones de sesiones



Figura 6.13: Interacción entre roles

Como se observa en las Figuras 6.11, 6.12 y 6.13 los docentes que probaron ese apartado están de acuerdo que es de utilidad el apartado que se implementó en tutorías docentes, pero al ser una muestra muy baja no podemos afirmar que así sea pero si que está implementado el entorno de monitoreo para las tutorías docentes, por ende se cumple el objetivo número tres.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

Para concluir el trabajo realizado es importante mencionar que el objetivo general del proyecto

- Establecer un mecanismo de control y gestión interno para el proceso de ayudantías y tutorías docentes realizadas por la escuela de Ingeniería Civil en Computación, mediante el desarrollo de una aplicación móvil y aplicación web de administración.

Analizando los resultados obtención de los resultados de las encuestas de usabilidad y las encuestas para verificar los objetivos específicos podemos afirmar que el sistema proporciona un mecanismo que permite gestionar las ayudantías de tal forma que disminuye el tiempo invertido en este cometido. Además de proporcionar un entorno de monitoreo para tutorías docentes por ende se cumple el objetivo general de este proyecto, pero con un rango de mejora en la usabilidad del sistema implementado.

Cabe destacar que esta disminución en la usabilidad se debe al poco tiempo que se tuvo para desarrollar el sistema, ya cerca del periodo de finalización de este, hubieron cambios en los requerimientos por parte del cliente, lo que afectó tanto a la planificación y diseño inicial del sistema, incluso este cambio puso en juego la continuación o no del proyecto, pero gracias a la flexibilidad que nos otorgan las metodologías ágiles ante los cambios es que se logró incorporar las nuevas funcionalidades. Sumándose a lo anterior, de esta forma se respalda la elección de la metodología ágil Scrumban fue una decisión correcta ya que permitió continuar el proyecto, pero de igual forma

me enseñó lo difícil que es seguir al pie de la letra una metodología, ya que como en todo desarrollo existen retrasos y dificultades que enfrentar.

Sumado a lo anterior es importante destacar que el tamaño del proyecto no era menor, ya que contemplaba tres flujos de procesos distintos, los cuales tuvieron que ser analizados cada uno para lograr una unificación en el sistema desarrollado, el cual involucró desarrollar dos aplicaciones distintas para fines distintos pero que interaccionan entre si, mediante una base de datos en común. Esta es otra de las razones por las que no se logró el mejor estándar de usabilidad, debido a todo el proceso y dificultad de plasmar estos procesos dentro del sistema.

Por último mencionar que el desarrollo de este proyecto fue útil para mi formación académica ya que es la primera vez que realizó un proyecto completo desde cero, lo que me permitió aprender nuevas tecnologías web y móvil, como lo son Angular y Android haciendo uso del lenguaje Kotlin. Además de aprender a utilizar bases de datos No SQL como lo es Cloud Firestore del servicio de Firebase.

7.2. Trabajo Futuro

En el transcurso del desarrollo del sistema hubo cambios en los requisitos por parte del cliente, por lo que hay varias funcionalidades que no se lograron y otros que no están logradas al 100% por lo que en futuro deberían ser implementadas, estas son:

- Comptatibilidad con iOS.
- Mejorar la adaptación de la interfaz a dispositivos móviles de menos de 5.4 pulgadas de pantalla.
- Mejorar la usabilidad y navegación de la aplicación móvil y web.
- Considerar incluir evaluaciones a tutores estudiantiles.
- Incluir la posibilidad de gestionar recordatorios de las reuniones agendadas.
- Incluir la posibilidad al docente de pedir un aumento de horas de ayudantías.
- Terminar de manera automática las reuniones que están siendo ejecutadas.
- Implementar un apartado de super administrador para el registro de nuevos usuarios para la aplicación web.

Glosario

Activity: es una clase especializada que permite pintar interfaces gráficas al usuario.

Fragment: es una porción de la interfaz que puede agregarse o eliminarse de una actividad y puede ser reutilizado en otras actividades.

View Pager: es un componente gráfico que nos permite agrupar varias vistas, permitiéndonos ir desplazándonos entre ellas.

Tab: es un componente gráfico que nos permite separar una porción de una interfaz gráfica en una pestaña.

TabLayout: es un componente gráfico que nos permite separar una interfaz gráfica en pestañas o tabs.

Recycler View: es un componente gráfico que nos permite listar un conjunto de elementos de forma dinámica.

Adapter: es un puente entre la vista y un conjunto de elementos que prepara los datos para ser mostrados de forma dinámica.

Bibliografía

- [1] Calendly. *El mejor software de programación automatizada de reuniones para usted y para todas las personas con las que se reúne*. <https://calendly.com/es/pages/features>. Online; accessed 19 de Mayo de 2020.
- [2] Doodle. *Haga que las reuniones tengan lugar Con Doodle, programar reuniones es rápido y sencillo*. <https://doodle.com/es/>. Online; accessed 19 de Mayo de 2020.
- [3] SoftExpert. *SoftExpert Reunión Gestión de reuniones corporativas*. <https://doodle.com/es/>. Online; accessed 19 de Mayo de 2020.
- [4] Edwin Mwendu. “Software Frameworks, Architectural and Design Patterns”. En: *Journal of Software Engineering and Applications* 07 (ene. de 2014), págs. 670-678. DOI: 10.4236/jsea.2014.78061.
- [5] Jimmy Lin. “What is Cloud Computing”. En: *Class lecture Presentation* (2008).
- [6] Jan Lindström. “Real time database systems”. En: *Wiley Encyclopedia of Computer Science and Engineering* (2007), págs. 1-13.
- [7] Ian Warren y col. “Push notification mechanisms for pervasive smartphone applications”. En: *IEEE Pervasive Computing* 13.2 (2014), págs. 61-71.
- [8] OS Android. “Android”. En: *Retrieved February 24* (2011), pág. 2011.
- [9] Sergio Luján-Mora. *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario, 2002.
- [10] Manuel Báez y col. *Introducción a android*. 2019.
- [11] Margaretha Ohhyver y col. “The comparison firebase realtime database and MySQL database performance using Wilcoxon signed-rank test”. En: *Procedia Computer Science* 157 (2019), págs. 396-405.

- [12] Gaurav Kumar y Pradeep Kumar Bhatia. “Impact of Agile methodology on software development process”. En: *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* 2.4 (2012), págs. 46-50.
- [13] ORA. *Kanban vs Scrum vs Scrumban: ¿Cuales son las diferencias?* <https://ora.pm/es/blog/scrum-vs-kanban-vs-scrumban>. Online; accessed 19 de Mayo de 2020.
- [14] Alejandro Villa Betancur y Jorge E Giraldo Plaza. “Automatización de pruebas unitarias de códigos PHP”. En: *Scientia et technica* 2.50 (2012), págs. 147-151.
- [15] Sam McLellan, Andrew Muddimer y S Camille Peres. “The effect of experience on System Usability Scale ratings”. En: *Journal of usability studies* 7.2 (2012), págs. 56-67.
- [16] Grady Booch. *The unified modeling language user guide*. Pearson Education India, 2005.
- [17] Helen C. Purchase y col. “Comprehension of Diagram Syntax: An Empirical Study of Entity Relationship Notations”. En: *Int. J. Hum.-Comput. Stud.* 61.2 (ago. de 2004), 187–203. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2004.01.003. URL: <https://doi.org/10.1016/j.ijhcs.2004.01.003>.

ANEXOS

A. Flujo de trabajo de la aplicación web

A.1. Carga de datos masiva de cursos, estudiantes y docentes.

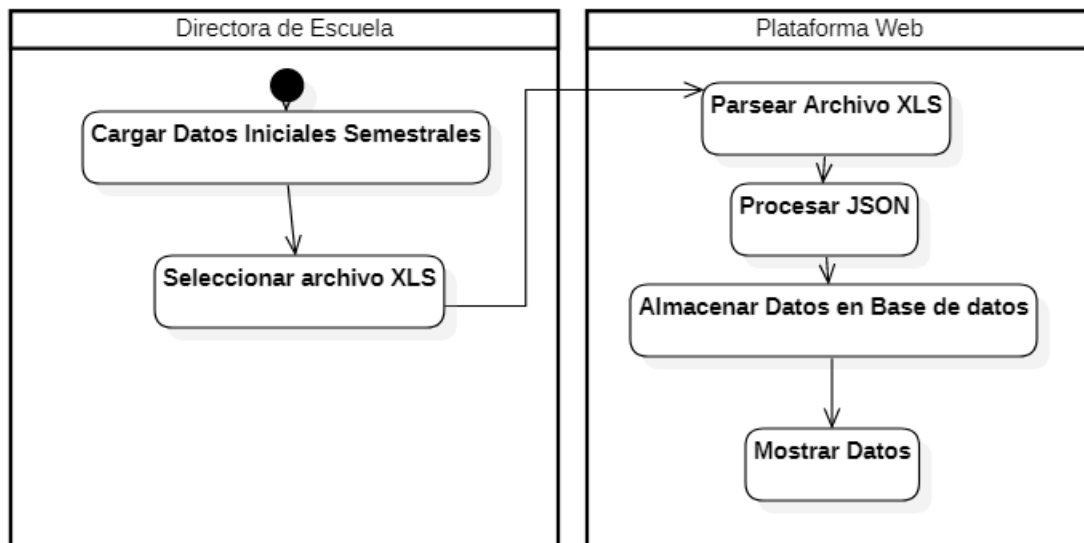


Figura A.1: Flujo para realizar la carga de datos masiva de cursos, docentes y estudiantes

A.2. Cálculo de horas propuestas de ayudantías

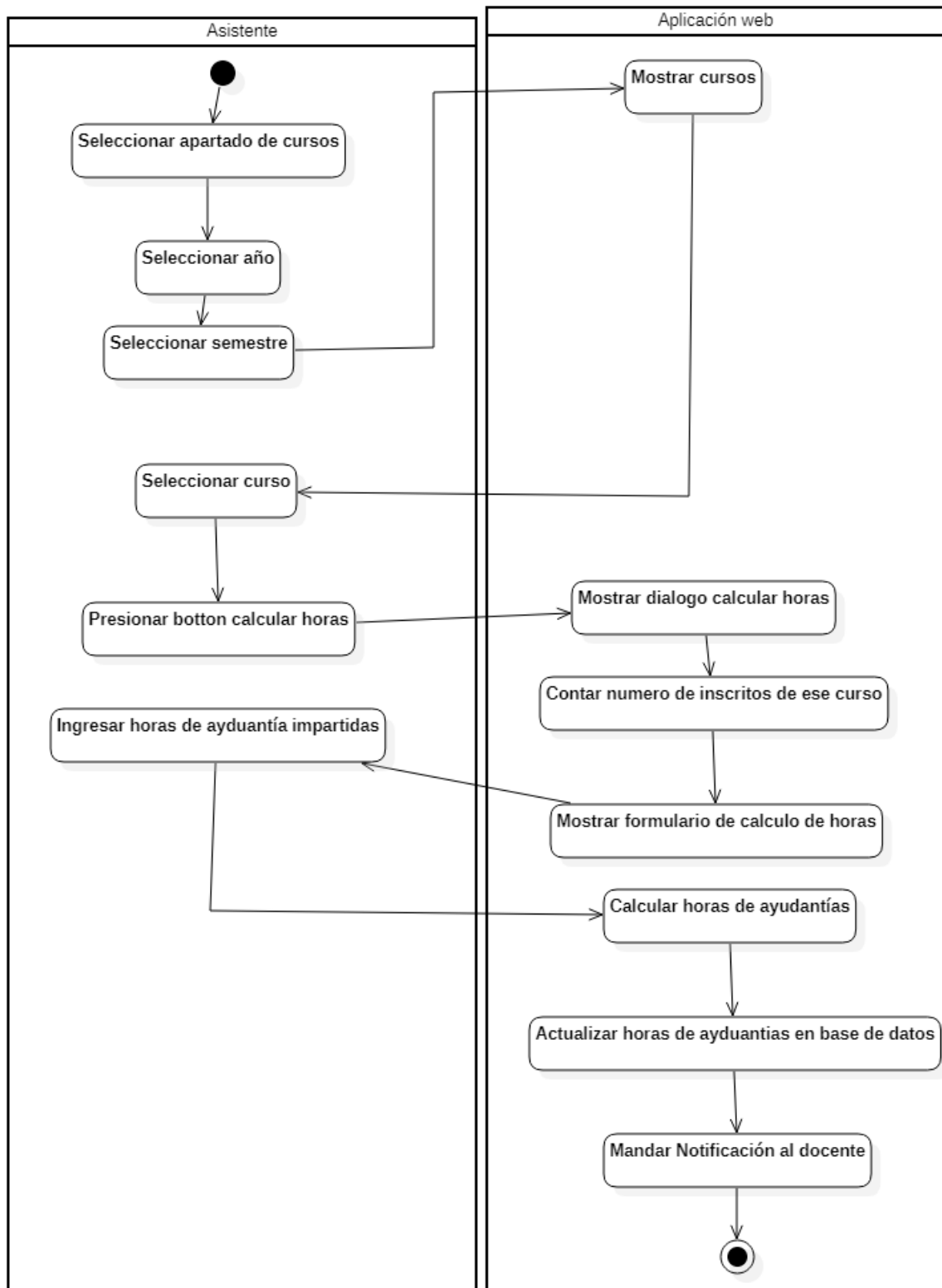


Figura A.2: Flujo para realizar el cálculo de horas propuestas de ayudantías

A.3. Asignación de estudiantes a docente tutor

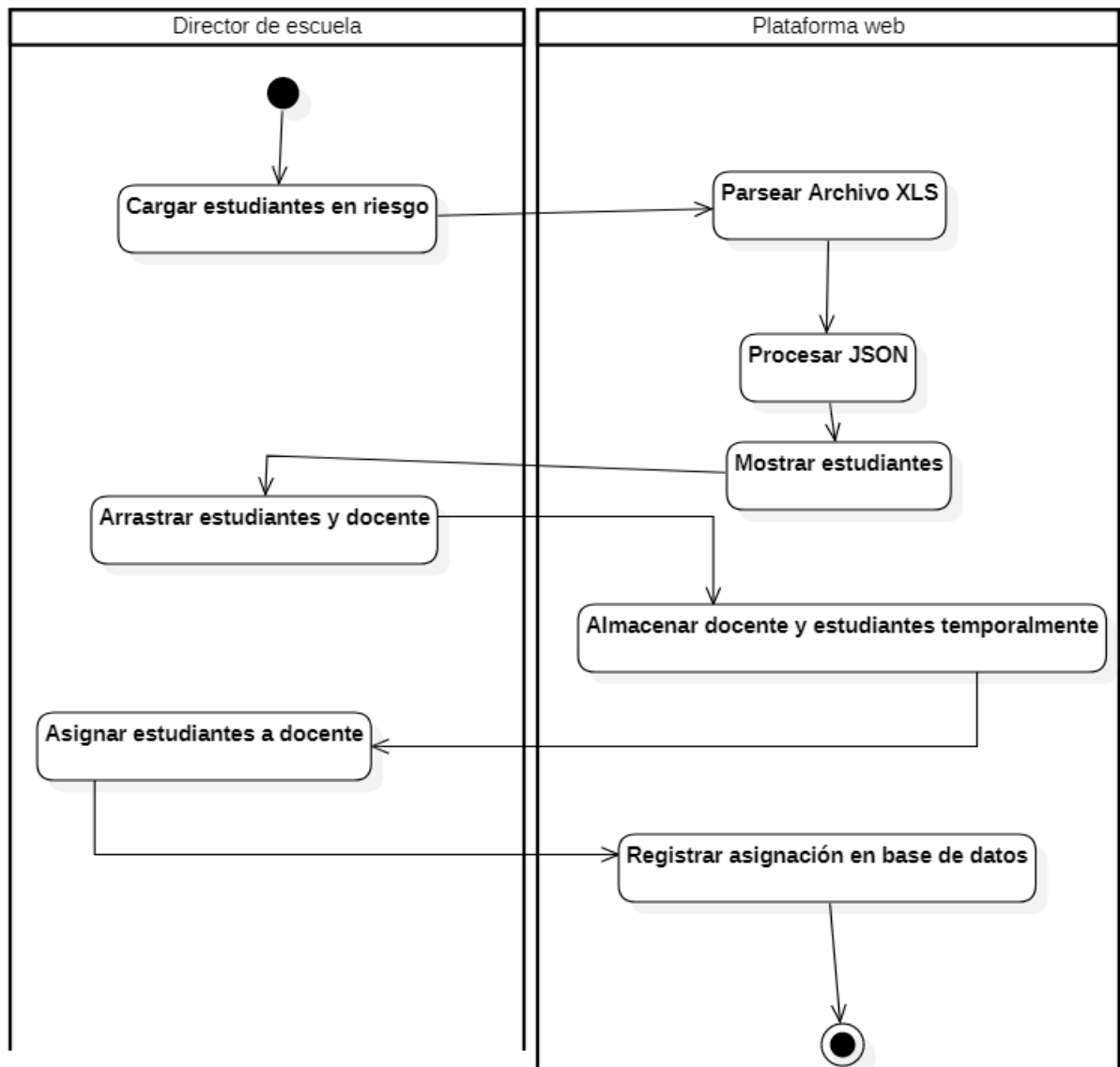


Figura A.3: Flujo que se realiza para la asignación de estudiantes a docentes

A.4. Asignación de estudiantes a tutor estudiantil

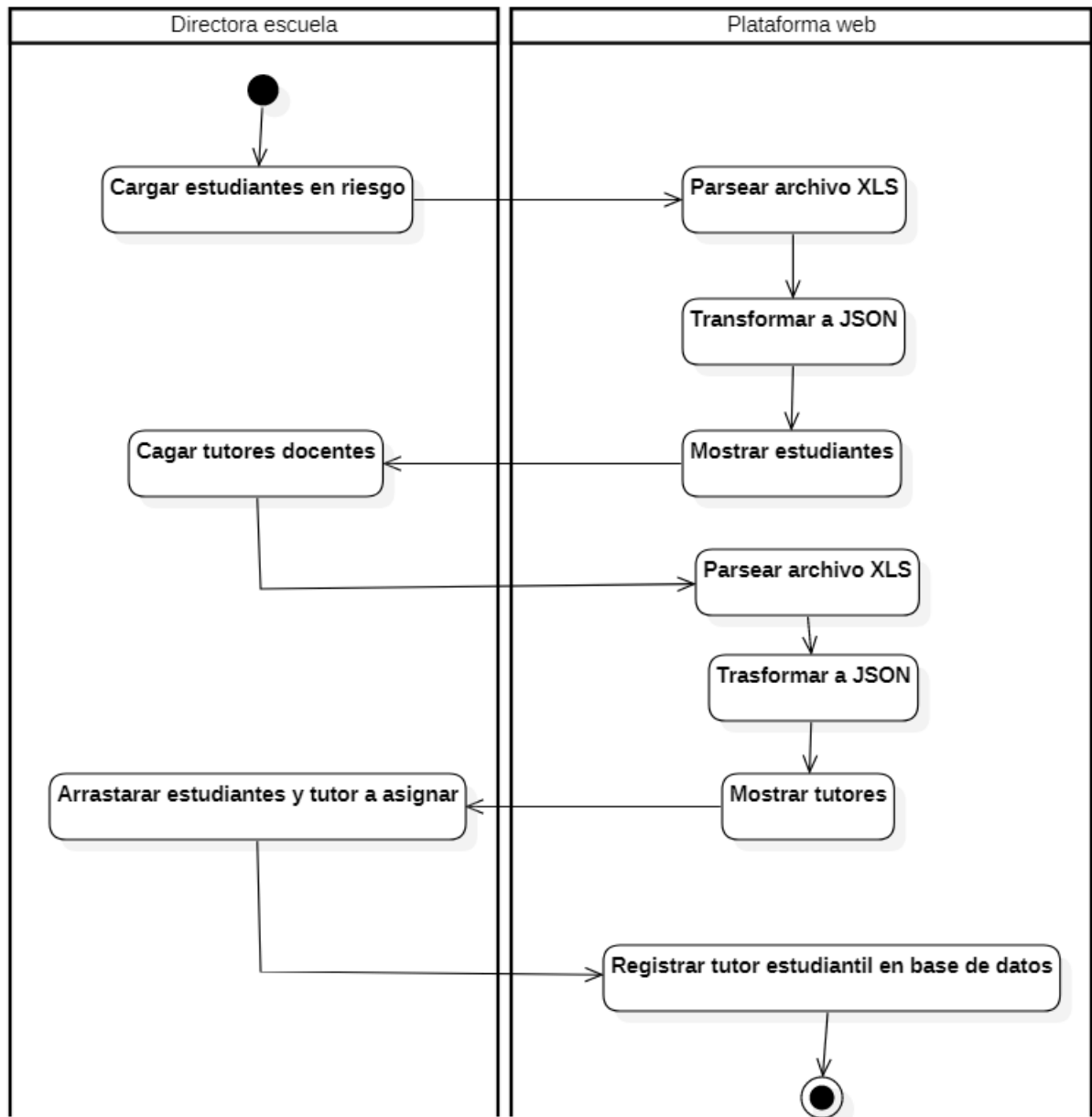


Figura A.4: Flujo que se realiza para la asignación de estudiantes a tutores estudiantiles

B. Flujo de trabajo de la aplicación móvil

B.1. Activación de cuenta de estudiante

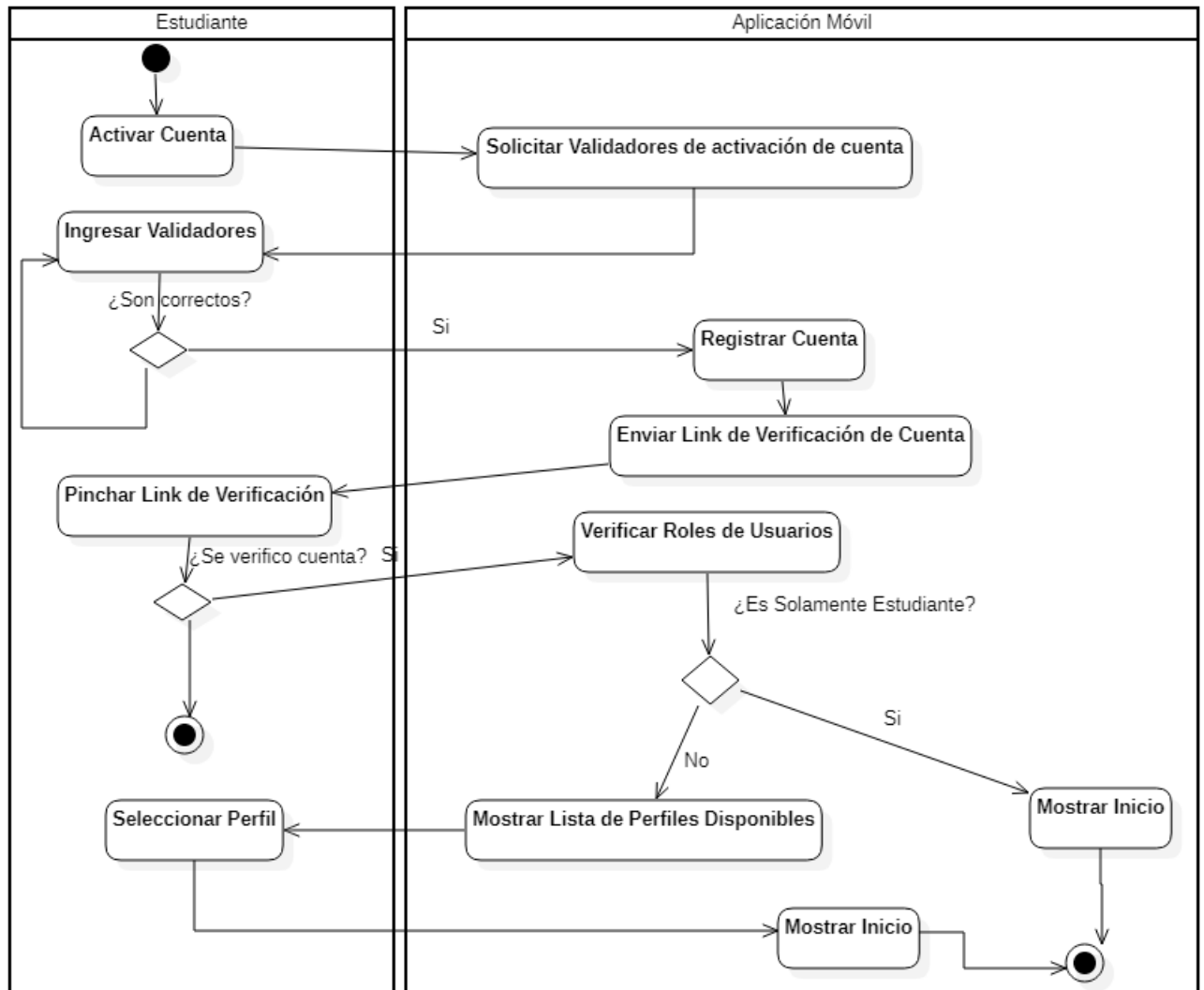


Figura B.1: Flujo que se realiza para activar una cuenta de estudiante en la aplicación móvil

