

INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo de titulación

**" Sistema de enlace de comunicación *edutor* para la
mejora en la educación básica"**

Que para cumplir con la opción de titulación curricular

en la carrera de

"Ingeniería en Sistemas Computacionales"

Presentan

Antonio Feregrino Bolaños

Ariana Estefania Jaime Evia del Puerto

Director:

M. en C. Edgardo Adrián Franco Martínez





Instituto Politécnico Nacional

Escuela Superior de Cómputo

SUBDIRECCIÓN ACADÉMICA



No. de Registro: ISCCR 110-2014-A003/2015

Fecha: 11 Septiembre 2015

Documento Técnico

“Sistema de enlace de comunicación *edutor* para la mejora en la educación básica”

Presentan:

Antonio Feregrino Bolaños ¹

Ariana Estefanía Jaime Evia del Puerto ²

Director:

M. en C. Edgardo Adrián Franco Martínez

RESUMEN

El trabajo que a continuación se presenta es un sistema enfocado a buscar mejorar la calidad de la educación principalmente en escuelas de nivel básico al combatir la falta de comunicación entre padres y tutores que se puede explicar por muchos factores, incluyendo entre ellos la falta de tiempo por parte de ambos. El sistema está basado por completo en tecnologías web en forma de un servicio consumible por una gran cantidad de dispositivos, facilitando así su acceso para el usuario final.

Palabras clave – Academia de ingeniería de software, Aplicaciones móviles, Apoyo a la educación básica, servicios Web

¹aferegrinob0900@alumno.ipn.mx

²ajaimee1100@alumno.ipn.mx

Advertencia

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.” La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

Agradecimientos

Primero: A la señora Ofelia, que siempre ha estado ahí hasta para ayudarme con las ecuaciones diferenciales (aunque no le supiera). Al señor Antonio, que tanto me ha sabido dar (“bueno, ¿pero ya cuando terminas la escuela?”). A Maximiliano, que siempre me distrajo impidiendo que me volviera loco. Gracias papá, mamá y hermanito.

Segundo: A Carlos, este camino empezó tratando de seguir tus pasos. A mi familia en general, desde un consejo hasta su compañía, se los agradezco. A mis grandes (y más antiguos) amigos Mario, Alonso y Alfonso, ya media vida a su lado. A mis amigos universitarios (de la UPIITA) Rodrigo, Mauricio, Rosendo, Priscila, y a los que alguna vez fuimos PLs o Lennins o lo que fuera (¿en qué estábamos pensando?). A mis amigos universitarios (de la ESCOM) Fanny, quién además me acompañó durante este trabajo, también a Julio, Miguel y Adrián (espero grandes cosas de ustedes). A la gente de Heurística Soluciones que siempre me ha apoyado, pero en especial a Marcos y Raúl. A los profesores que siempre me inspiraron, entre ellos Itzamá, Edgardo, Citlali y Ukranio.

Tercero: A todos los demás que alguna vez les agradecí en persona pero que no mencioné arriba.

Cuarto: Libros, vídeos, podcasts, gente del internet, gente famosa y demás que jamás leerán esto.

Feregrino Bolaños Antonio

Agradezco primeramente a Dios y a la vida por permitirme llegar a concluir esta etapa, te agradezco señor por la familia tan hermosa que me diste sin duda sin su apoyo no hubiera logrado muchos de mis objetivos.

También agradezco a mi mami, la señora Virginia Evia por estar a mi lado a cada instante de mi vida por nunca dejarme y siempre ser mi pilar, mi apoyo y mi soporte, madre el día que leas esto ten por seguro que fue gracias a ti como cada uno de mis logros, te amo con todo mi corazón.

A mi hermano Beto que siempre ha estado a mi lado y siempre ha confiado en mí, y que gracias a su admiración he logrado esto, para no fallarle y siempre quiera ser como su hermana mayor, así como el solía decir de pequeño.

A mi papá por su apoyo.

Así mismo quiero agradecer a mi familia materna que siempre desde bebé me han cuidado y protegido, y se han sentido orgullosos de mí en todo momento.

A lo largo de mi carrera conocí mucha gente que vale la pena mencionar como mis amigos de Algoritmo de baile que más que mis amigos considero la familia que yo elegí, ellos que sin importar nada están siempre conmigo apoyándome y ayudándome. Gracias a todos y cada uno.

A mi compañero y amigo Fere, el pichon quien me acompañó durante esta aventura del trabajo. Gracias amigo eres un gran amigo y una persona muy inteligente.

De igual forma una mención especial a mi amiga Raquel Crespo, a quien quiero mucho y considero una gran persona y sobre todo una gran amiga, gracias “mi peque” por tantos momentos inolvidables, tantas rizas, tantos llantos, cada palabra de aliento, cada momento a tu lado es incomparable.