B.2. Postulación a una oferta de ayudantía

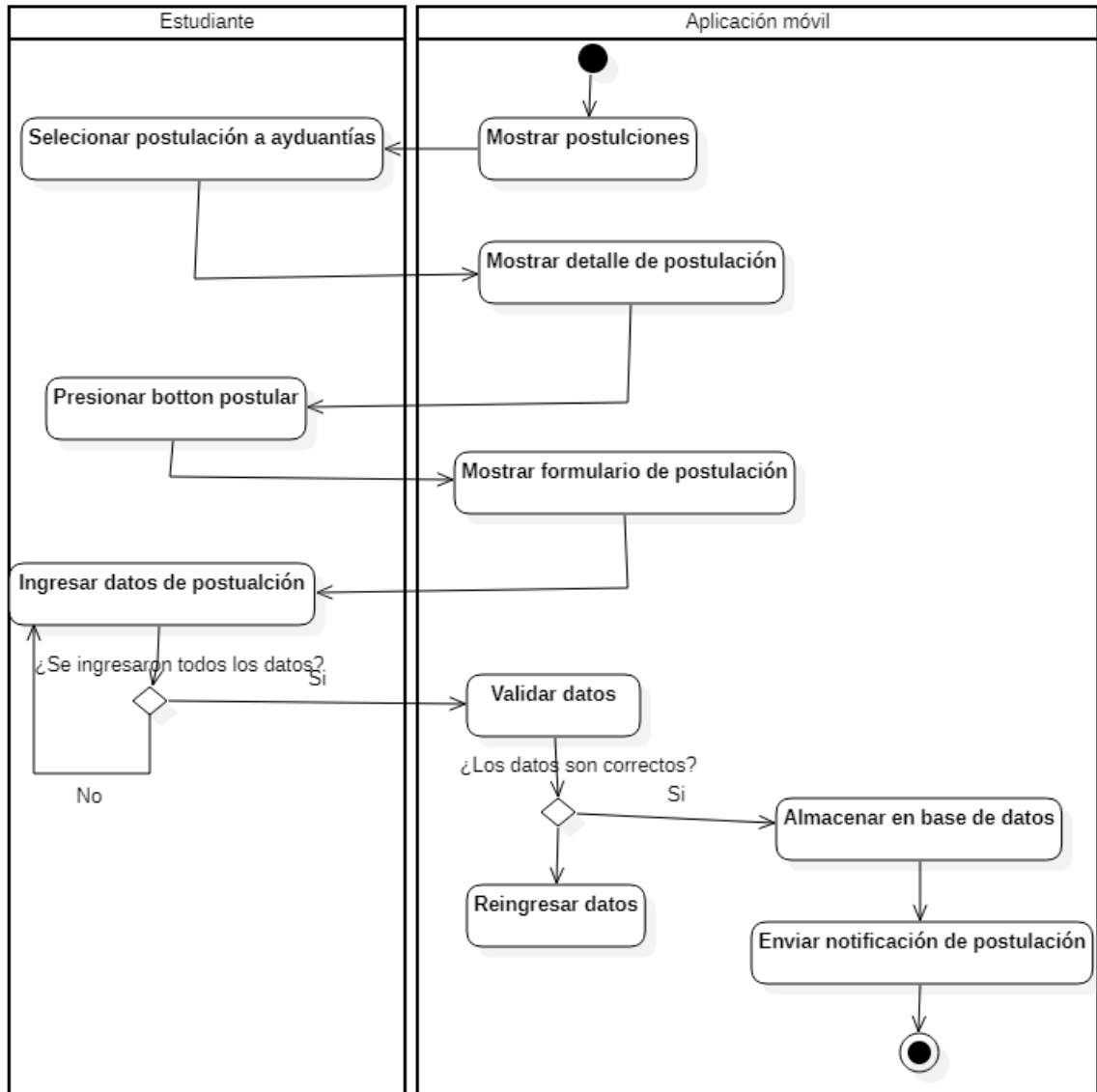


Figura B.2: Flujo que se realiza para postular a una oferta de ayudantía

B.3. Selección de un ayudante

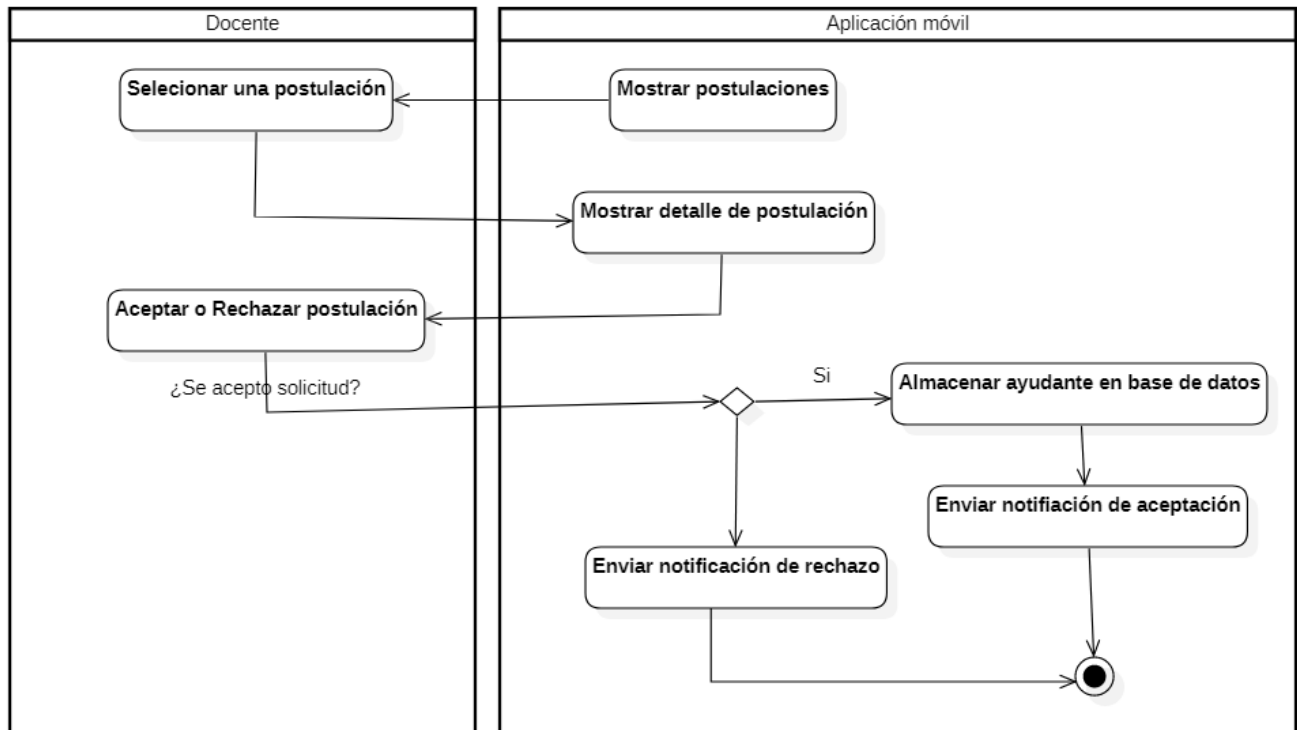


Figura B.3: Flujo que se realiza para seleccionar a un ayudante

B.4. Evaluar un ayudante

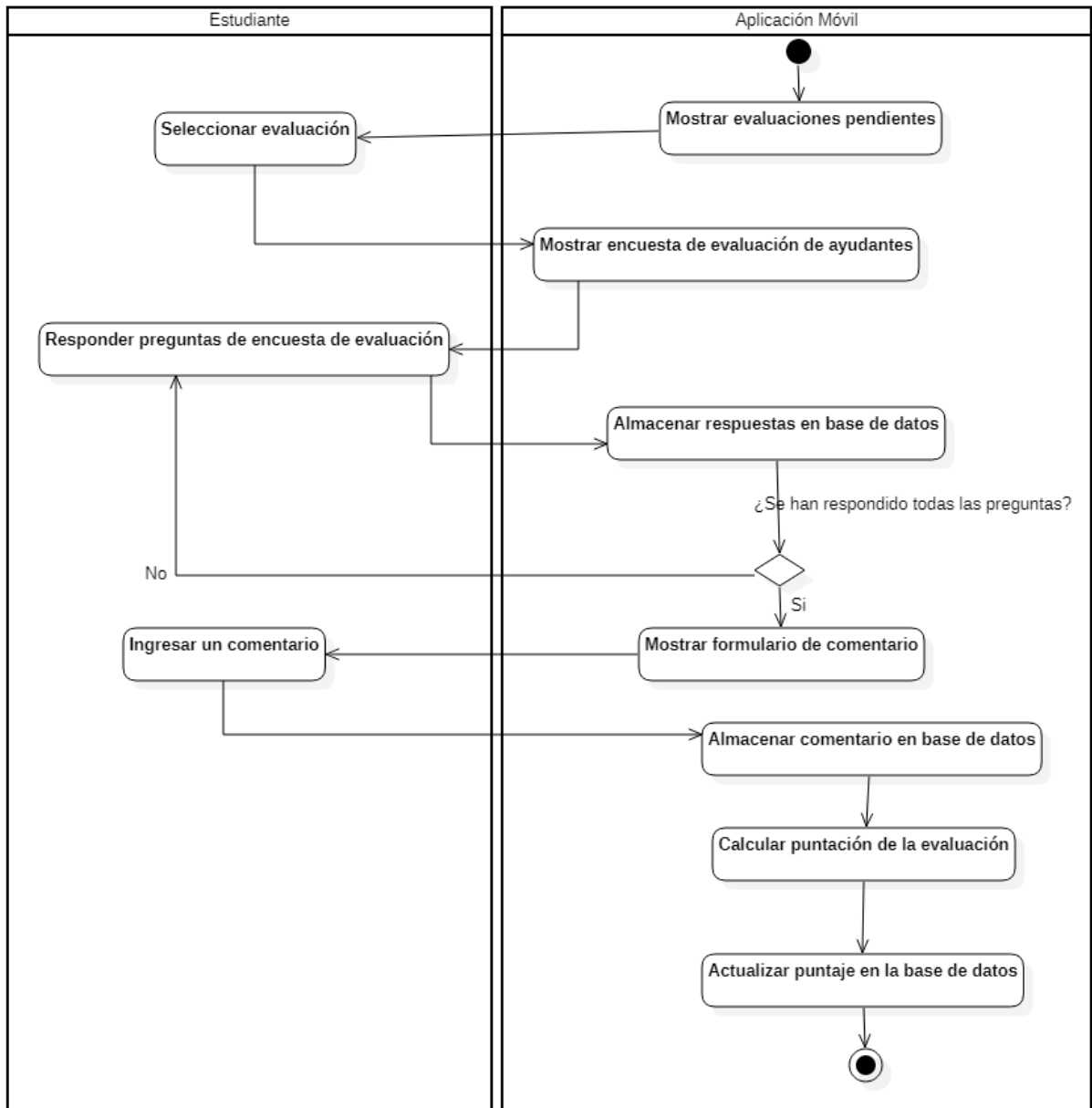


Figura B.4: Flujo de realizado para evaluar un ayudante

B.5. Crear una reunión de ayudantías

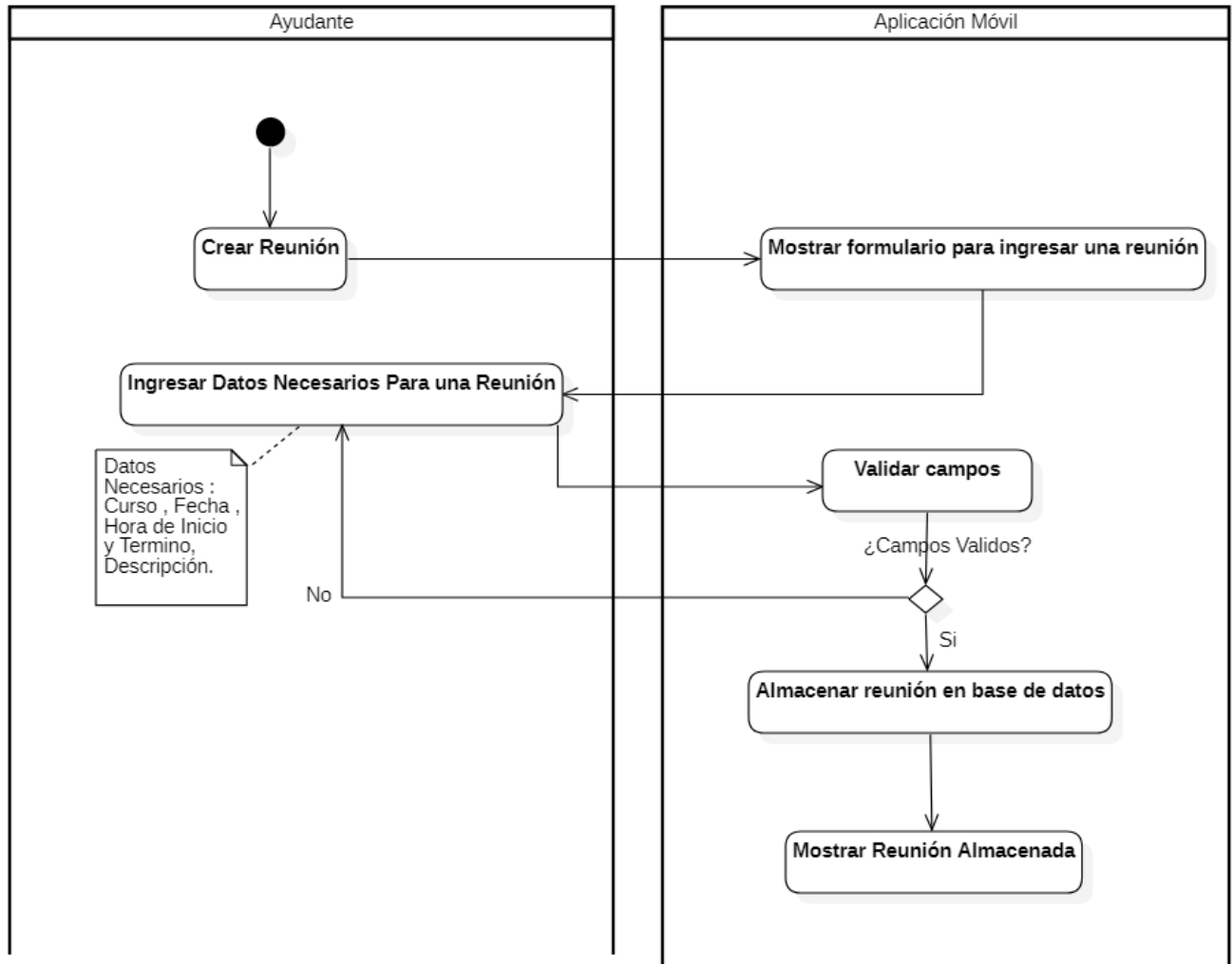


Figura B.5: Flujo que se realiza para crear reunión de ayudantía

B.6. Registrar asistencia de ayudantías

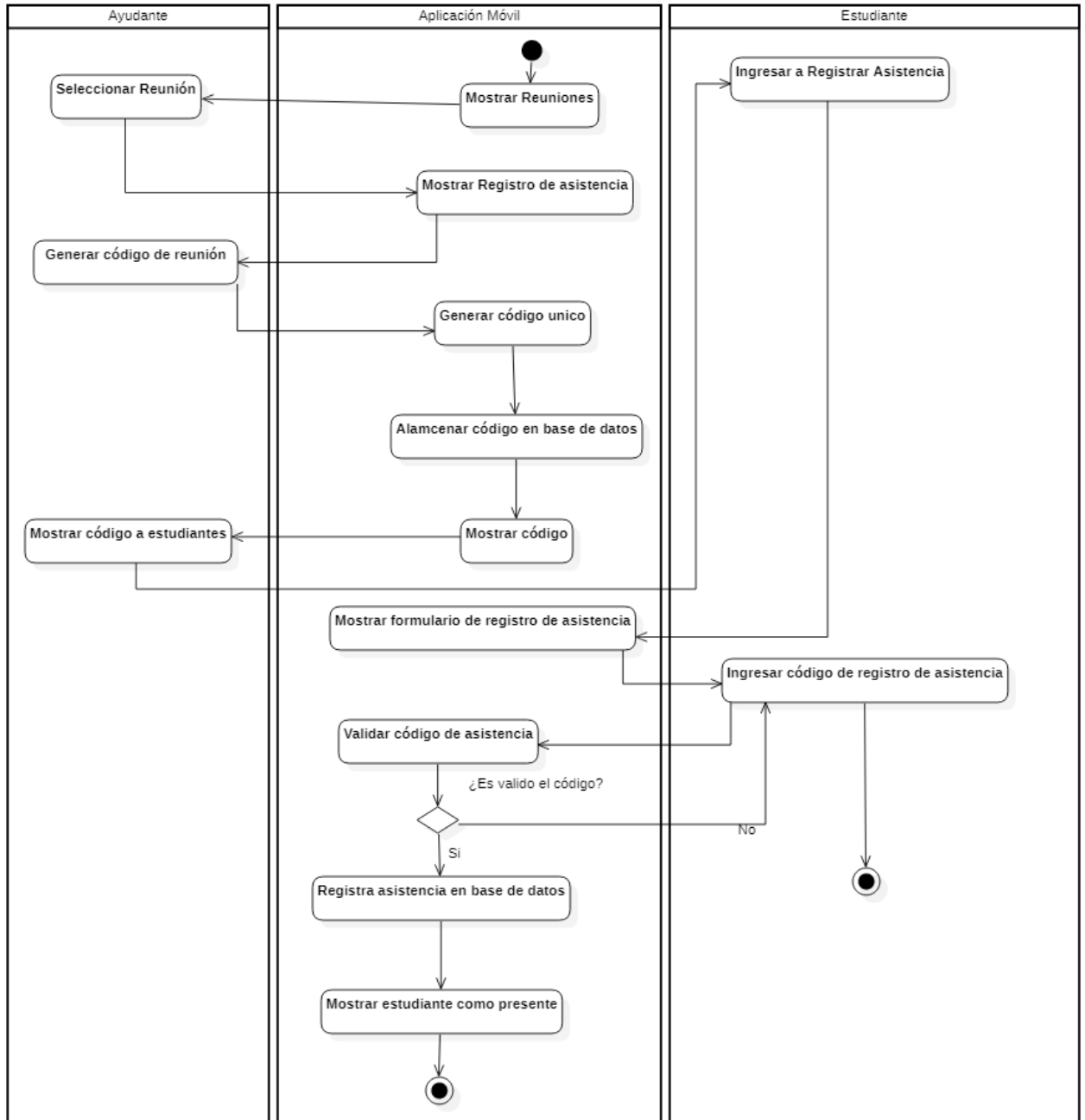


Figura B.6: Flujo que se realiza para registro de asistencia reunión de ayudantía

B.7. Agendar reuniones con estudiantes tutor docente

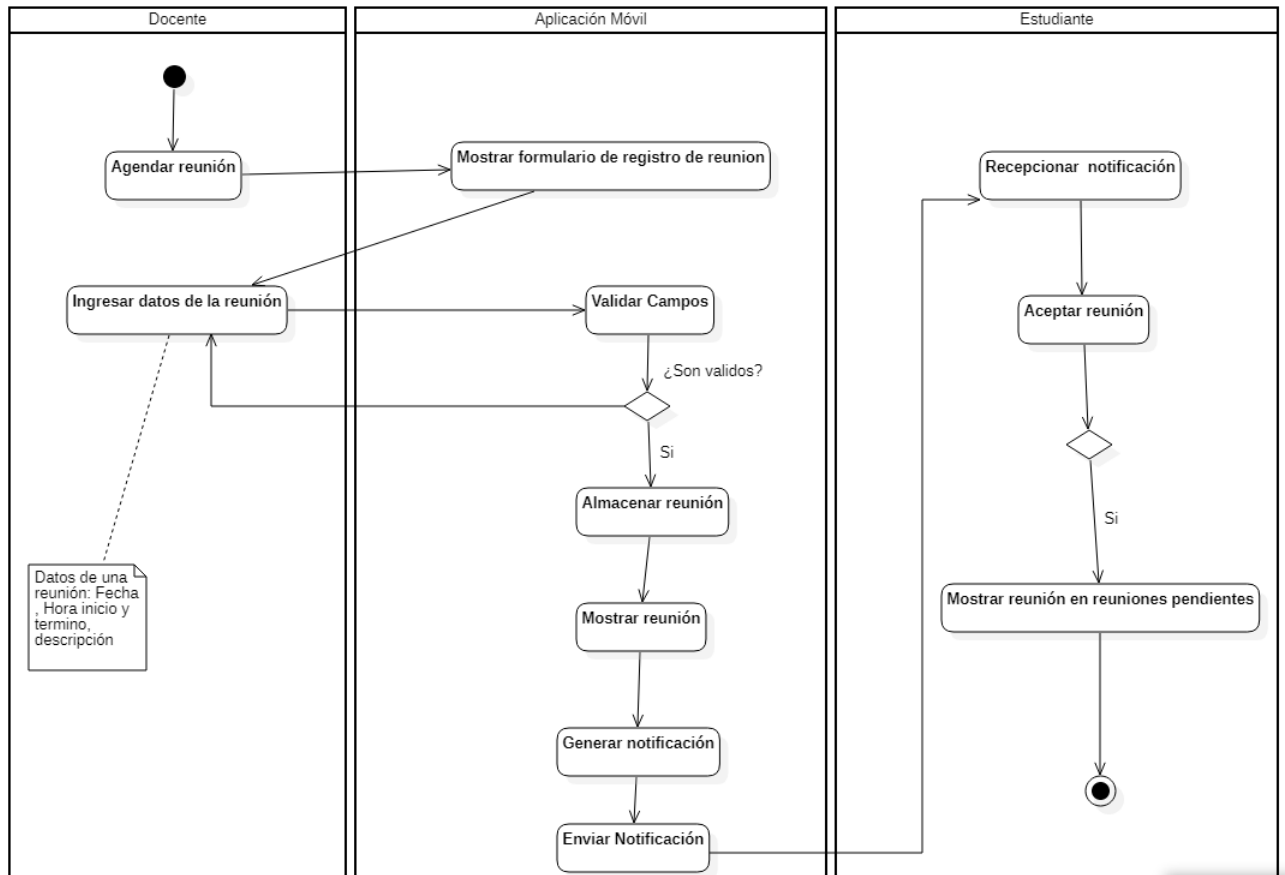


Figura B.7: Flujo que se realiza para agendar reuniones de tutorías docentes

B.8. Registrar anotaciones de sesiones

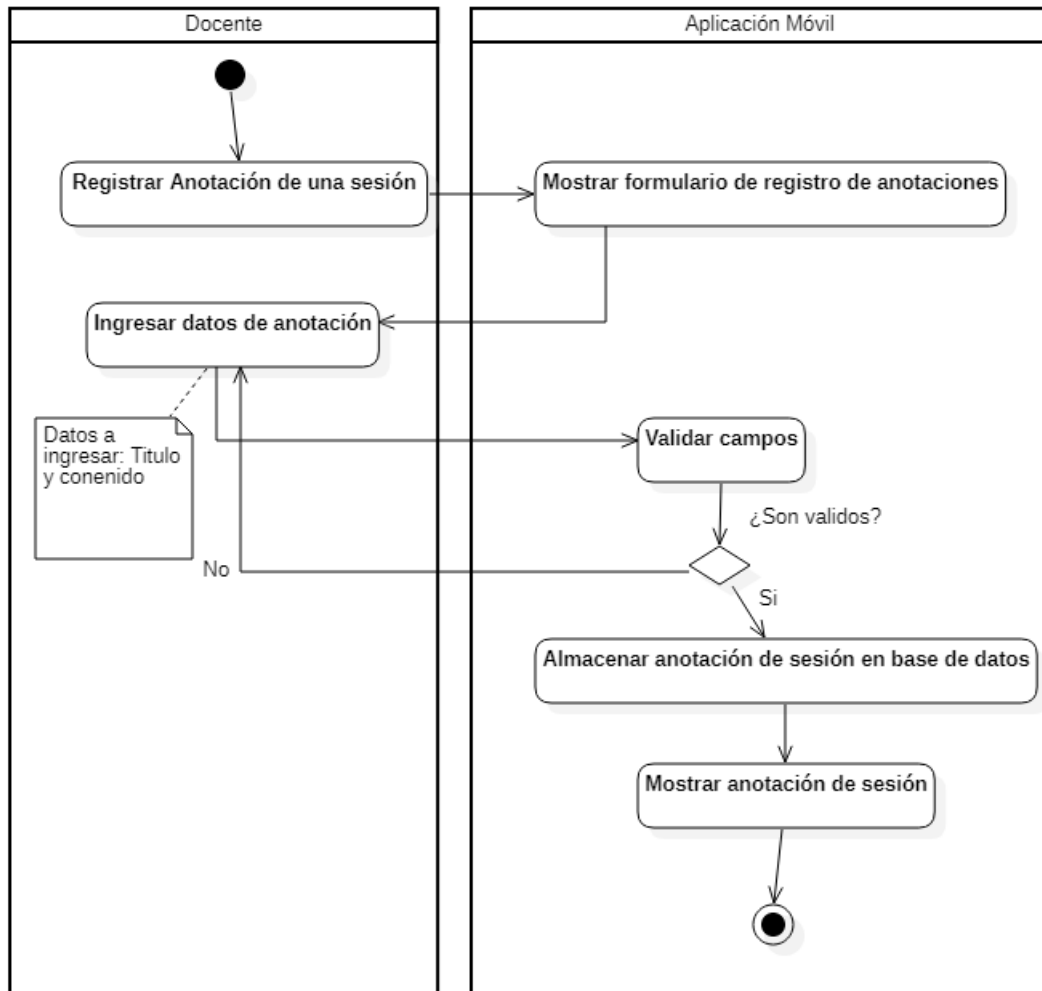


Figura B.8: Flujo para registrar una anotación de una sección

B.9. Reportar el avance que tuvo el estudiante durante el semestre

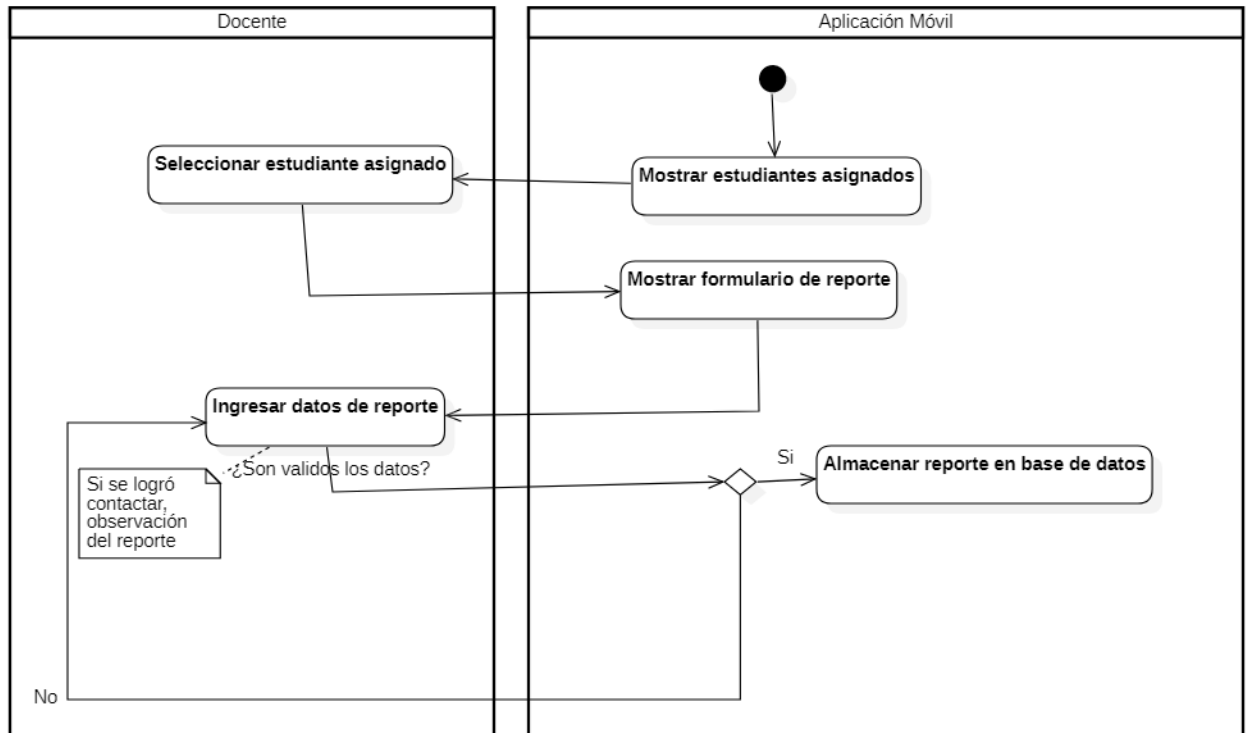


Figura B.9: Flujo que se realiza para reportar el avance que obtuvo el estudiante durante el semestre

B.10. Crear planificación para tutorías estudiantiles

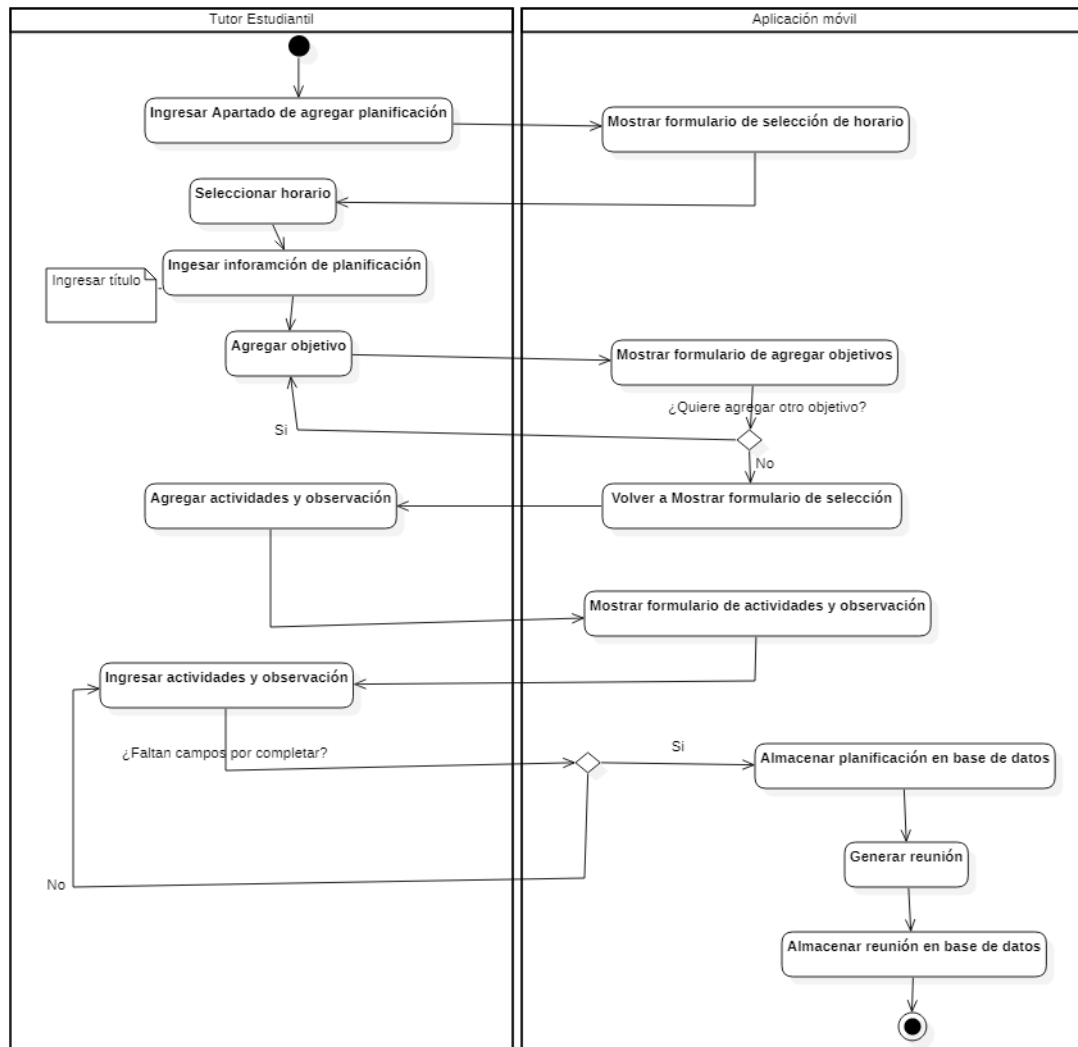


Figura B.10: Flujo que se realiza para realizar una planificación para un tutoría estudiantil

C. Diagrama de clases de la aplicación móvil

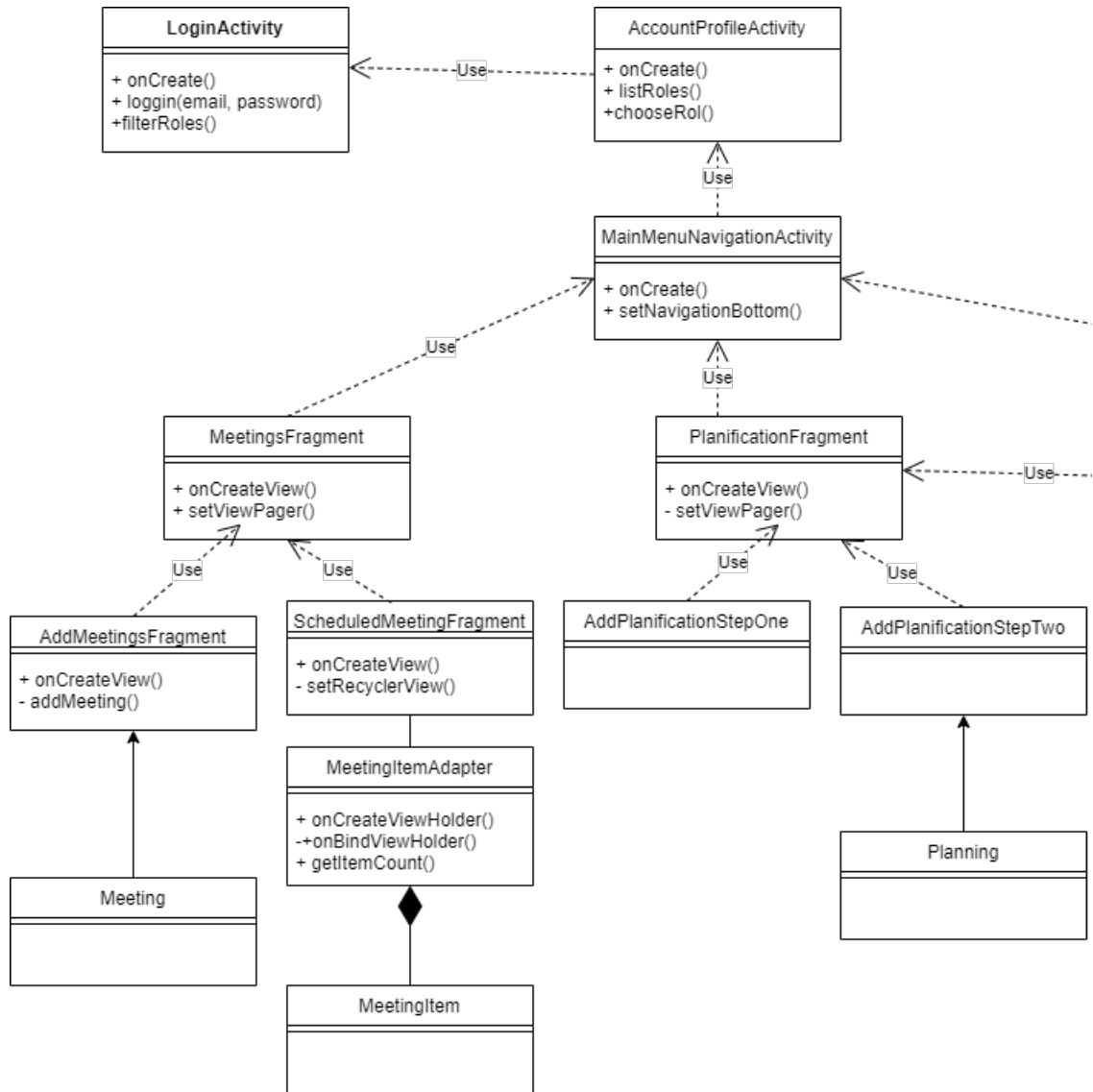


Figura C.1: Parte del diagrama de clases del rol de ayudante para el agenda de reuniones y planificación de ayudantías

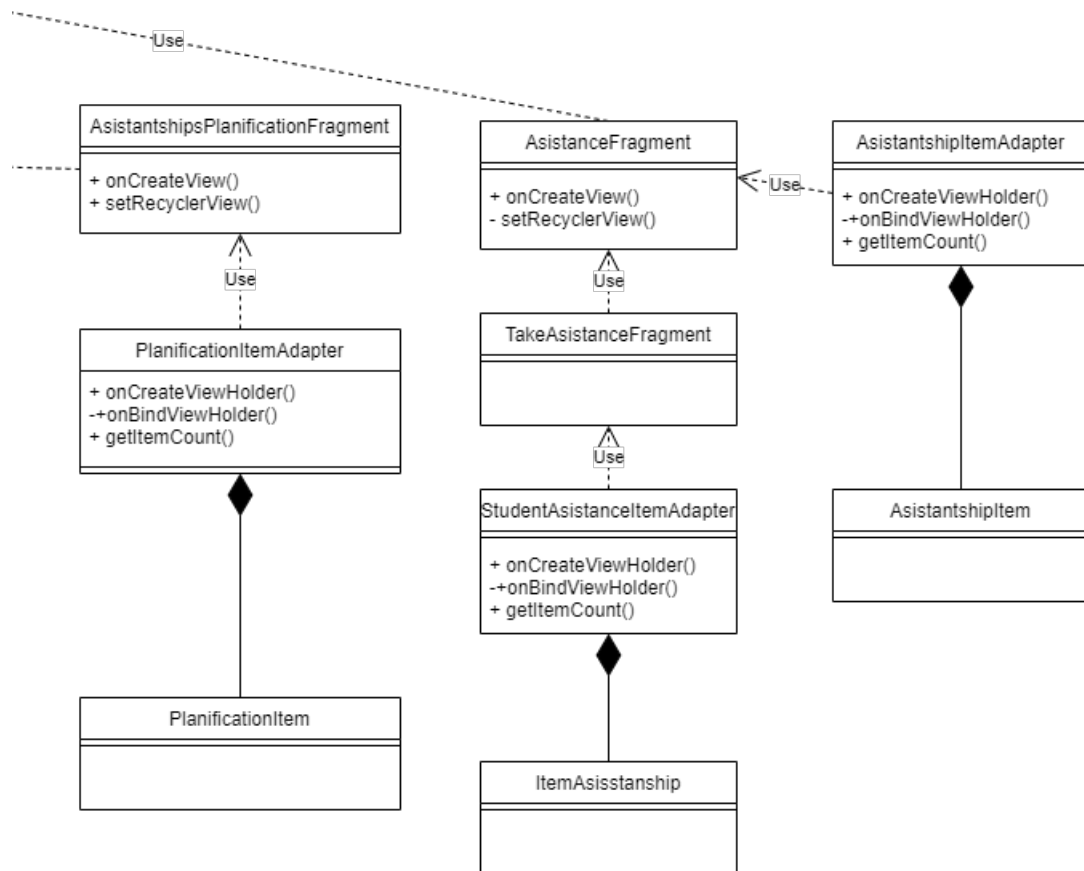


Figura C.2: Parte del diagrama de clases del rol de ayudante para el registro de asistencia de ayudantías

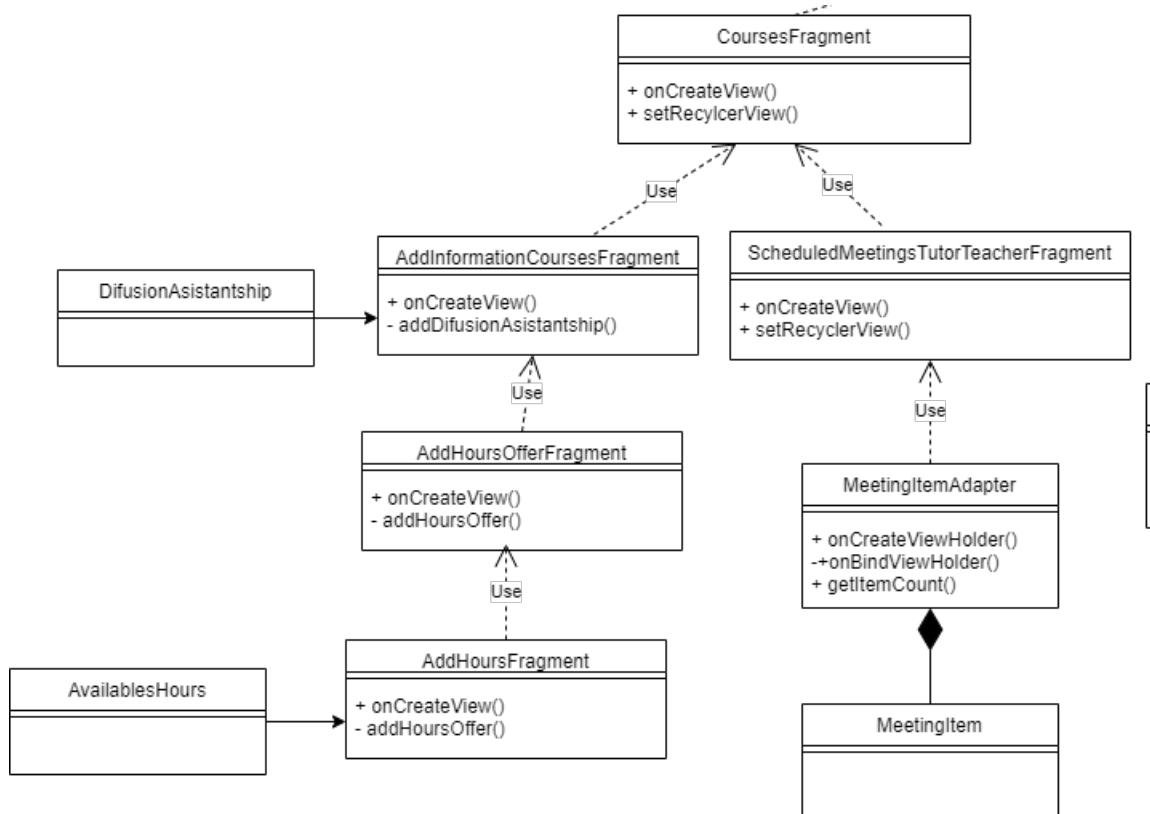


Figura C.3: Parte del diagrama de clases del rol de docente visualización de cursos y difusión de ayudantías

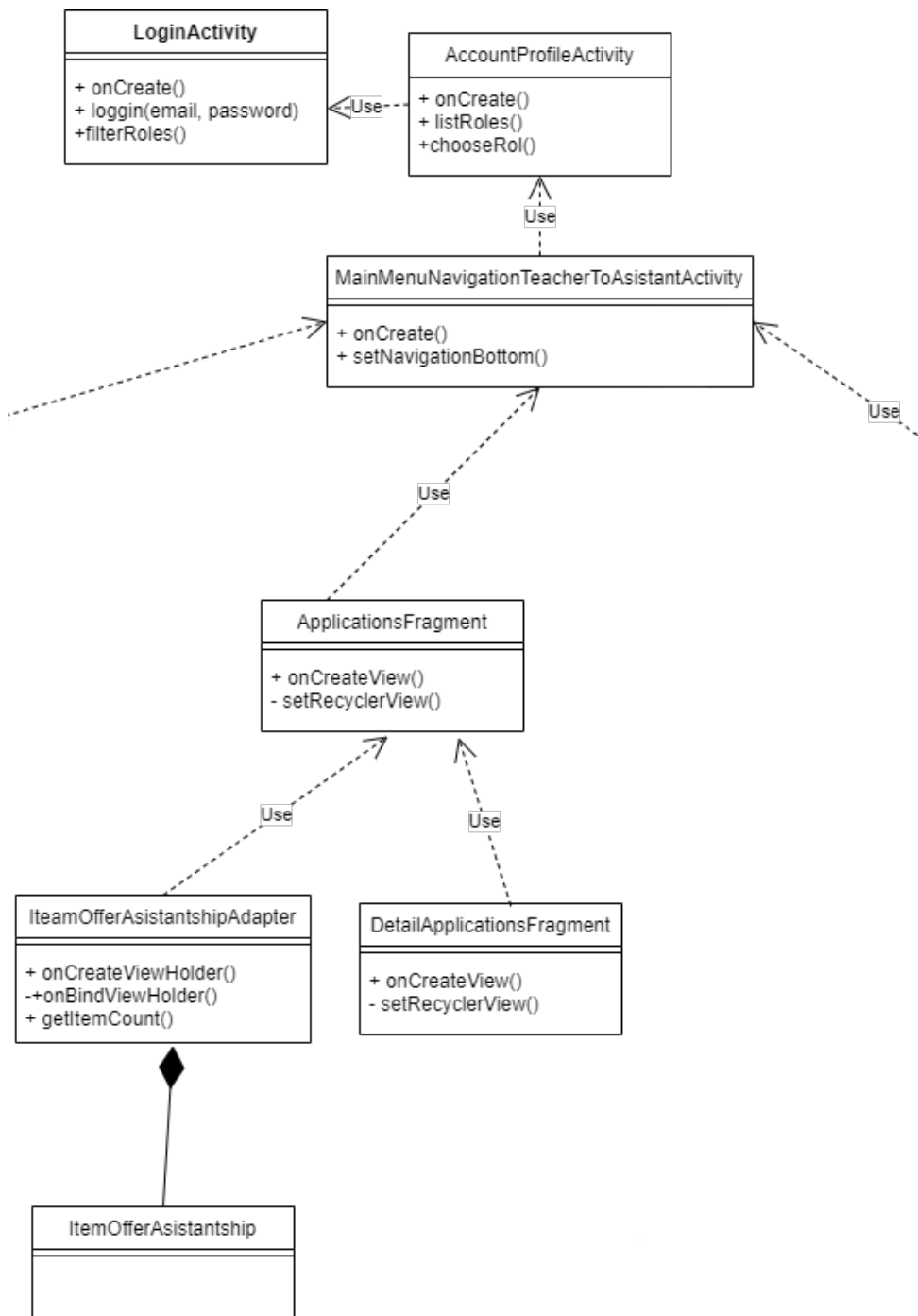


Figura C.4: Parte del diagrama de clases del rol de docente visualización de postu-
laciones y selección de ayudantes

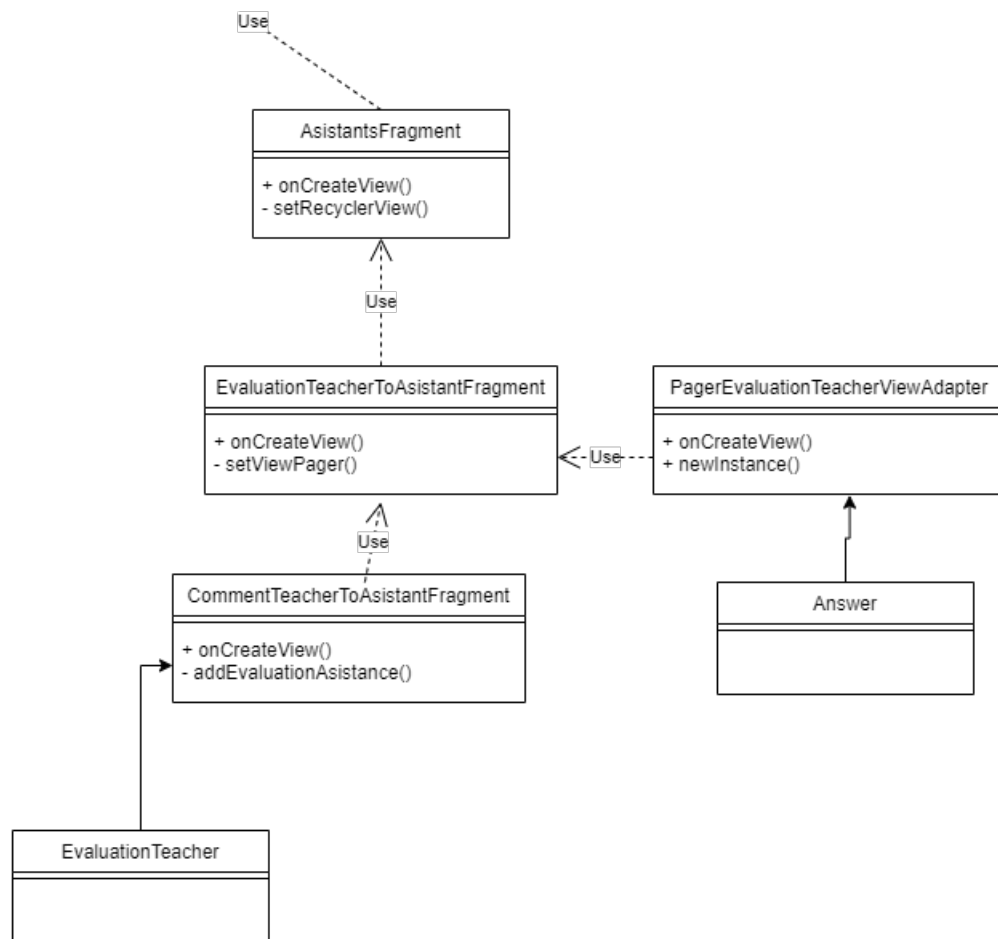


Figura C.5: Parte del diagrama de clases del rol de docente visualización de ayudantes y evaluación