Por último y no por ser menos importante, sino porque pienso cerrar con broche de oro, durante mi carrera conocí al amor de mi vida, y hablo literalmente, he pasado a su lado momentos increíbles dentro y fuera de la escuela, a su lado me siento feliz, plena y sobre todo enamorada. Siempre ha estado ahí apoyándome y compartiendo cada uno de mis triunfos y sé que muy pronto celebrare uno más a su lado, porque ya te falta poquito para titularte. Gracias por ser parte de mi vida y por dejarme formar parte de la tuya, sin duda eres lo mejor que me ha pasado, quisiera gritarle al mundo lo mucho que te amo y lo especial que eres para mí, pero todo a su tiempo.

Jaime Evia del Puerto Ariana Estefania

Índice

Índice	4
Índice de ilustraciones	13
Índice de tablas	18
Introducción.....	21
Capítulo I. Antecedentes.....	22
Problemática	22
Estado del arte	23
Situación actual.....	24
Capítulo II. Propuesta de solución	25
Propuesta.....	25
Objetivos	26
Objetivos particulares.....	26
Análisis del mercado	27
Estudio del impacto.....	29
Capítulo III. Marco teórico	49
Educación básica.....	49
Arquitectura Orientada a Servicios (SOA).....	50
Servicio Web.....	50
API.....	51
Aplicación móvil.....	52
Capítulo IV. Análisis de tecnologías de desarrollo	53
Arquitectura del servicio web	53
SOAP	53
REST.....	53
Framework y lenguaje de desarrollo	55
Recess / PHP	55
Recess fue creado por Kris Jordan, un científico computacional que lo liberó bajo la licencia MIT en GitHub [21].....	56
Play / Java.....	56
Nancy / C#.....	56

Tornado / Python	57
ASP.NET Web API / C#.....	57
Base de datos.....	57
Oracle.....	57
MySQL.....	58
SQL Server.....	58
Proveedor de plataforma web	59
Hosting propio.....	59
Microsoft Azure.....	59
Amazon Web Services.....	59
Rackspace.....	60
Capítulo V. Análisis y diseño.....	61
Diagrama inicial de clases.....	61
Requerimientos funcionales	62
Requerimientos no funcionales	69
Métricas.....	73
Estimación inicial.....	73
Reconciliación de métricas	75
Estimación empleando COCOMO Básico	76
Costo de desarrollo.....	76
Análisis de riesgos	77
Arquitectura del proyecto.....	82
Usuarios	83
Dispositivos	83
Casos de uso	84
General.....	84
Casos de uso acceso	85
Casos de uso gestión escolar.....	86
Casos de uso módulo de encuestas.....	92
Casos de uso módulo de eventos.....	96

Casos de uso módulo de mensajería.....	99
Casos de uso módulo de notificaciones.....	101
Diseño de la base de datos.....	104
Diseño lógico de la base de datos	104
Diseño físico de la base de datos.....	105
Diccionario de datos.....	106
Diagramas de clases.....	113
Edutor.Web.Api.Models.....	113
Edutor.Web.Api	114
Diagrama de paquetes	115
Edutor.Common	115
Edutor.Web.Common.....	116
Edutor.Data	116
Edutor.Data.SqlServer	117
Edutor.Web.Api.Models.....	117
Edutor.Web.Api	118
Diagrama de secuencia.....	118
Diagrama de Secuencia para Acceso.....	119
Diagrama de Secuencia para Alta de alumno	119
Diagrama de Secuencia para Modificar Alumno	120
Diagrama de Secuencia para Eliminar un Alumno	121
Diagrama de Secuencia para Añadir un Profesor	121
Diagrama de Secuencia para Modificar un Profesor.....	122
Diagrama de Secuencia para Eliminar un Profesor.....	122
Diagrama de Secuencia para Añadir un Grupo.....	123
Diagrama de Secuencia para Modificar un Grupo	123
Diagrama de Secuencia para Eliminar un Grupo.....	124

Diagrama de Secuencia para Vincular un Profesor con un Grupo	125
Diagrama de Secuencia para Desvincular un Profesor de un Grupo	125
Diagrama de Secuencia para Vincular un Alumno con un Grupo	126
Diagrama de Secuencia para Desvincular un Alumno de un Grupo.....	126
Diagrama de Secuencia para Generar Encuesta.....	127
Diagrama de Secuencia para Asignar Duración a una Encuesta	127
Diagrama de Secuencia para Añadir Respuesta.....	128
Diagrama de Secuencia para Elegir un Grupo	128
Diagrama de Secuencia para Consultar Resultados	129
Diagrama de Secuencia para Contestar Encuesta	129
Diagrama de Secuencia para Crear un Evento.....	130
Diagrama de Secuencia para Asignar Fecha a Evento	131
Diagrama de Secuencia para Elegir Tipo de Evento.....	131
Diagrama de Secuencia para Elegir Invitados al Evento.....	132
Diagrama de Secuencia para Responder Invitación a Evento	132
Diagrama de Secuencia para Enviar Mensaje a un Tutor.....	133
Diagrama de Secuencia para Enviar Mensaje a un Profesor	133
Diagrama de Secuencia para Elegir un Tutor Receptor	134
Diagrama de Secuencia para Elegir un Profesor Receptor.....	134
Diagrama de Secuencia para Establecer comunicación	135
Diagrama de Secuencia para Notificar Mensaje	135
Diagrama de Secuencia para Enviar Notificación como Escuela	135
Diagrama de Secuencia para Enviar Notificación como Profesor	136
Diagrama de Secuencia para Elegir un Receptor	136
Diagrama de Secuencia para Notificaciones Recibidas	137
Diagramas de Actividades.....	138