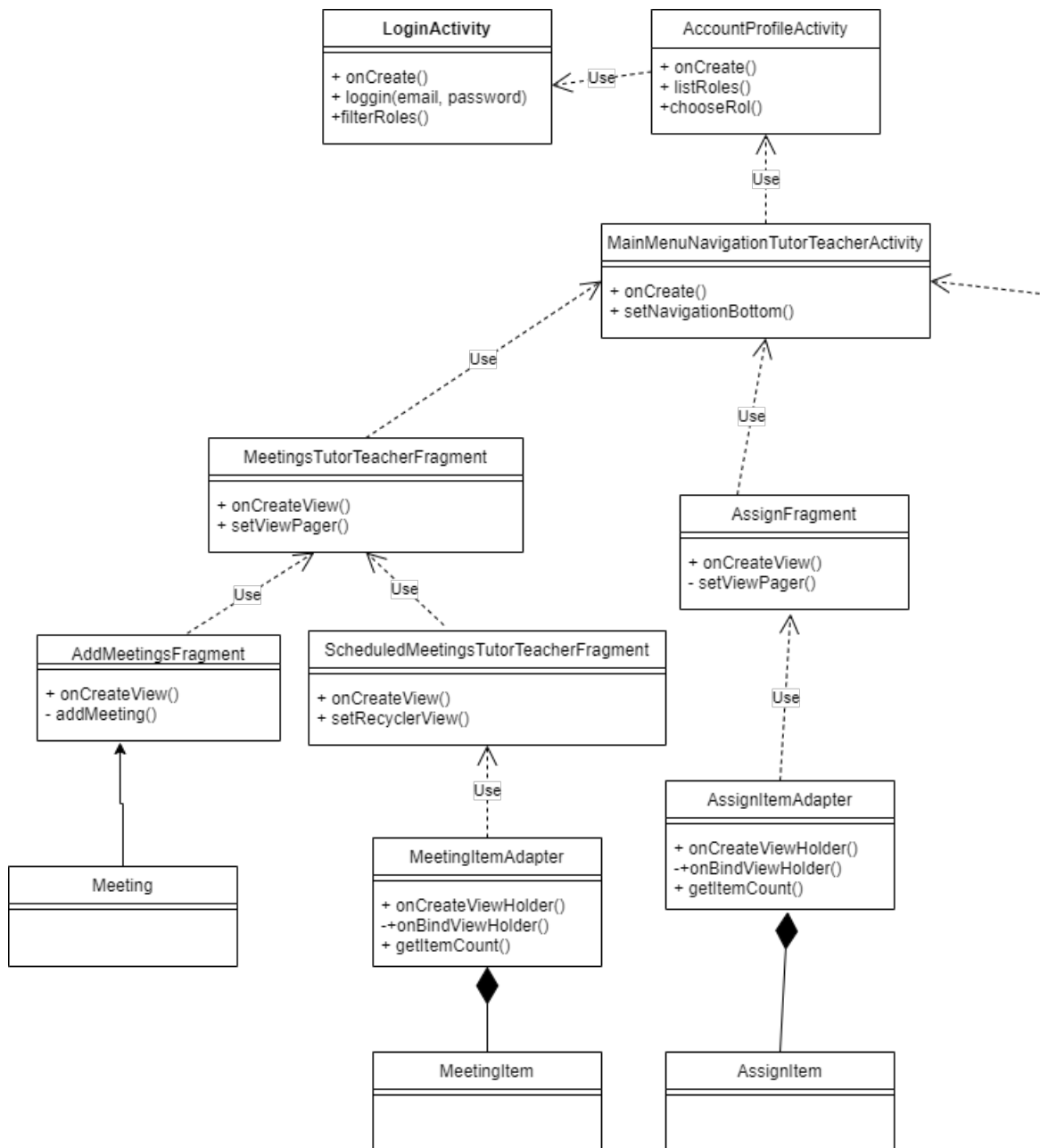


Figura C.6: Parte del diagrama de clases del rol de tutor docente visualización de asignaciones y reuniones con estudiantes

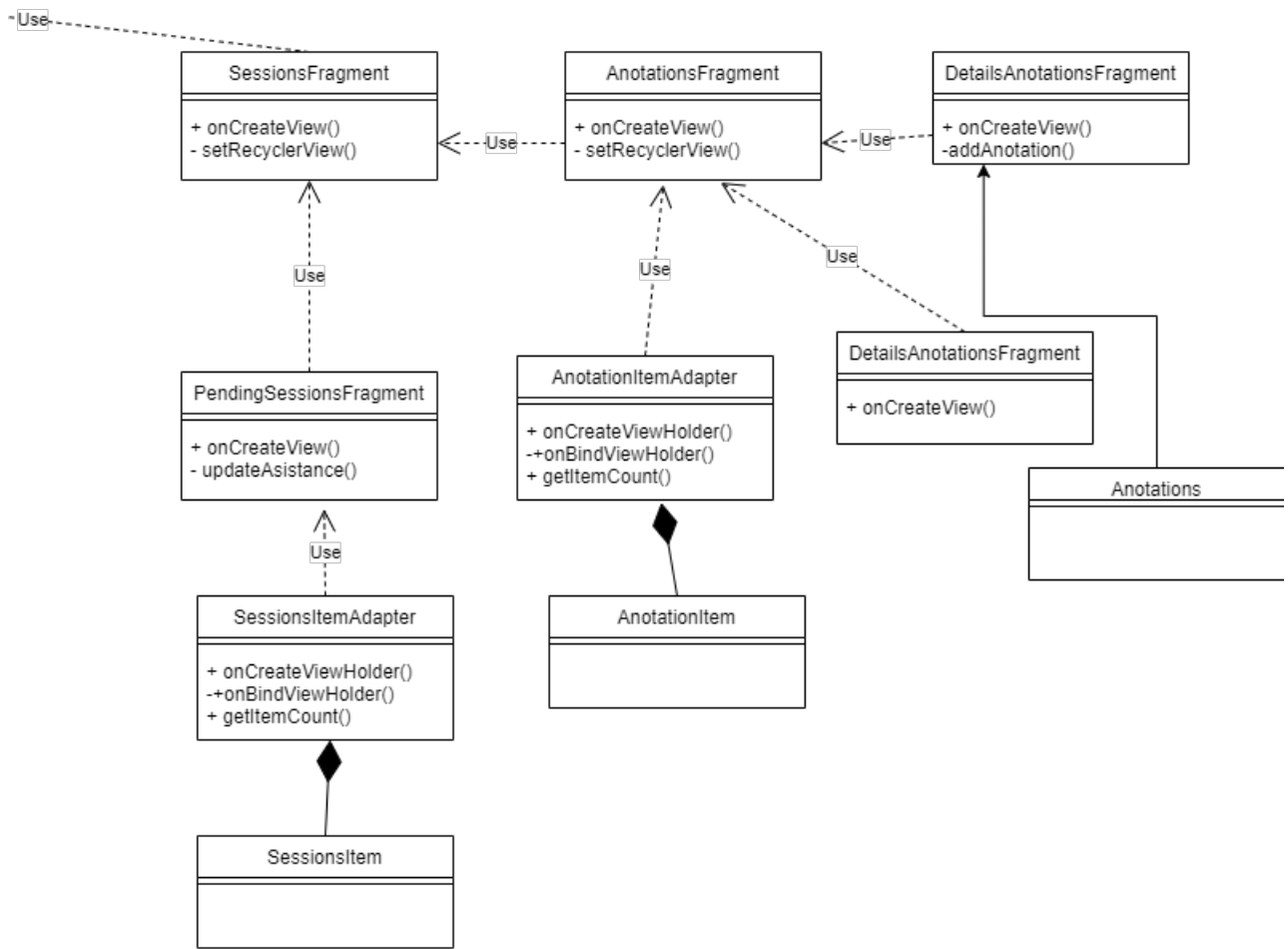


Figura C.7: Parte del diagrama de clases del rol de tutor docente sesiones y anotaciones con estudiantes

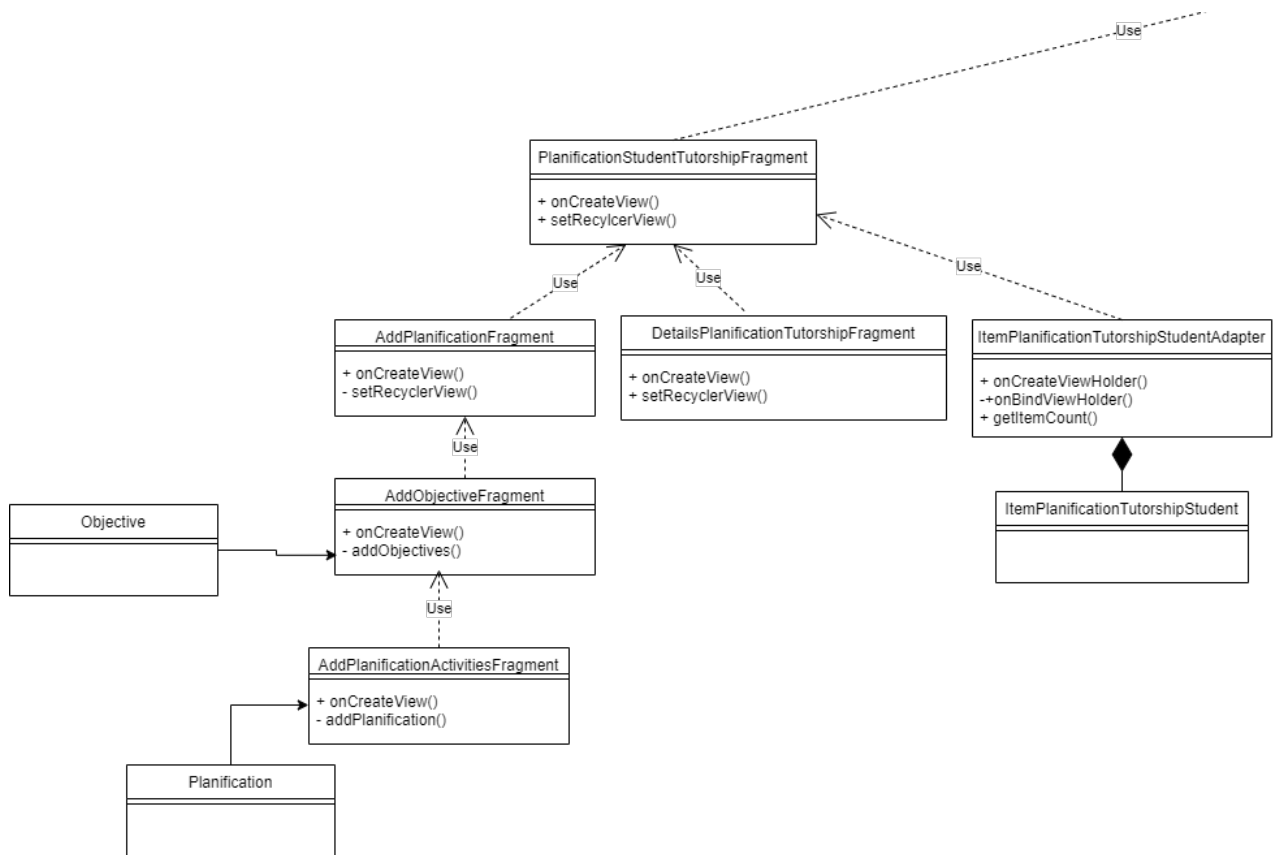


Figura C.8: Parte del diagrama de clases del rol de tutor estudiantil planificaciones de tutorías estudiantiles

D. Navegación de la aplicación web

D.1. Navegación de apartado de ayudantías

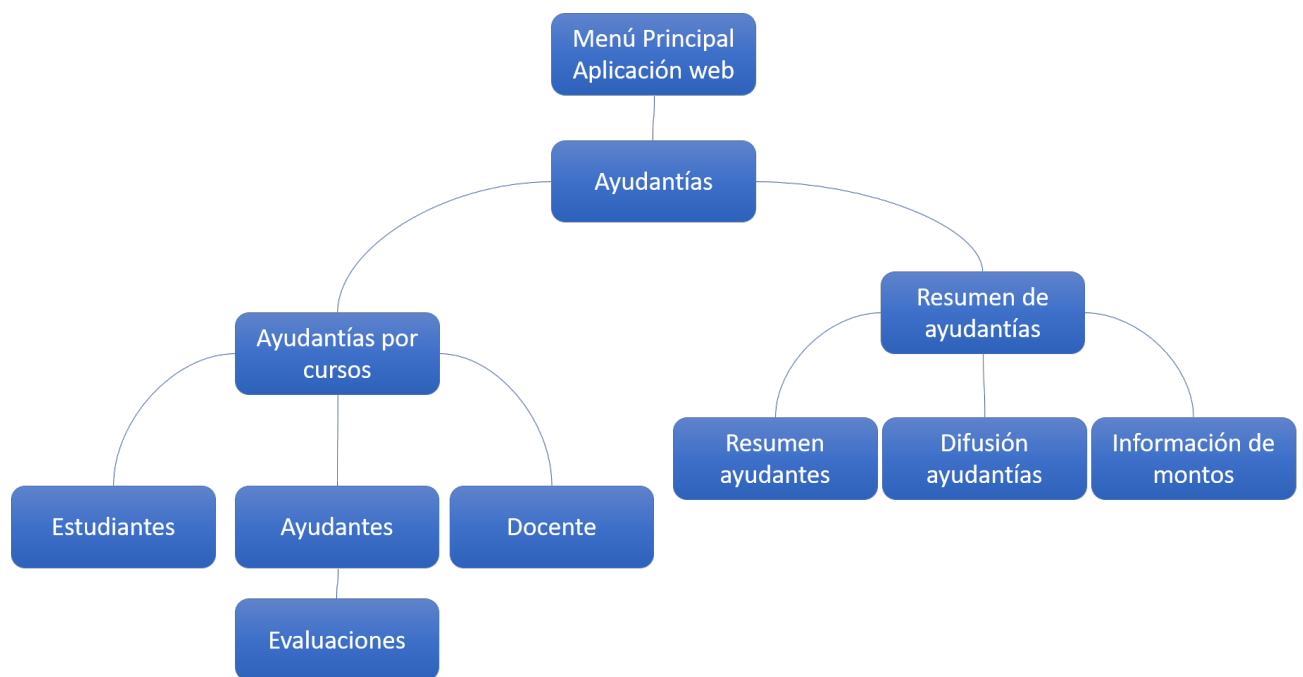


Figura D.1: Navegación entre ventanas en el apartado de ayudantías

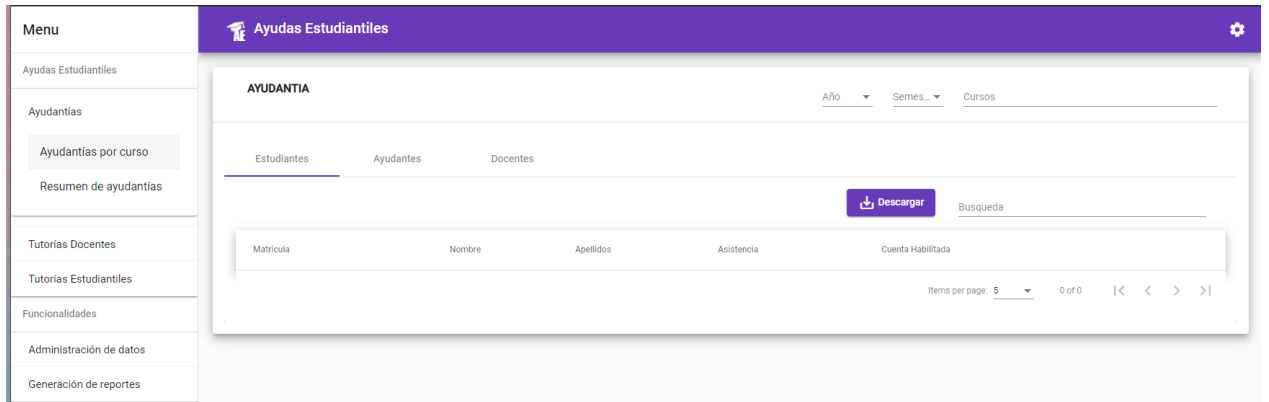


Figura D.2: Vista de ayudantías por curso

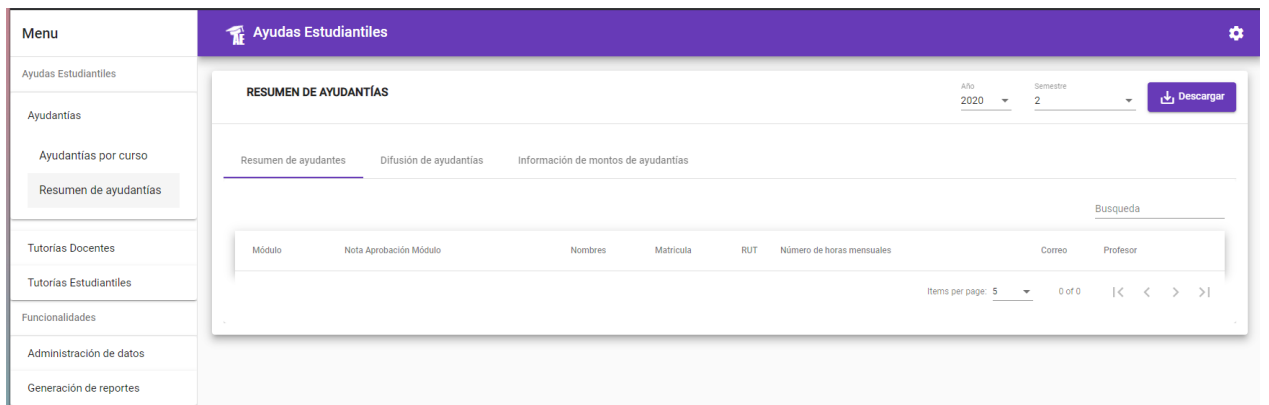


Figura D.3: Vista de resumen de ayudantías

D.2. Navegación de tutorías docentes

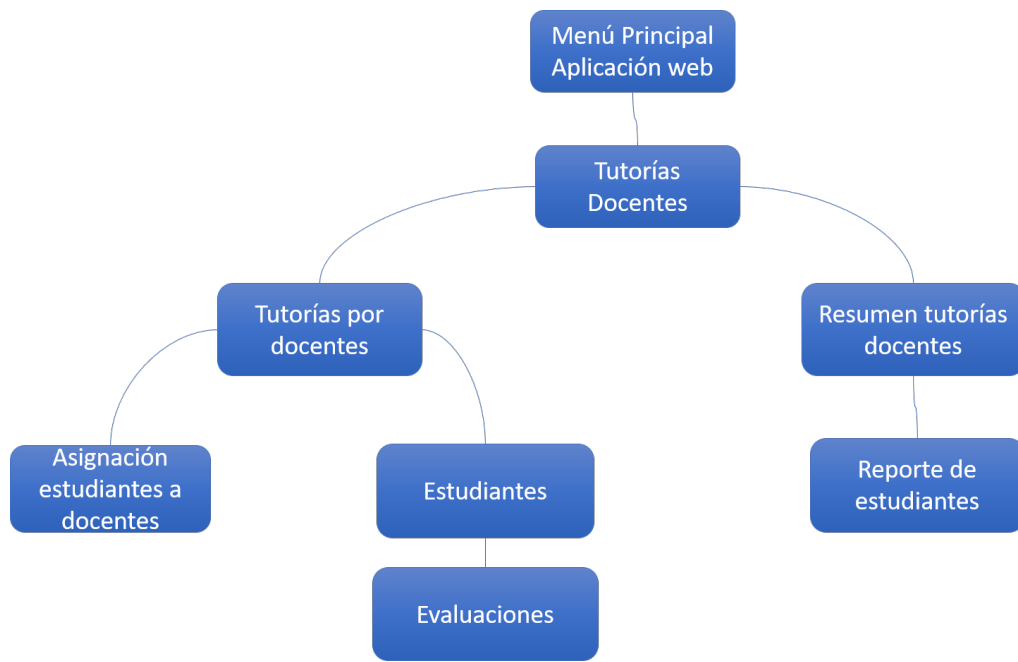


Figura D.4: Navegación entre ventanas en el apartado de tutorías docentes

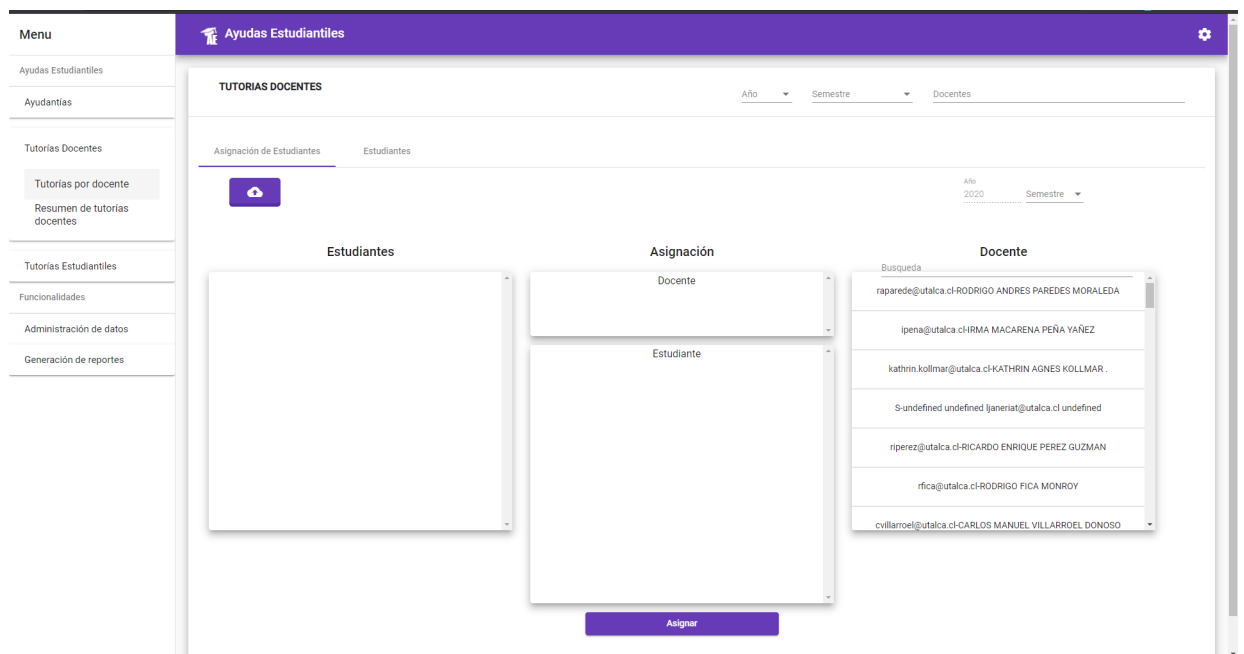


Figura D.5: Vista de tutorías por docente

Menu

- Ayudas Estudiantiles
- Ayudantías
- Tutorías Docentes
 - Tutorías por docente
 - Resumen de tutorías docentes
- Tutorías Estudiantiles
- Funcionalidades
- Administración de datos
 - Estudiantes
 - Docentes
 - Cursos
- Generación de reportes

TUTORIAS DOCENTES

Año: 2020 Semestre: 2 Docentes: salvestre@utaica.cl

Asignación de Estudiantes Estudiantes

Ver Evaluación

Matricula	Nombre
2015407023	DIEGO MATUS CAMPOS

Items per page: 5 0 of 0 |< < > >|

Figura D.6: Vista de estudiantes de tutorías docentes

Menu

- Ayudas Estudiantiles
- Ayudantías
- Tutorías Docentes
 - Tutorías por docente
 - Resumen de tutorías docentes
- Tutorías Estudiantiles
- Funcionalidades
- Administración de datos
- Generación de reportes

RESUMEN DE TUTORIAS DOCENTES

Año Semestre Descargar

Resumen de reportes tutorías docentes

Busqueda

Matricula Estudiante	Nombre Estudiante	Nombre Profesor a cargo	¿Logró contactarse?	Observación
----------------------	-------------------	-------------------------	---------------------	-------------

Items per page: 5 0 of 0 |< < > >|

Figura D.7: Vista de resumen de tutorías docentes

D.3. Navegación de tutorías estudiantiles

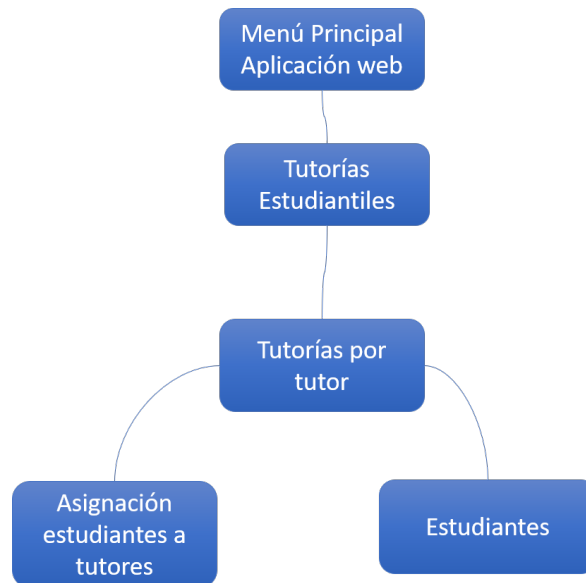


Figura D.8: Navegación entre ventanas en el apartado de tutorías estudiantiles

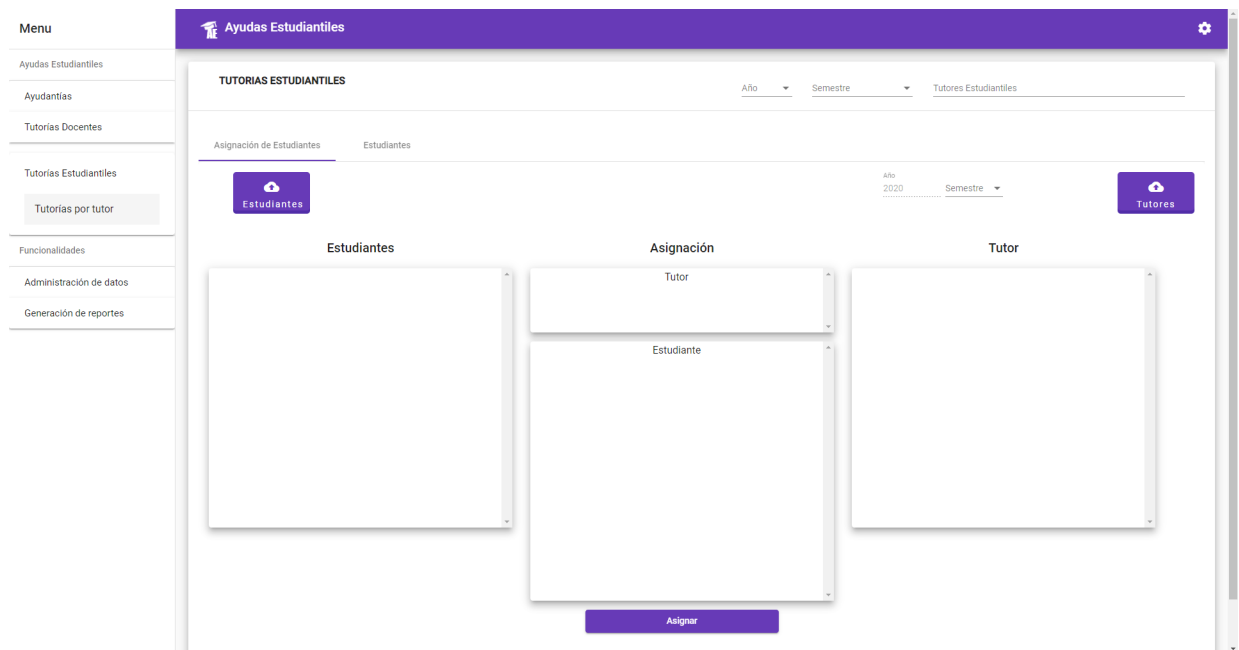


Figura D.9: Vista de asignación de estudiantes de tutores estudiantiles

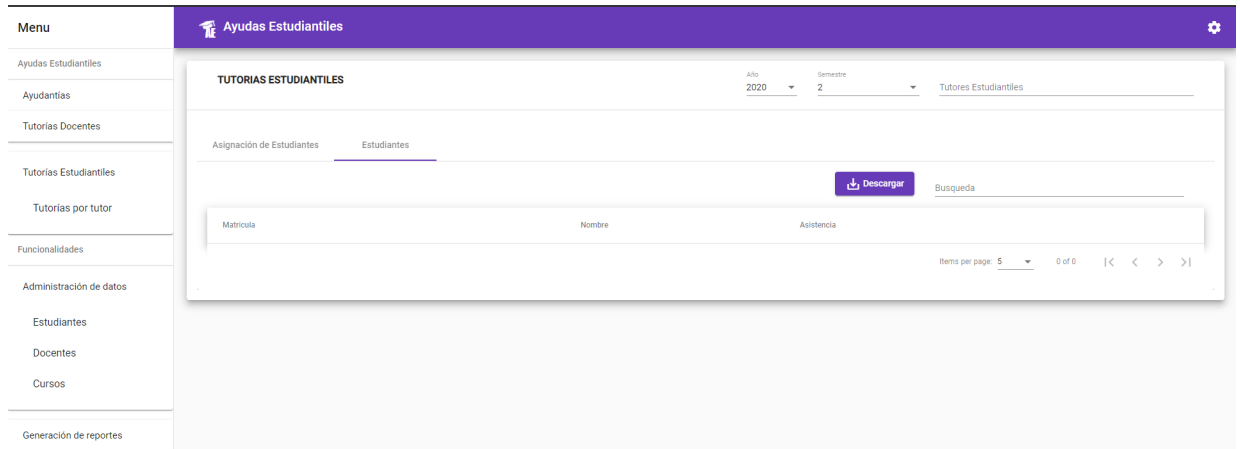


Figura D.10: Vista de estudiantes en el aparato de tutorías estudiantiles

E. Navegación de la aplicación móvil

E.1. Navegación del perfil de ayudante

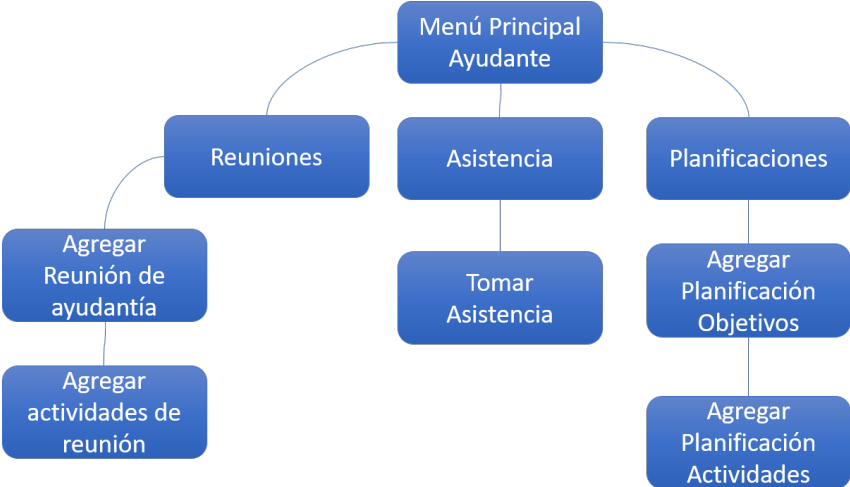


Figura E.1: Navegación del menú de ayudantes



Figura E.2: Vistas de agregar reunión y tomar asistencia de una ayudantía

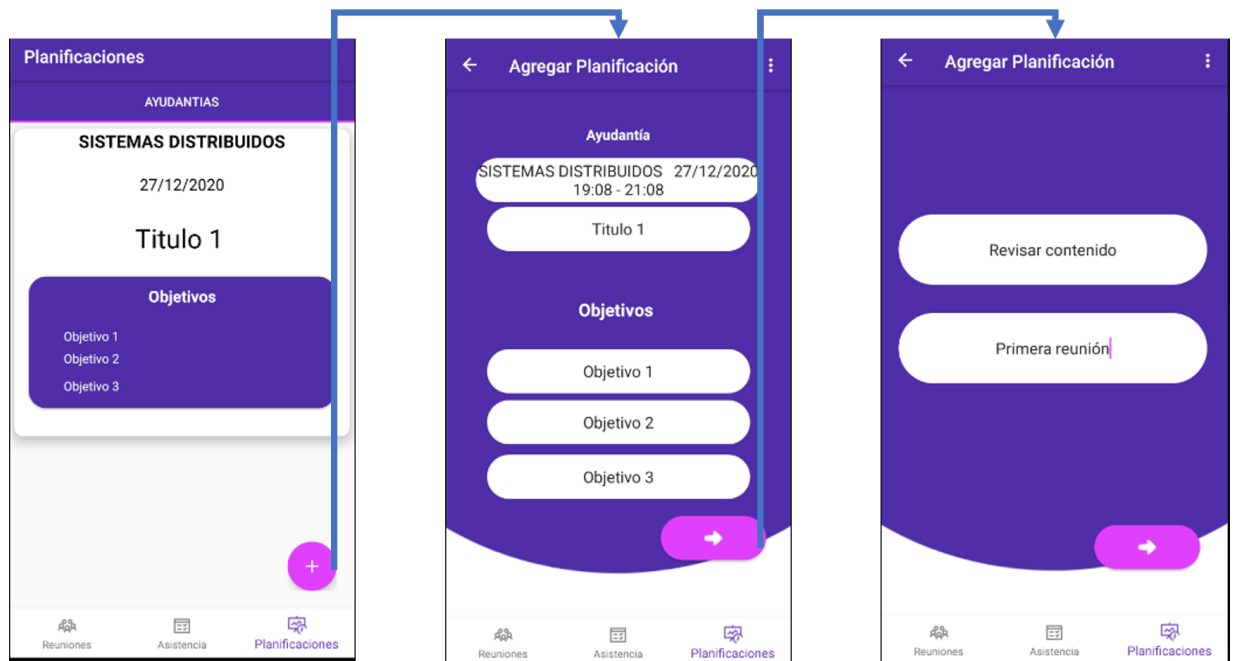


Figura E.3: Vistas para agregar una planificación de ayudantías

E.2. Navegación perfil estudiante

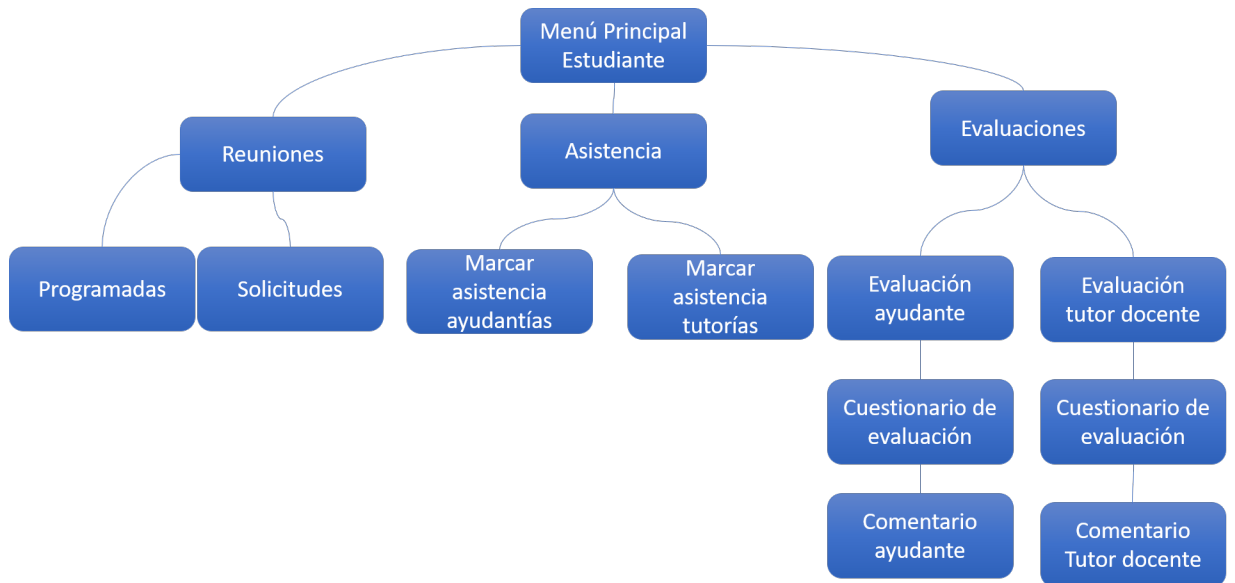


Figura E.4: Navegación del perfil de estudiante

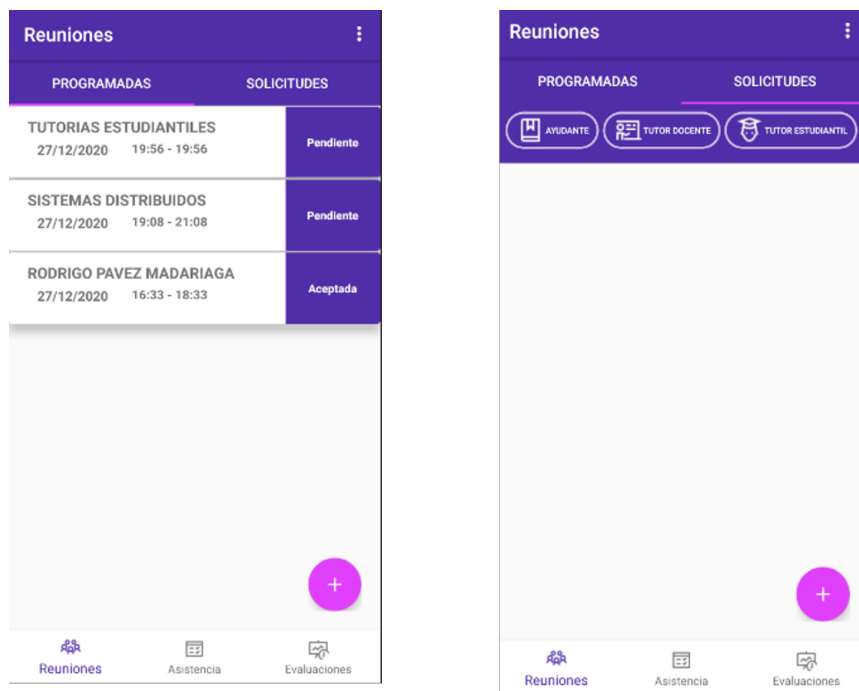


Figura E.5: Vistas para visualizar reuniones y solicitudes en el perfil de estudiante

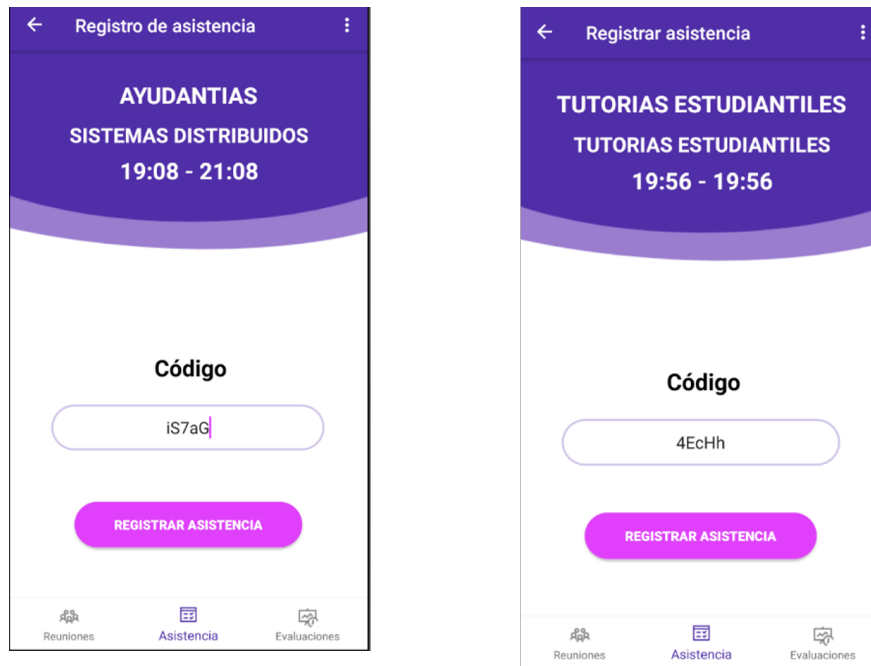


Figura E.7: Vistas para ingresar el código para marcar asistencia en un ayudantía y tutoría estudiantil

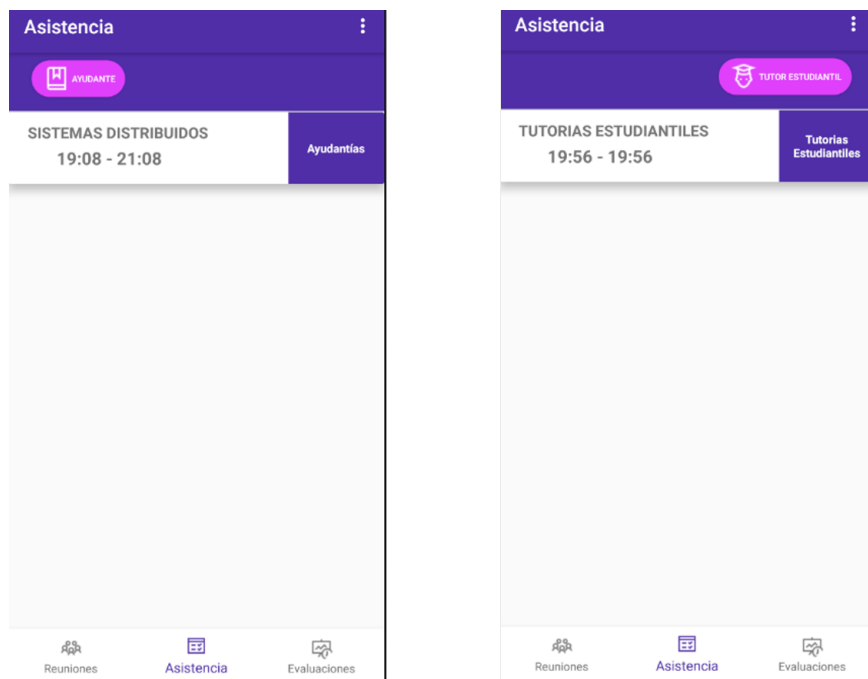


Figura E.6: Vistas para marcar asistencia para ayudantías y tutorías estudiantiles

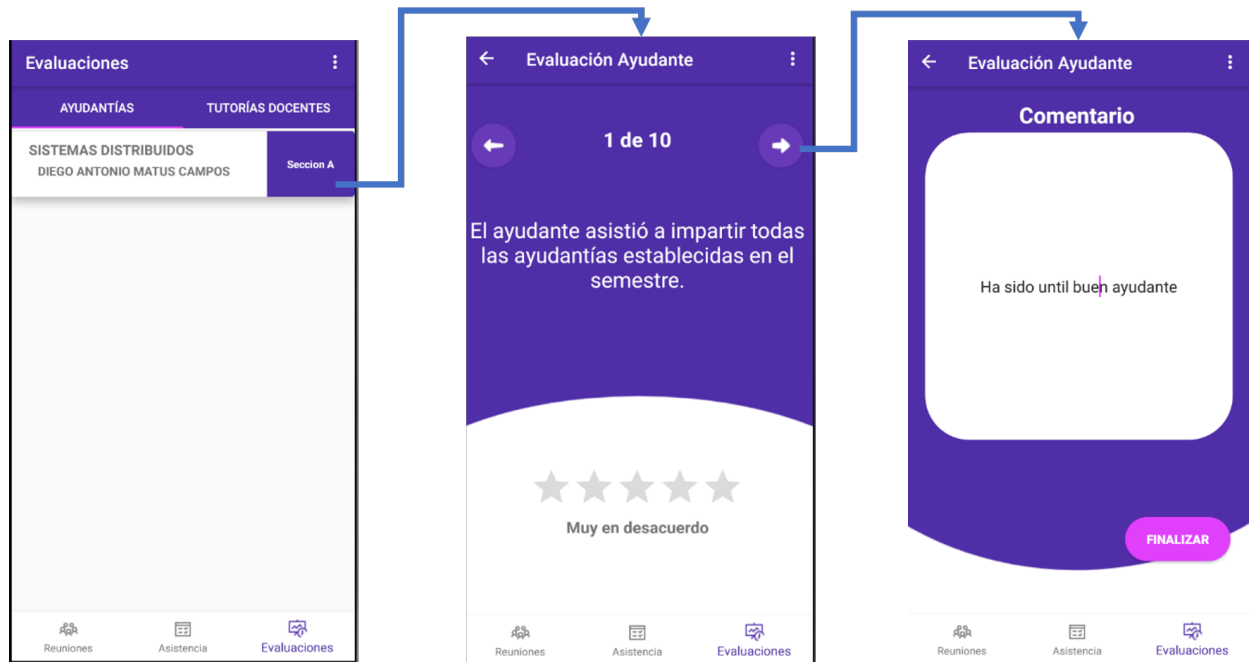


Figura E.8: Vistas para evaluar un ayudante

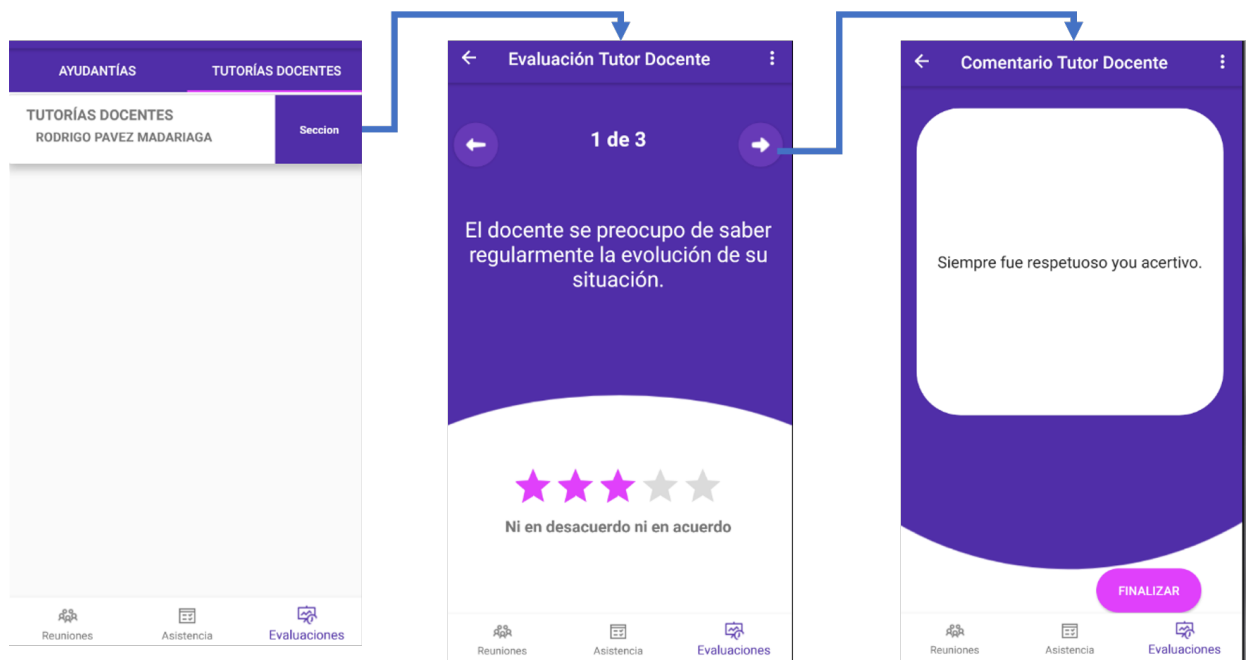


Figura E.9: Vistas para evaluar un tutor docente

E.3. Navegación perfil de docente



Figura E.10: Navegación entre vistas perfil del docente

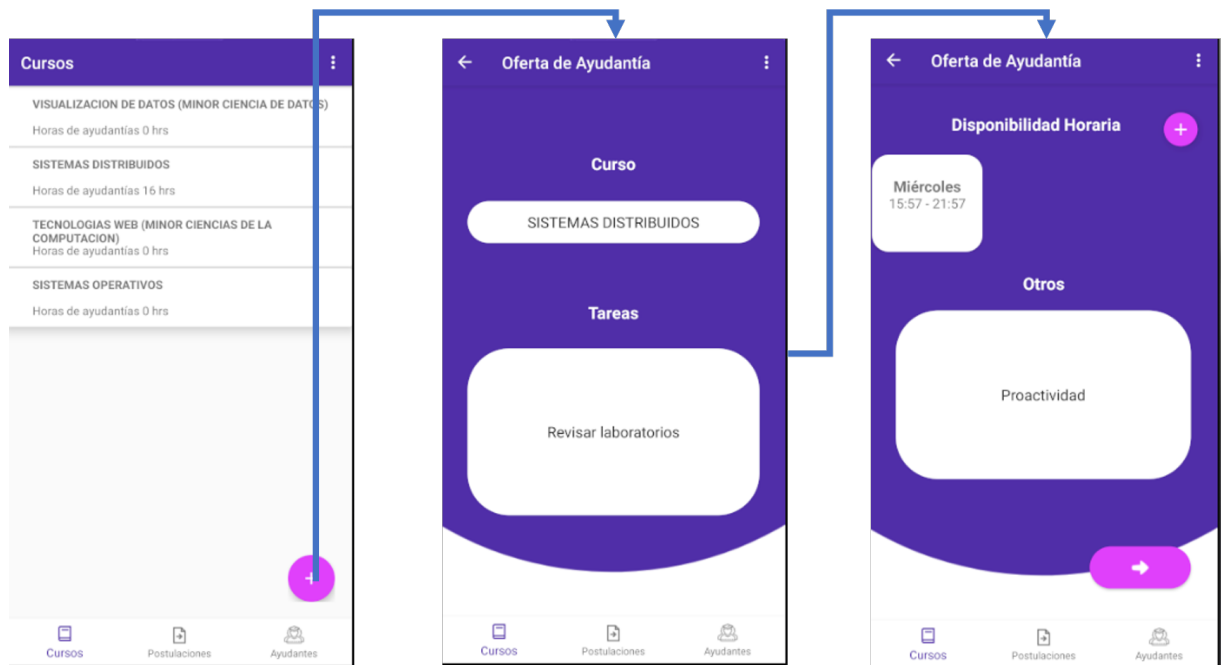


Figura E.11: Vistas para agregar una oferta de ayudantía

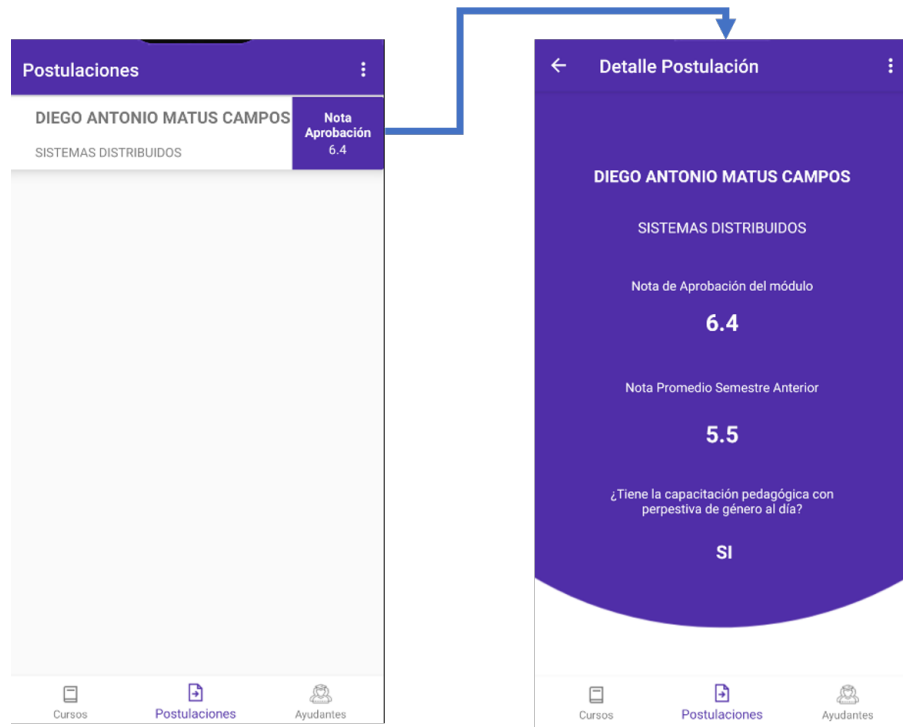


Figura E.12: Vistas para visualizar las postulaciones de ayudantías

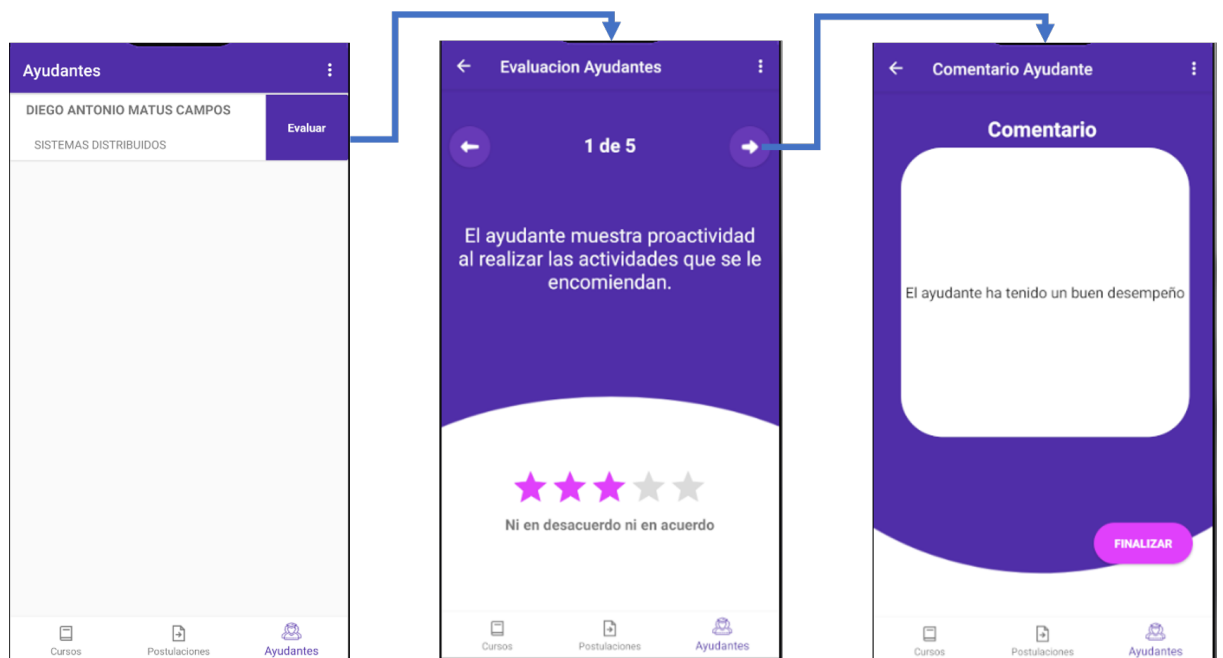


Figura E.13: Vistas para evaluar un ayudante

E.4. Navegación del perfil de tutor docente



Figura E.14: Navegación del perfil tutor docente

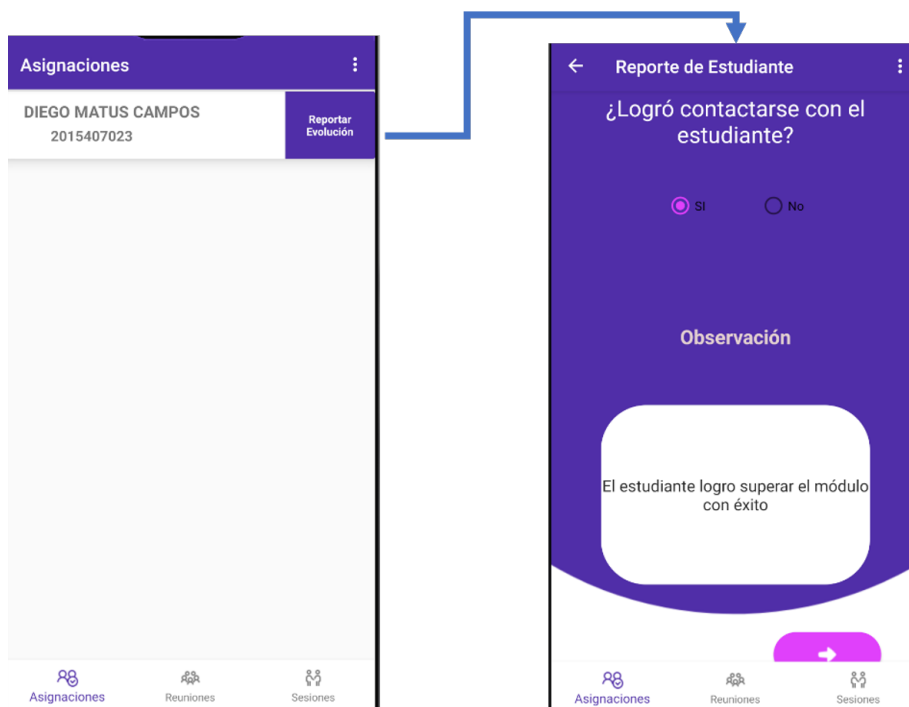


Figura E.15: Vistas para visualizar asignaciones y reportar avance del estudiante

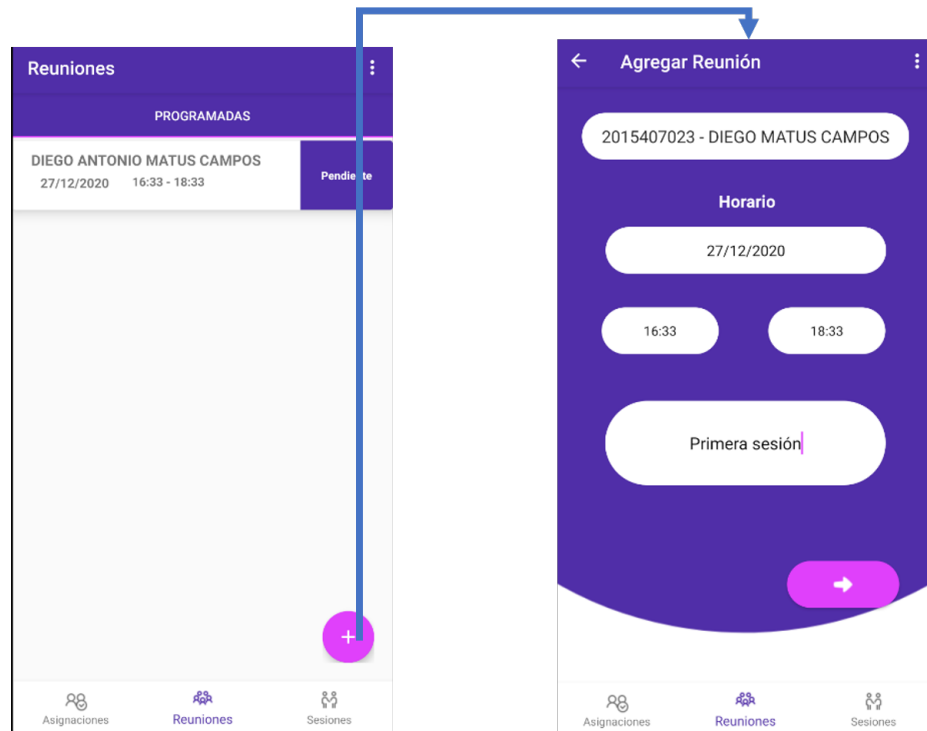


Figura E.16: Vistas para agregar una reunión de tutorías docentes

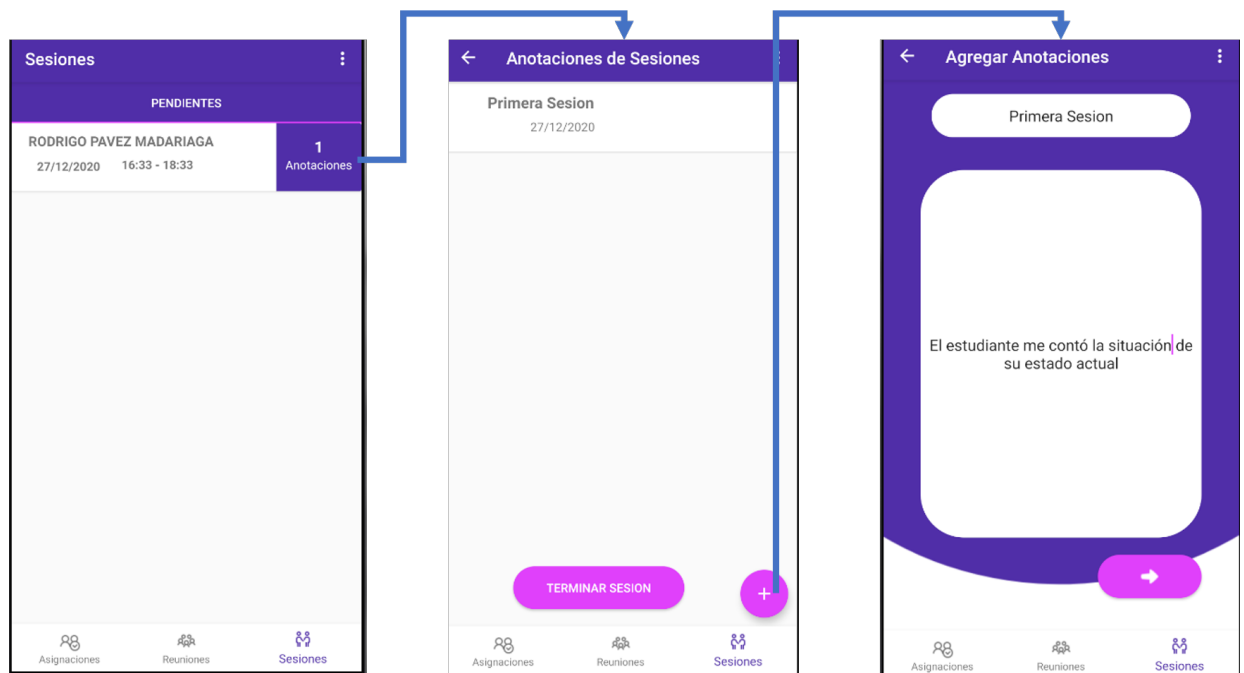


Figura E.17: Vistas para visualizar sesiones y anotaciones

E.5. Navegación tutorías estudiantiles



Figura E.18: Navegación del perfil de tutor estudiantil

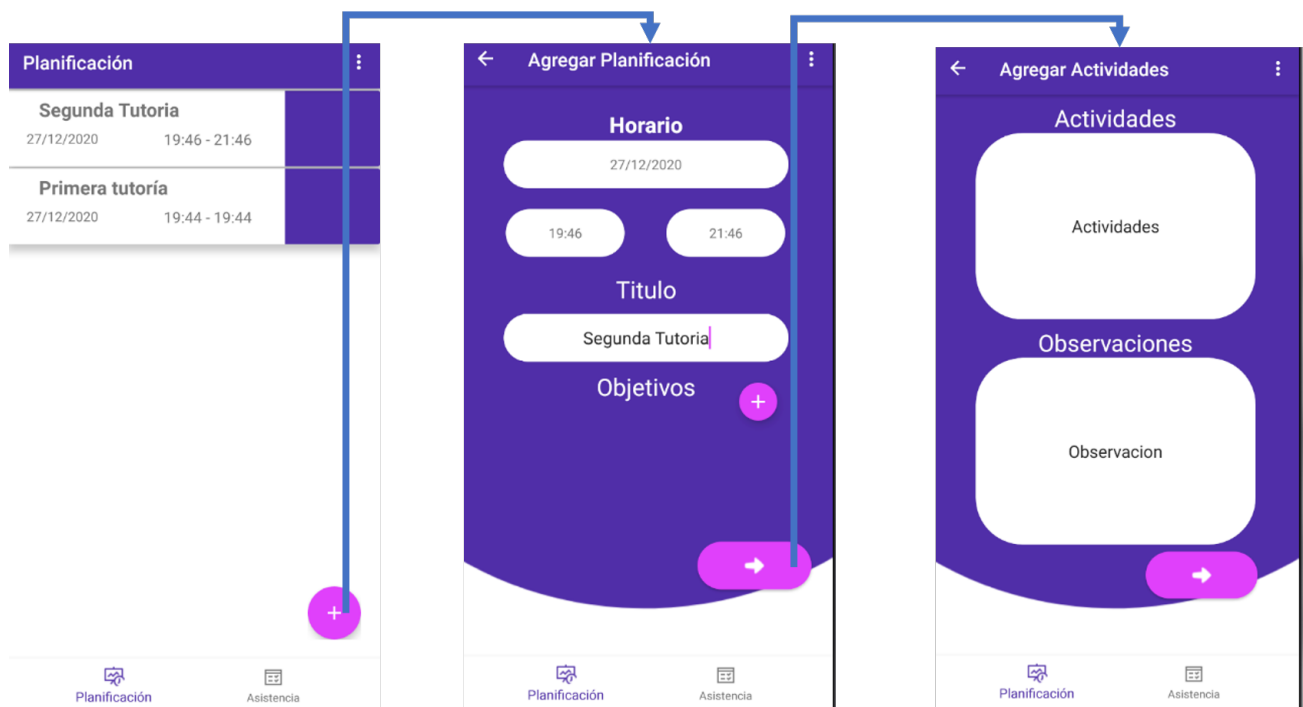


Figura E.19: Vistas para agregar una planificación e tutorías estudiantiles

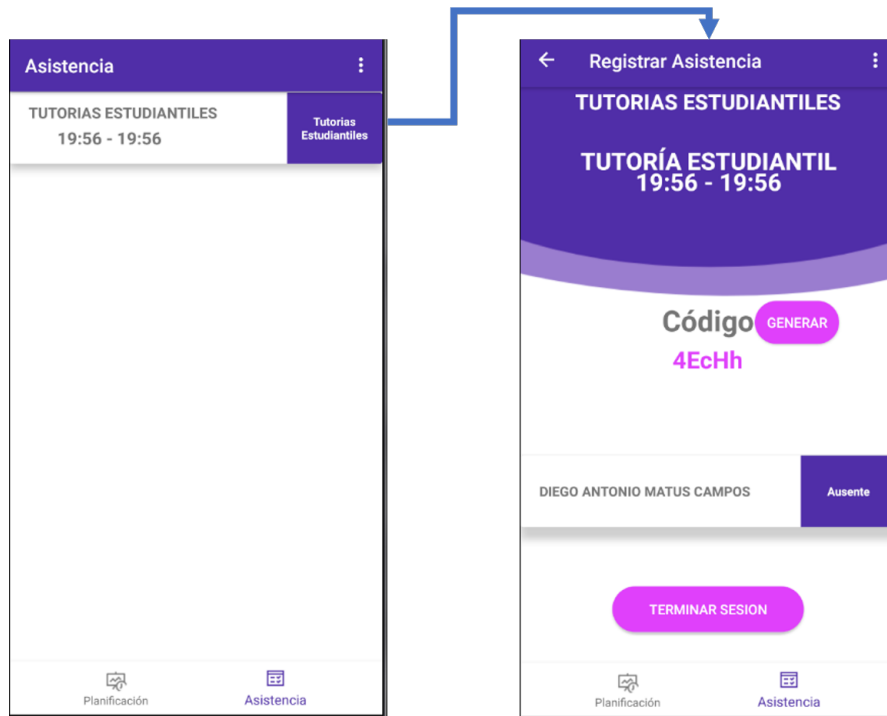


Figura E.20: Vistas para tomar asistencia de tutorías estudiantiles

E.6. Navegación para postulaciones a ayudantías

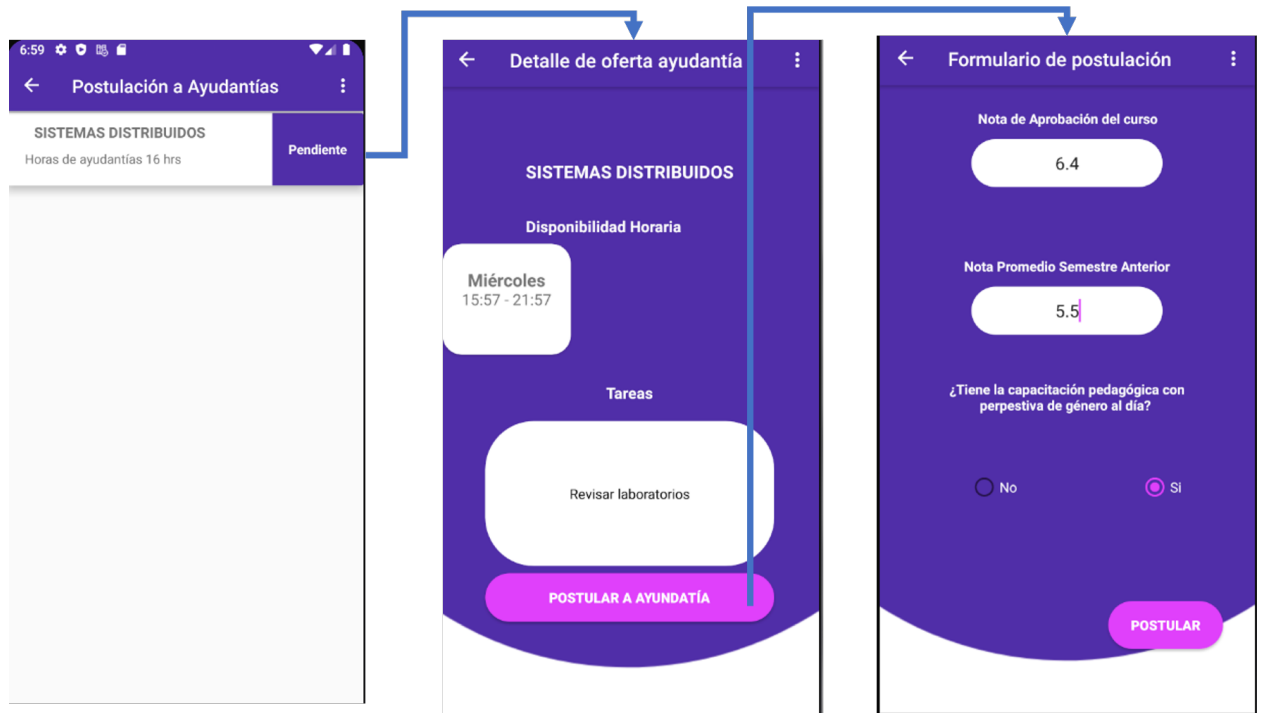


Figura E.21: Vistas para postular a una ayudantía

F. Evidencia de pruebas de aplicación web

F.1. Prueba de carga masiva de datos

F.1.1. Entradas de la prueba

NBE_ASIGNATURA	COD_ASIGNATUI	NBE_CURSO	COD_CUR	SECCO	CUI	ALUMNO
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	AVENDAÑO NUÑEZ ELIA
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	BARAHONA LEAL DIEGO
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	BECERRA FLORES EDWA
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	CANCINO RAMOS SEBA
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	ERENCHUN MARQUEZ N
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	GARCÍA OLIVARES IAN I
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	GARCÍA SEGURA LUCIAN
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	GAVILÁN RETAMAL CAM
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	GONZALEZ FERNÁNDEZ
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	HENRIQUEZ GALLARDO
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	IBÁÑEZ LÓPEZ JOAQUÍN
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	MARTÍNEZ HERNÁNDEZ
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	OLAVE REYES FELIPE IGN
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	ORTIZ SANCHEZ FELIPE
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	PEREDO ZAMORANO EL
ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	ADMINISTRACION DE REDES Y SISTEMAS COMPUTACIONALES	3407B423	A	35	PINO MORENO FERNAN

Figura F.1: Entradas para la prueba de carga masiva de datos - Archivo XLS

F.1.2. Salidas de la prueba

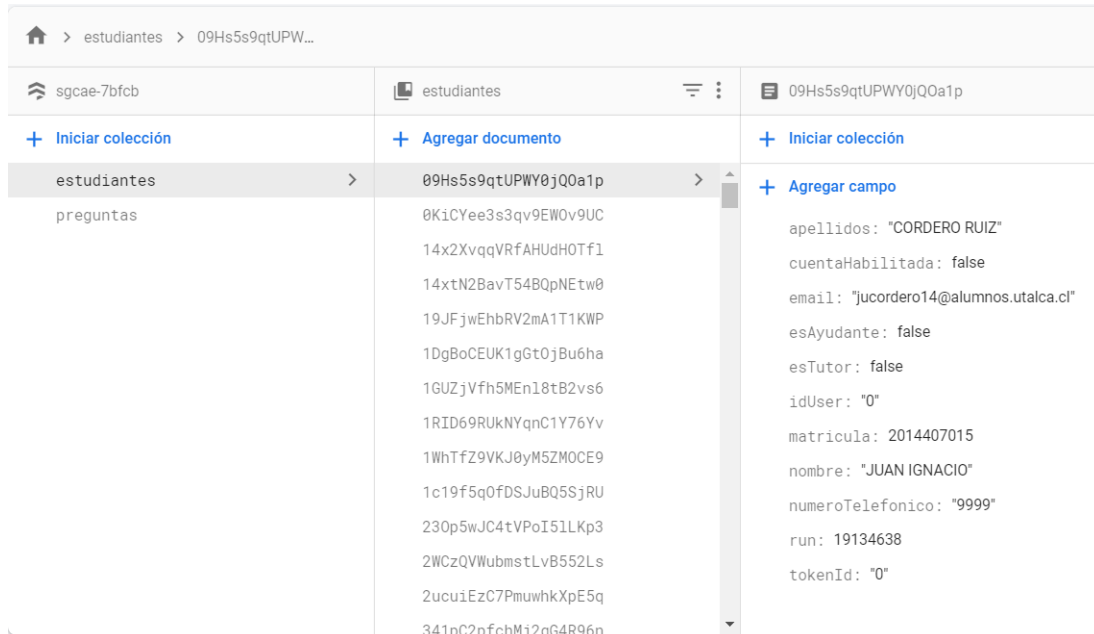


Figura F.2: Salidas de la prueba de carga masiva de datos - registro de colección estudiantes en base de datos

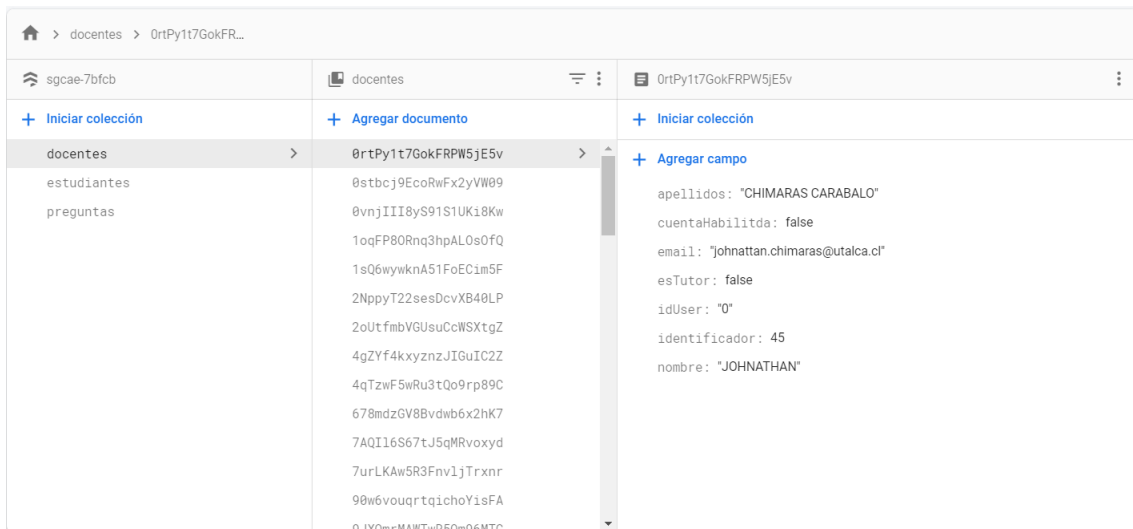


Figura F.3: Salidas de la prueba de carga masiva de datos - registro de colección docentes en base de datos

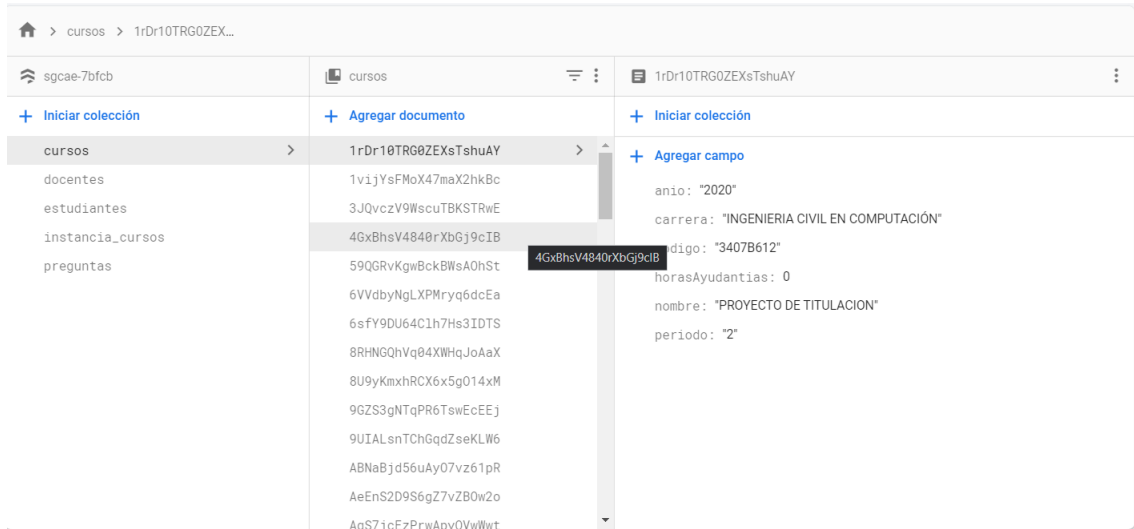


Figura F.4: Salidas de la prueba de carga masiva de datos - registro de colección cursos en base de datos

F.2. Prueba de asignación de estudiantes a docentes tutores

F.2.1. Entradas de la prueba

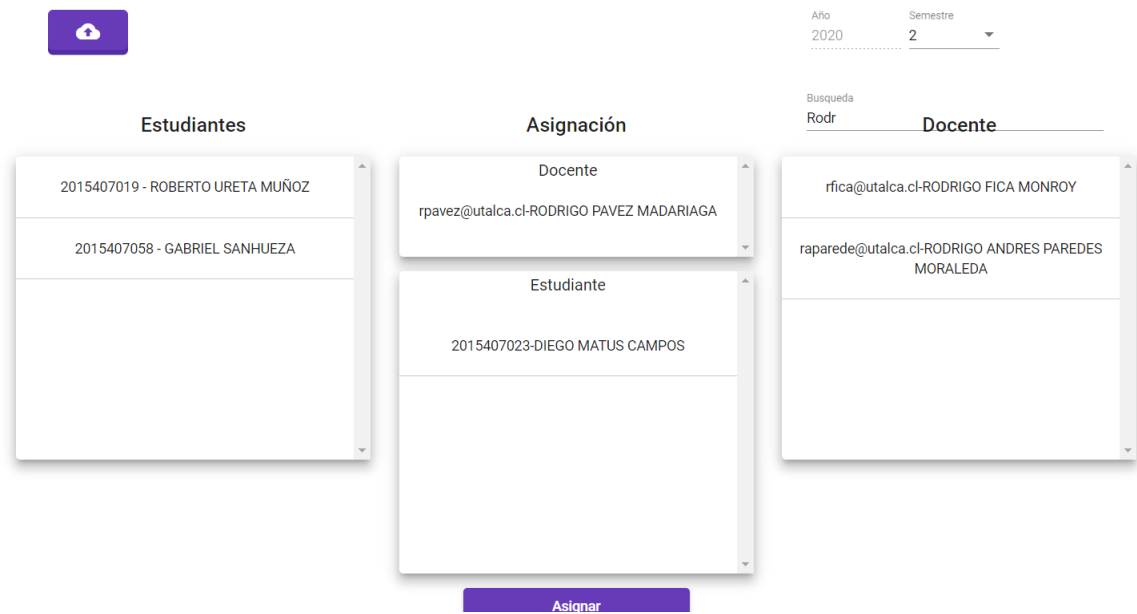


Figura F.5: Entradas de las prueba de asignación - estudiantes y docente

F.2.2. Salidas de la prueba

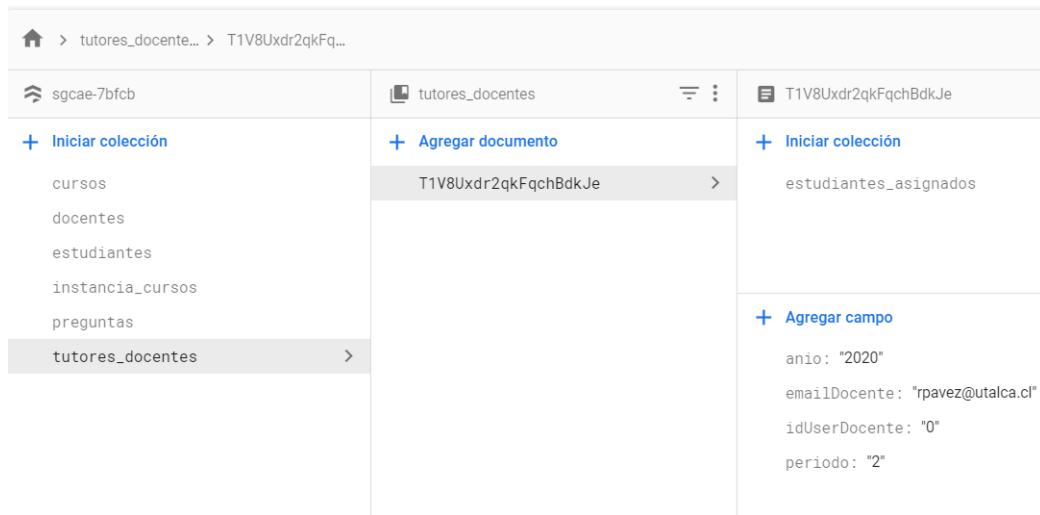


Figura F.6: Salidas de la prueba de asignación de estudiantes a tutores docentes - registro en base de datos

F.3. Prueba de asignación de estudiantes a tutores estudiantiles

F.3.1. Entrada de la prueba

The screenshot displays the 'Asignación de Estudiantes' interface. At the top, there are tabs for 'Asignación de Estudiantes' and 'Estudiantes'. Below the tabs, there are two purple buttons: 'Estudiantes' on the left and 'Tutores' on the right. In the top right corner, there are dropdown menus for 'Año' (set to 2020) and 'Semestre' (set to 2). The main area is divided into three columns: 'Estudiantes', 'Asignación', and 'Tutor'. Each column contains a list of student names and IDs. The 'Asignación' column is currently empty, while the 'Estudiantes' and 'Tutor' columns contain the following entries:

Estudiantes	Asignación	Tutor
2015407019 - ROBERTO URETA MUÑOZ	Tutor 2015407023-DIEGO MATUS CAMPOS	2015407019-ROBERTO URETA MUÑOZ
2015407058 - GABRIEL SANHUEZA	Estudiante 2015407023-DIEGO MATUS CAMPOS	2015407058-GABRIEL SANHUEZA
		2016407035-PAULO PARRAGUEZ DÍAZ

At the bottom center, there is a purple button labeled 'Asignar'.

Figura F.7: Entradas de las prueba de asignación - estudiantes y tutor

F.3.2. Salidas de la prueba

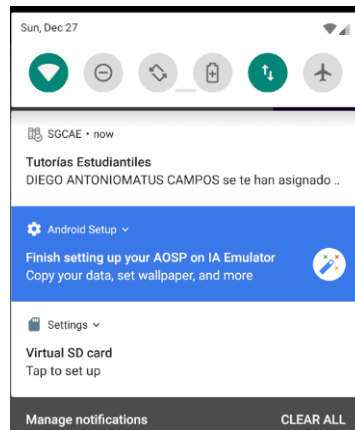


Figura F.8: Salida de las prueba de asignación de estudiantes a tutor estudiantil - registro de tutor estudiantil

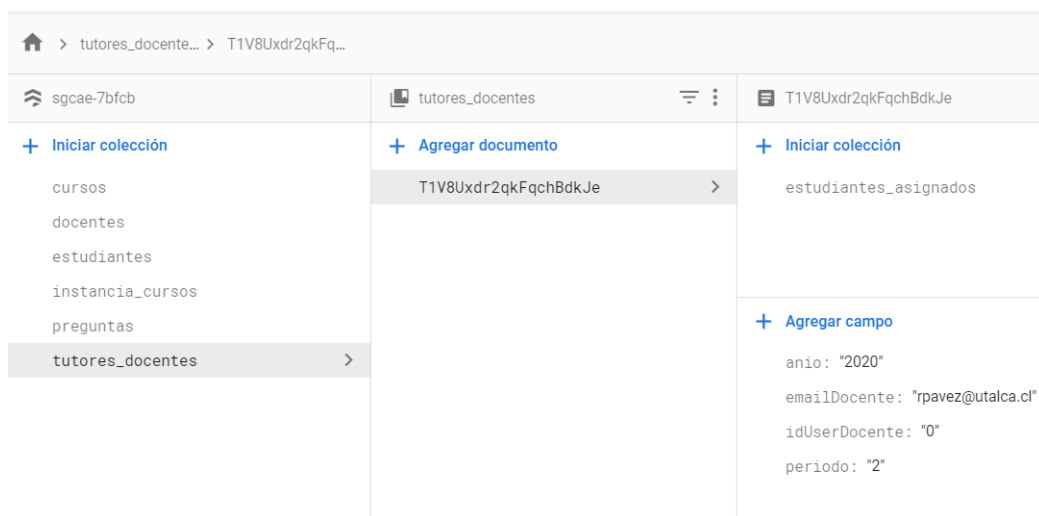
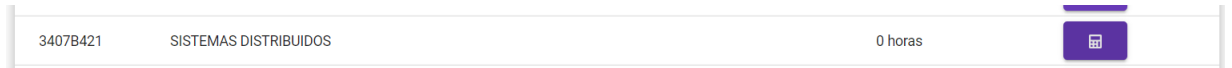


Figura F.9: Salidas de la prueba de asignación de estudiantes a tutores docentes - notificación a tutor

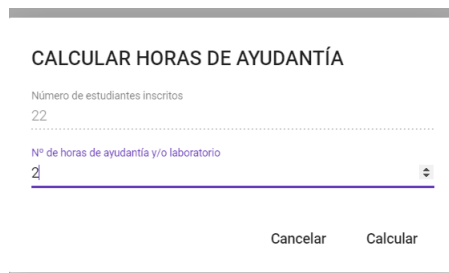
F.4. Prueba de cálculo de horas propuestas de horas de ayudantías

F.4.1. Entrada de la prueba



A horizontal bar interface for course selection. On the left, it displays the course ID '3407B421' and the name 'SISTEMAS DISTRIBUIDOS'. On the right, it shows '0 horas' and a purple button with a calculator icon.


Figura F.10: Entradas para la prueba de cálculo de horas - seleccionar un curso



A dialog box titled 'CALCULAR HORAS DE AYUDANTÍA'. It contains two input fields: 'Número de estudiantes inscritos' with the value '22', and 'Nº de horas de ayudantía y/o laboratorio' with the value '2'. At the bottom, there are two buttons: 'Cancelar' and 'Calcular'.

Figura F.11: Entradas para la prueba de cálculo de horas - ingreso de horas de ayudantías y/o laboratorio

F.4.2. Salidas de la prueba



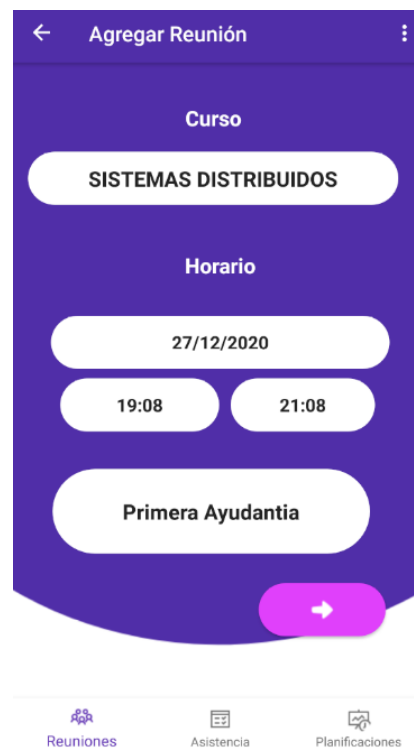
A horizontal bar interface showing the result of the calculation. It displays the course ID '3407B421' and the name 'SISTEMAS DISTRIBUIDOS'. On the right, it shows '16 horas' and a grey button with a calculator icon.

Figura F.12: Salidas para la prueba de cálculo de horas - curso con la horas calculadas

G. Evidencia de pruebas de la aplicación móvil

G.1. Prueba de registro de reuniones de ayudantías

G.1.1. Entradas de la prueba



The screenshot shows a mobile application interface for adding a meeting. The title bar at the top is purple and contains a back arrow, the text "Agregar Reunión", and a menu icon. Below the title bar, the form is divided into sections: "Curso" with a white rounded rectangle containing "SISTEMAS DISTRIBUIDOS"; "Horario" with a white rounded rectangle containing "27/12/2020", two smaller white rounded rectangles containing "19:08" and "21:08"; and "Primera Ayudantia" with a white rounded rectangle containing "Primera Ayudantia". At the bottom right of the form is a purple rounded rectangle with a white right-pointing arrow. The bottom navigation bar is white and contains three icons with labels: "Reuniones" (purple icon), "Asistencia" (purple icon), and "Planificaciones" (purple icon).

Figura G.1: Entradas para agregar una reunión de ayudantías

G.1.2. Salidas de la prueba

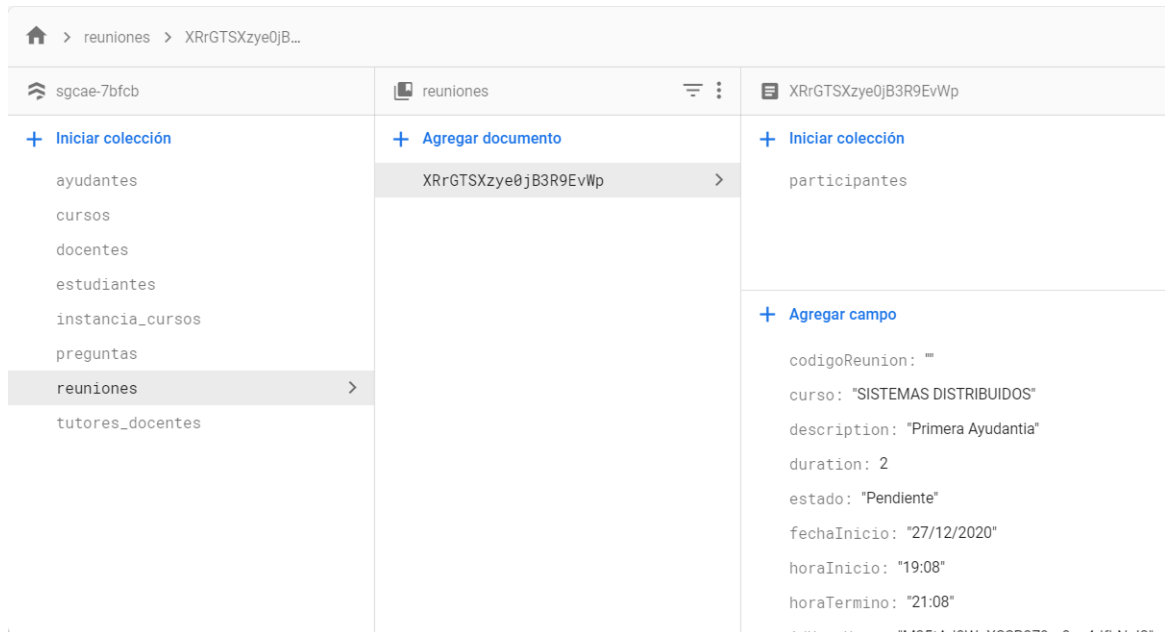


Figura G.2: Salidas para la prueba de agregar una reunión de ayudantía

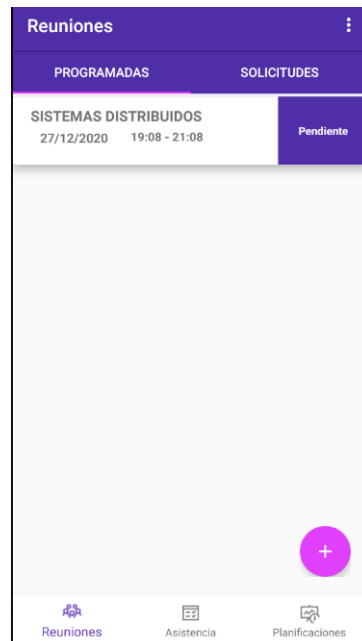


Figura G.3: Muestra de a reunión agregada

G.2. Prueba de registro de planificaciones de ayudantías

G.2.1. Entradas de la prueba

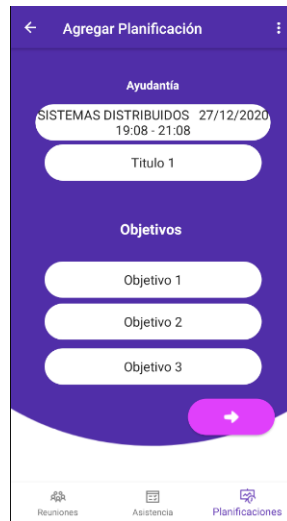


Figura G.4: Entradas de registro de planificación - Primer paso

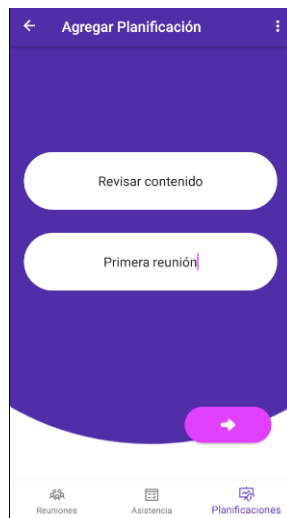


Figura G.5: Entradas de registro de planificación - Segundo paso

G.2.2. Salidas de las pruebas

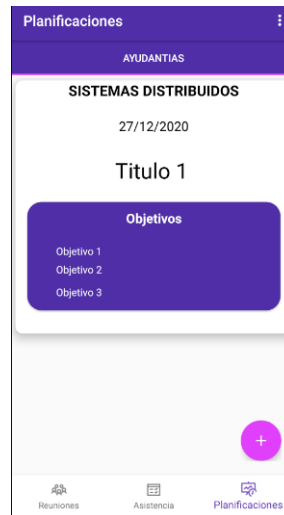


Figura G.6: Salidas para prueba de registro de planificación

G.3. Prueba de registro de asistencia de ayudantías ayudante

G.3.1. Entradas de la pruebas

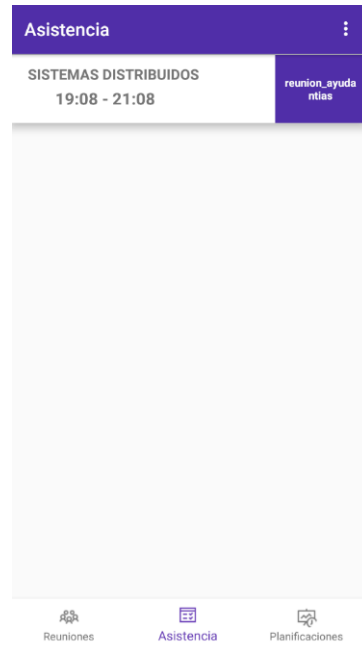


Figura G.7: Entradas para registrar asistencia - Seleccionar una reunión



Figura G.8: Entradas para registrar asistencia - Generar código

G.3.2. Salidas de la pruebas

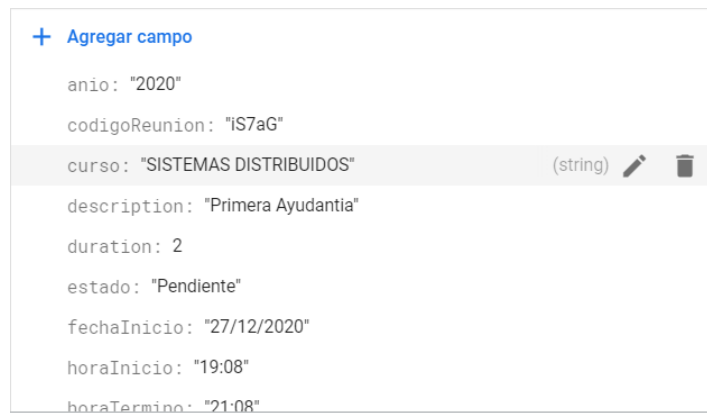


Figura G.9: Salida de la prueba de registro de asistencia - Actualización del código

G.4. Prueba de registro de asistencia de ayudantías estu- diente

G.4.1. Entradas de la prueba

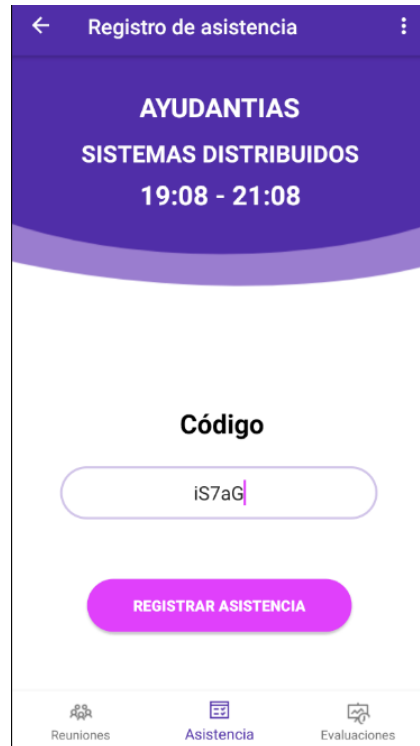


Figura G.10: Entradas de la prueba de registro de asistencia estudiante

G.4.2. Salidas de la prueba

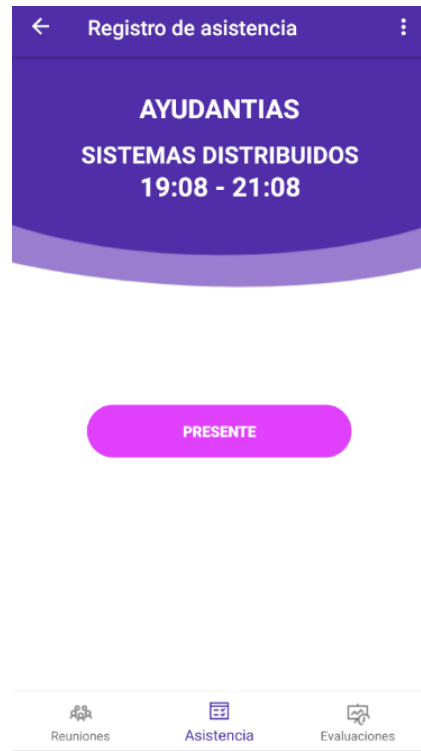


Figura G.11: Salidas de la prueba de registro de asistencia estudiantes

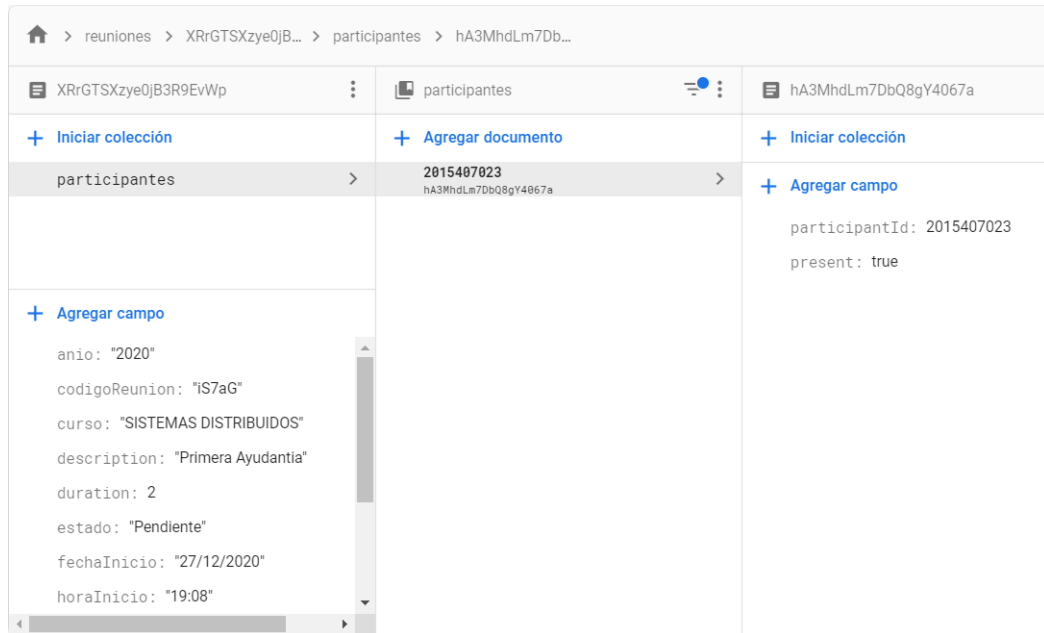


Figura G.12: Salidas de la prueba de registro de asistencia estudiante - Actualización en base de datos

G.5. Prueba de postulación de ayudante

G.5.1. Entrada de la prueba

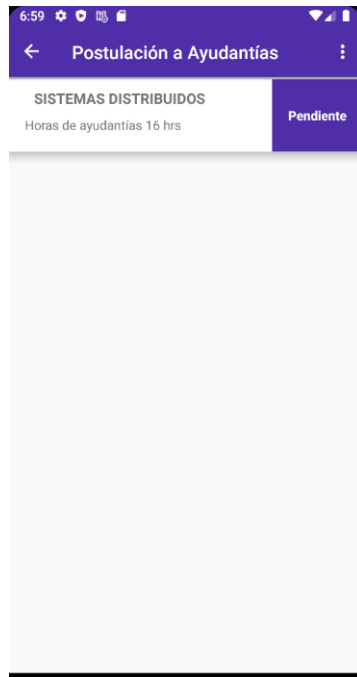


Figura G.13: Entradas de la prueba de selección de ayudante - Selección de postulación

Formulario de postulación

Nota de Aprobación del curso

6.4

Nota Promedio Semestre Anterior

5.5

¿Tiene la capacitación pedagógica con perspectiva de género al día?

No Si

POSTULAR

Figura G.14: Entradas de la prueba de selección de ayudante - Ingresar datos de postulación

G.5.2. Salidas de la prueba

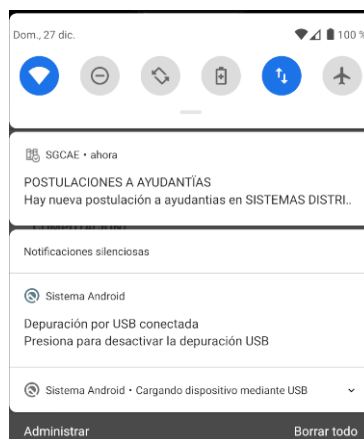


Figura G.15: Salidas de la prueba de selección de ayudante - Notificación a docente

G.6. Prueba de selección de ayudante

G.6.1. Entradas de la prueba

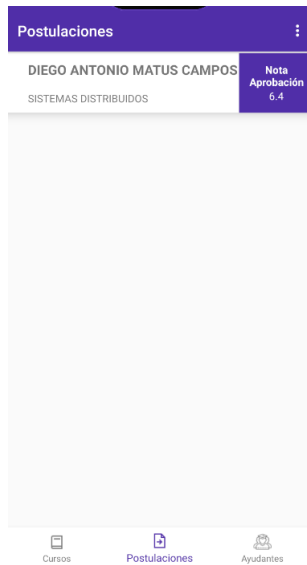


Figura G.16: Entradas para la prueba de selección - Selección de postulaciones

G.6.2. Salidas de la prueba

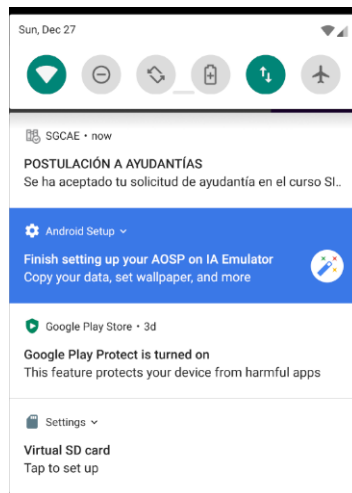
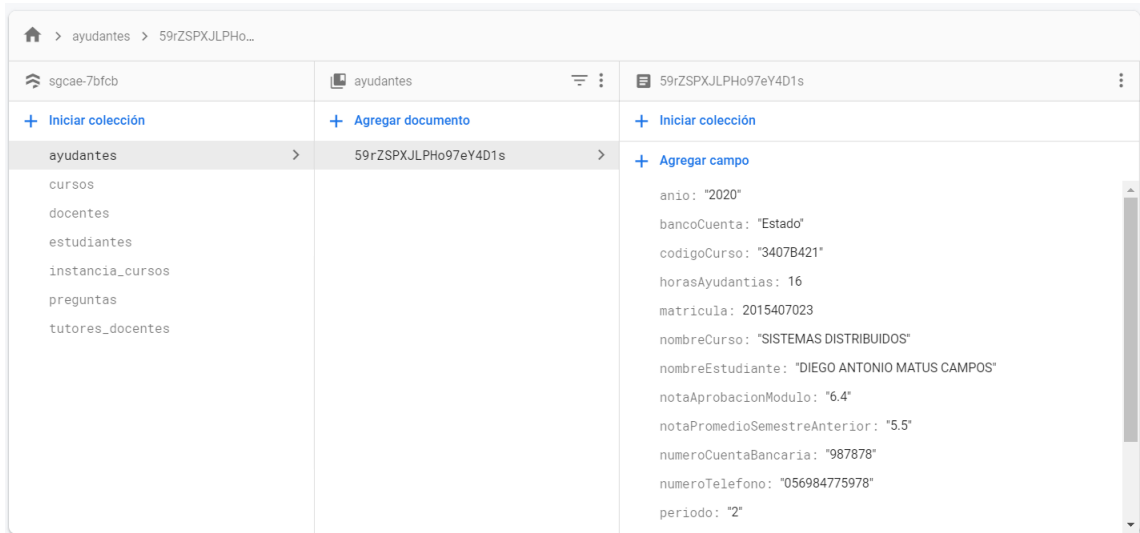


Figura G.17: Salidas para la prueba de selección - Notificación de aceptación



ayudantes	59rZSPXJLPHo97eY4D1s	
<ul style="list-style-type: none"> cursos docentes estudiantes instancia_cursos preguntas tutores_docentes		<ul style="list-style-type: none"> + Iniciar colección + Agregar campo año: "2020" bancoCuenta: "Estado" codigoCurso: "3407B421" horasAyudantias: 16 matricula: 2015407023 nombreCurso: "SISTEMAS DISTRIBUIDOS" nombreEstudiante: "DIEGO ANTONIO MATUS CAMPOS" notaAprobacionModulo: "6.4" notaPromedioSemestreAnterior: "5.5" numeroCuentaBancaria: "987878" numeroTelefono: "056984775978" periodo: "2"

Figura G.18: Salidas para prueba de selección de ayudante - Registro de ayudante en base de datos

G.7. Prueba de evaluación de ayudantes

G.7.1. Entradas de la prueba

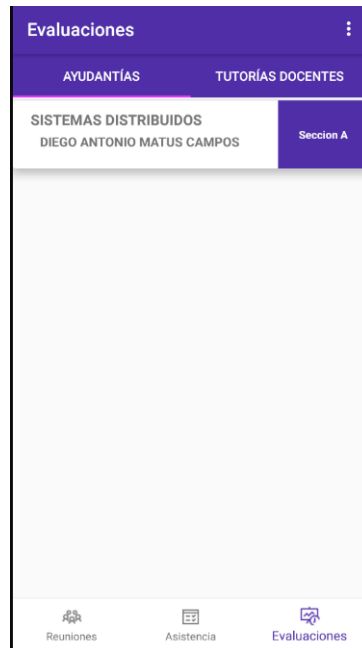


Figura G.19: Entradas para la prueba de evaluación de ayudantes - Selección de de ayudante a evaluar

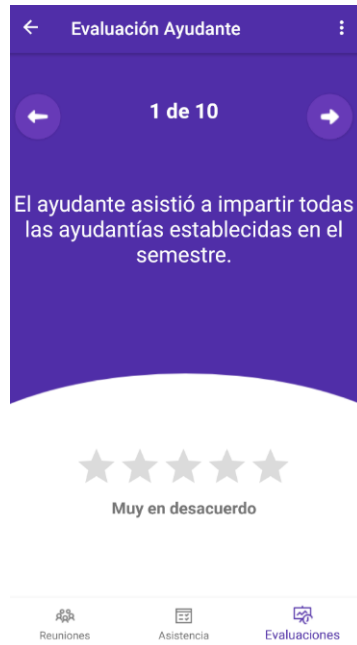


Figura G.20: Entradas para la prueba de evaluación de ayudantes - Respuestas del cuestionario de evaluación

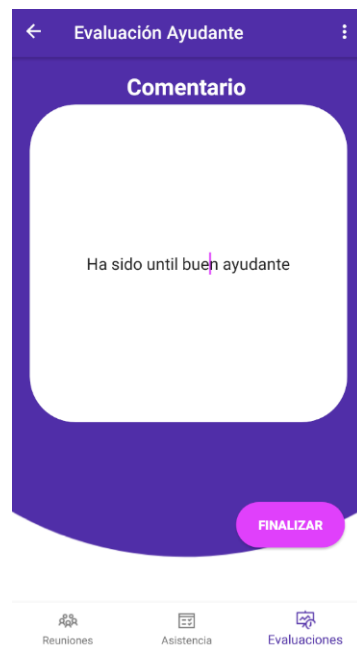
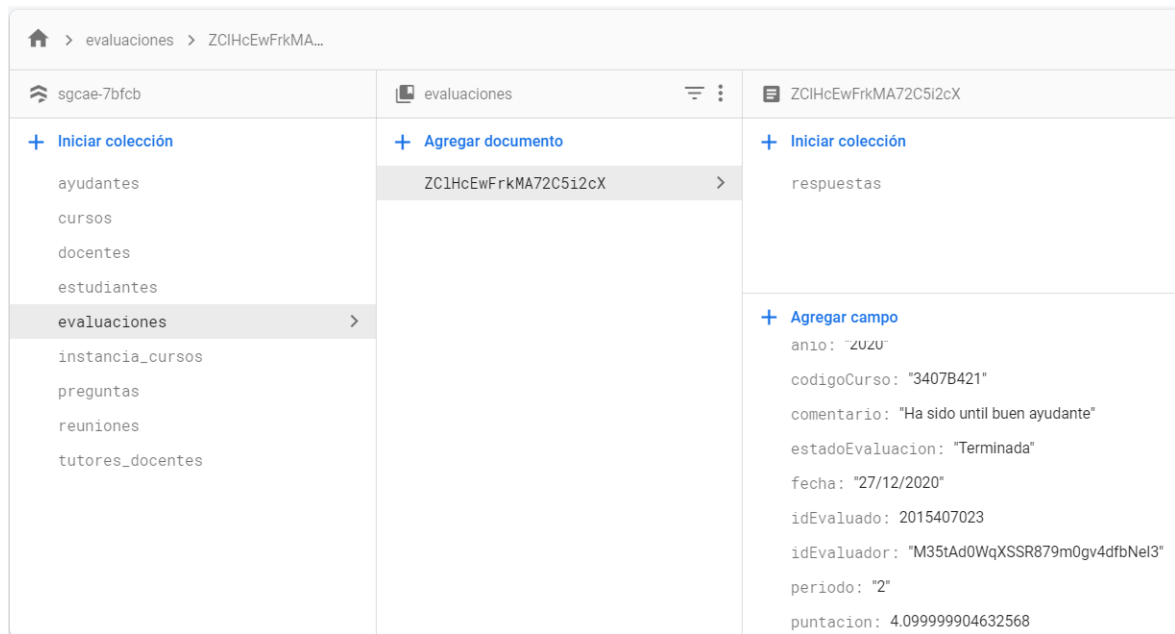


Figura G.21: Entradas para la prueba de evaluación de ayudantes - Comentario de al ayudante

G.7.2. Salidas de la prueba de evaluación de ayudante

sgcae-7bfc	evaluaciones	ZC1HcEwFrkMA72C5i2cX
+ Iniciar colección ayudantes cursos docentes estudiantes evaluaciones > instancia_cursos preguntas reuniones tutores_docentes	+ Agregar documento ZC1HcEwFrkMA72C5i2cX >	+ Iniciar colección respuestas + Agregar campo anio: "2020" codigoCurso: "3407B421" comentario: "Ha sido un buen ayudante" estadoEvaluacion: "Terminada" fecha: "27/12/2020" idEvaluado: 2015407023 idEvaluador: "M35tAd0WqXSSR879m0gv4dfbNel3" periodo: "2" puntacion: 4.099999904632568

Figura G.22: Salida de la prueba de evaluación de ayudantes - Registro en base de datos

G.8. Prueba de evaluación de ayudantes por parte del docente



Figura G.23: Entradas para la prueba de evaluación de ayudantes - Selección de ayudante a evaluar

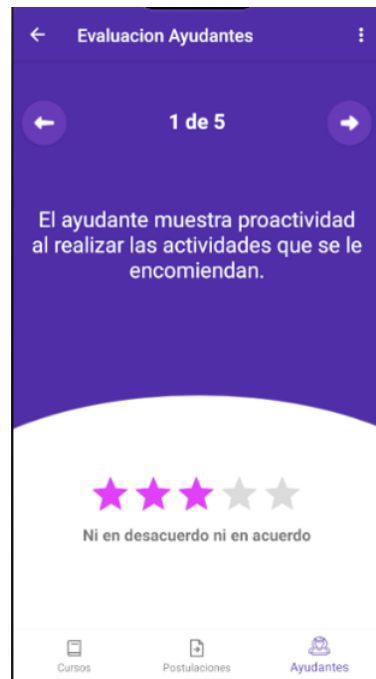


Figura G.24: Entradas para la prueba de evaluación de ayudantes - Respuesta del cuestionario de evaluación

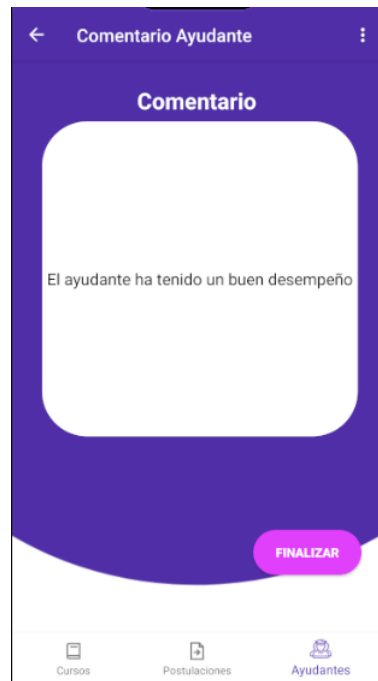


Figura G.25: Entradas para la prueba de evaluación de ayudantes - Comentario a ayudante

G.8.1. Salidas de la prueba

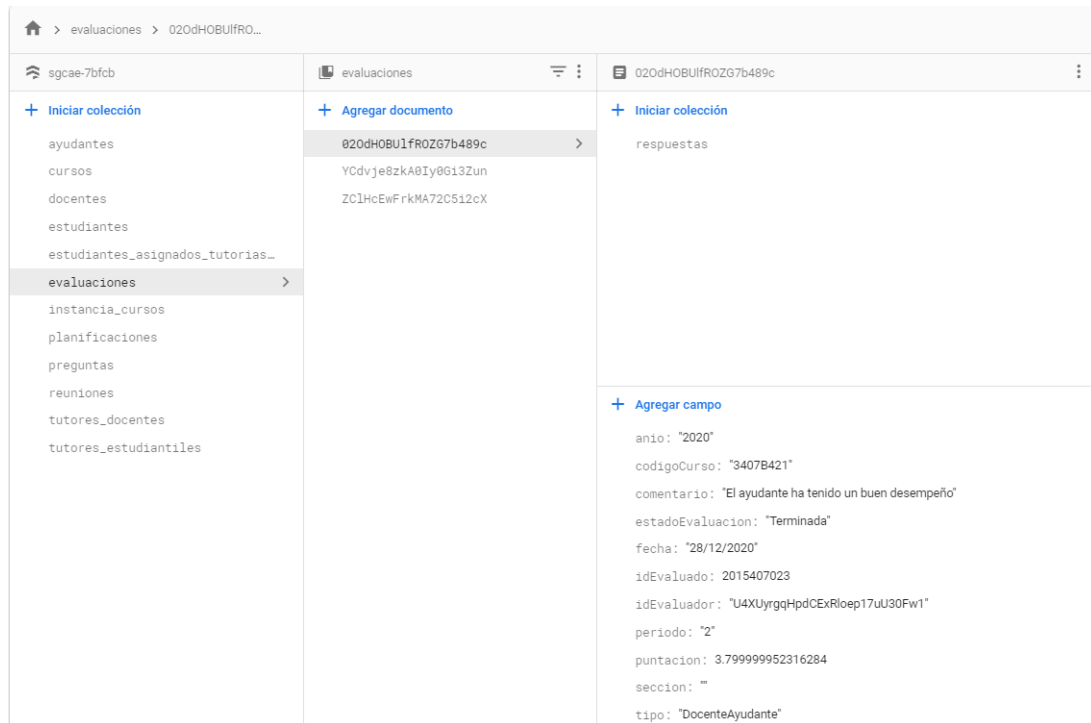


Figura G.26: Salida de la prueba de evaluación a ayudante - Registro en base de datos

G.9. Prueba de evaluación de tutores docentes

G.9.1. Entradas de la prueba

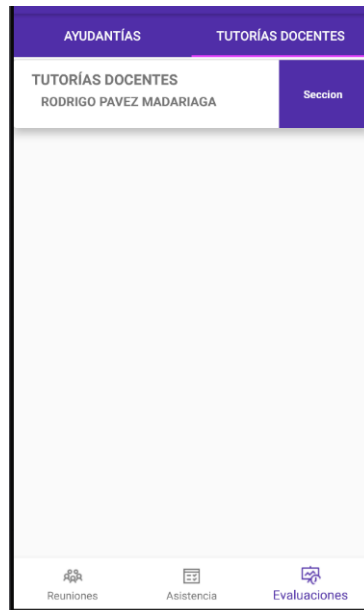


Figura G.27: Entradas para la prueba de evaluación de tutores docentes - Selección de tutor docente

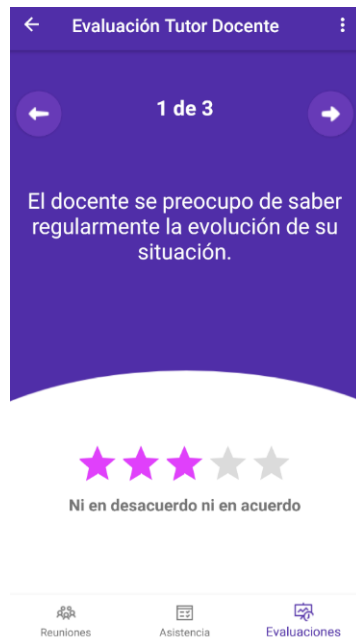


Figura G.28: Entradas para la prueba de evaluación de tutores docentes - Respuestas de cuestionario

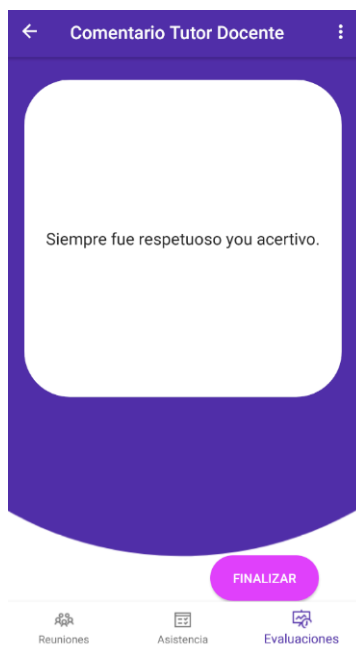


Figura G.29: Entradas para la prueba de evaluación de tutores docentes - Comentario a tutor docente

G.9.2. Salidas de la prueba

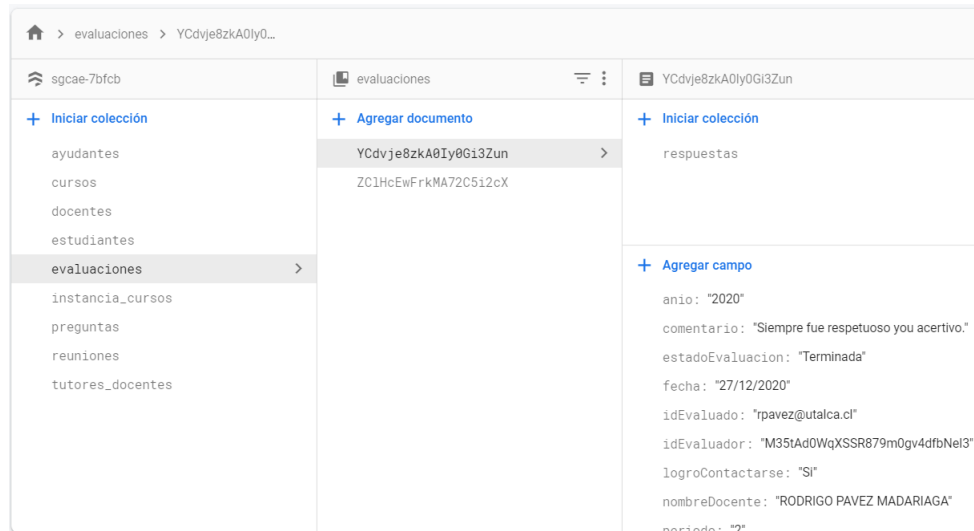


Figura G.30: Salidas para prueba de evaluación de tutores docentes - Registro en base de datos

H. Resumen de cuestionarios aplicados

H.1. Resumen de resultados del cuestionario de usabilidad de la aplicación móvil

Cuadro H.1: Tabla resumen del cuestionario de usabilidad de la aplicación móvil

Preguntas	Escala Likert				
	1	2	3	4	5
Creo que me gustaría utilizar esta aplicación móvil frecuentemente.	0	0	3	6	6
Creo que la aplicación móvil es innecesariamente compleja.	5	5	4	1	0
Creo que la aplicación móvil es fácil de usar.	0	0	7	4	4
Creo que necesitaría la ayuda de una persona que sea capaz de usar esta aplicación móvil .	5	5	3	1	1
Creo que las variadas funciones de esta aplicación móvil están bien integradas.	0	1	1	7	6
Pensé que había demasiada inconsistencia en la aplicación móvil.	2	6	5	2	0
Imagino que la mayoría de las personas podrán aprender a usar este aplicación móvil muy rápido.	1	0	3	6	5

Creo que la aplicación móvil es muy incómoda de usar.	7	4	3	1	0
Siento confianza usando esta aplicación móvil.	0	1	2	4	8
Necesité aprender muchas cosas antes de empezar a utilizar esta aplicación móvil.	7	5	3	0	0

H.2. Resumen de los resultados del cuestionario de usabilidad para la aplicación móvil.

Cuadro H.2: Tabla resumen del cuestionario de usabilidad de la aplicación web

Preguntas	Escala Likert				
	1	2	3	4	5
Creo que me gustaría utilizar esta aplicación web frecuentemente.	0	0	0	1	1
Creo que la aplicación web es innecesariamente compleja.	1	1	0	0	0
Creo que la aplicación web es fácil de usar.	0	0	0	2	0
Creo que necesitaría la ayuda de una persona que sea capaz de usar esta aplicación web.	0	1	0	0	1
Creo que las variadas funciones de esta aplicación web están bien integradas.	0	0	1	1	0
Pensé que había demasiada inconsistencia en la aplicación web.	1	0	1	0	0
Imagino que la mayoría de las personas podrán aprender a usar este aplicación web muy rápido.	1	0	0	0	1
Creo que la aplicación web es muy incómoda de usar.	1	1	0	0	0
Siento confianza usando esta aplicación web.	0	1	0	1	0
Necesité aprender muchas cosas antes de empezar a utilizar esta aplicación web.	0	1	0	0	1

H.3. Escala de puntajes utilizado para medir el resultado de usabilidad

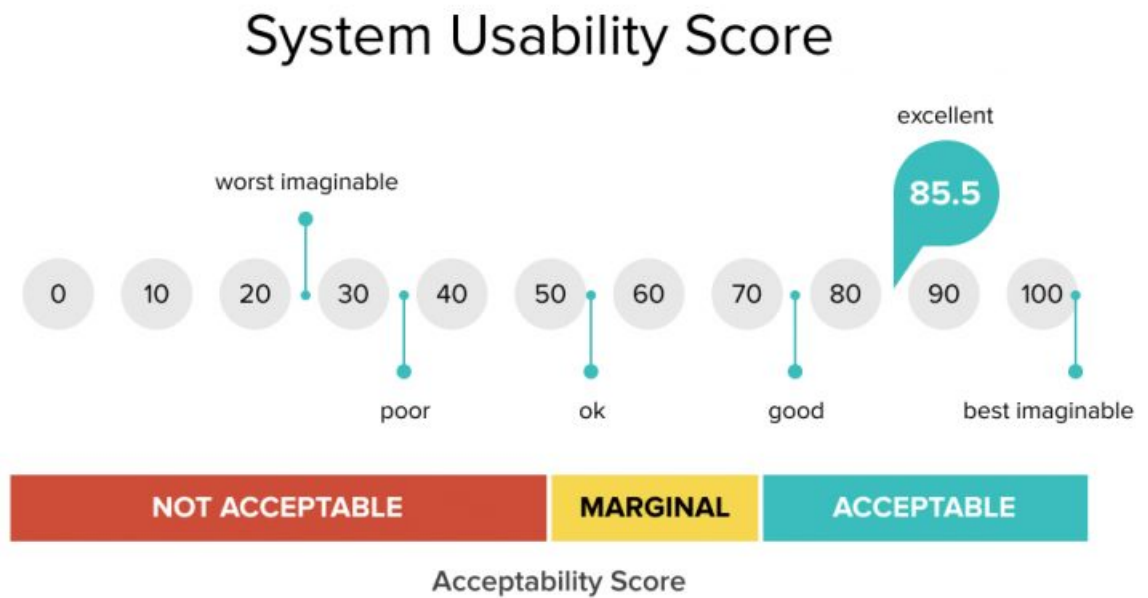


Figura H.1: Escala de puntaje utilizada para analizar el resultado de las pruebas de usabilidad