Diagrama de Actividades para Acceso.....	138
Diagrama de Actividades para Alta de alumno	139
Diagrama de Actividades para Modificar alumno	140
Diagrama de Actividades para Eliminar alumno	141
Diagrama de Actividades para Alta de profesor	142
Diagrama de Actividades para Modificar profesor	143
Diagrama de Actividades para Eliminar profesor	144
Diagrama de Actividades para Añadir un Grupo.....	145
Diagrama de Actividades para Modificar un Grupo	146
Diagrama de Actividades para Eliminar Grupo	147
Diagrama de Actividades para Vincular un Profesor con un Grupo.....	148
Diagrama de Actividades para Desvincular un Profesor de un Grupo.....	149
Diagrama de Actividades para Vincular un Alumno a un Grupo	150
Diagrama de Actividades para Desvincular un Alumno de un Grupo	151
Diagrama de Actividades para Generar Encuestas	152
Diagrama de Actividades para Asignar Duración a Encuesta	153
Diagrama de Actividades para Añadir Respuesta.....	154
Diagrama de Actividades para Elegir un Grupo	155
Diagrama de Actividades para Consulta de Resultados	156
Diagrama de Actividades para Contestar Encuesta.....	157
Diagrama de Actividades para Crear Evento.....	158
Diagrama de Actividades para Asignar Fecha al Evento	159
Diagrama de Actividades para Elegir Tipo de Evento	160
Diagrama de Actividades para Elegir Invitados al Evento	161
Diagrama de Actividades para Responder Invitación a Evento.....	162
Diagrama de Actividades para Enviar Mensaje a Tutor.....	163

Diagrama de Actividades para Enviar Mensaje a Profesor	164
Diagrama de Actividades para Notificar Mensaje	165
Diagrama de Actividades para Enviar Notificación como Escuela	166
Diagrama de Actividades para Enviar Notificación como Profesor	167
Diagrama de Actividades para Notificaciones Recibidas.....	168
Capítulo VI. Desarrollo de la API	169
Entorno de desarrollo	169
Proyectos de base de datos	169
Proyectos de librerías de clases.....	170
Proyecto de aplicación web	170
Arquitectura del sistema.....	170
Modelo	171
Edutor.Common	171
Edutor.Data	172
Edutor.Data.SqlServer	172
Controlador.....	173
Edutor.Web.Common.....	173
Edutor.Web.Api.Models.....	174
Edutor.Web.Api	178
Librería cliente de Edutor.....	185
Capítulo VI. Implementación y pruebas	186
Implementación.....	186
Servidor Web.....	186
Servidor de base de datos	187
Servidor Web para la etapa de desarrollo.....	187
Servidor de bases de datos para la etapa de desarrollo.....	188
Aprovisionamiento de los servidores necesarios	189
Creación de la base de datos.....	192
Gestión de roles a nivel de la base de datos.....	193

Conexión con la base de datos.....	194
Publicación de la aplicación web.....	194
Resumen de implementación del sistema	196
Pruebas de la API de Edutor	197
Herramientas	197
Fiddler.....	197
Peticiónes a través de Fiddler	197
Grupos	197
Aplicación administrativa	214
Resultados.....	223
Conclusiones.....	237
Trabajo a futuro	239
Referencias	241
Glosario	245

Índice de ilustraciones

Ilustración 1 Gráfica 1 Cantidad de alumnos por grupo	29
Ilustración 2 Gráfica 2 Problemáticas más frecuentes con los alumnos	30
Ilustración 3 Gráfica 3 Problemáticas más frecuentes con los alumnos	31
Ilustración 4 Gráfica 4 Cantidad de reuniones padre-profesor al año	32
Ilustración 5 Gráfica 5 Formato de las reuniones padre-profesor	33
Ilustración 6 Gráfica 6 Temática de las reuniones padre-profesor	35
Ilustración 7 Gráfica 7 Temática de las reuniones padre-profesor	36
Ilustración 8 Gráfica 8 Razones por las que no convoca a reuniones.....	37
Ilustración 9 Gráfica 9 Interés en convocar más reuniones	38
Ilustración 10 Gráfica 10 Flexibilidad para convocar reuniones.....	39
Ilustración 11 Gráfica 11 Información crucial sobre el alumno en el aprendizaje.....	40
Ilustración 12 Gráfica 12 Frecuencia de tareas	41
Ilustración 13 Gráfica 13 Requisitos de las tareas.....	42
Ilustración 14 Gráfica 14 Situación de la relación padre-profesor	43
Ilustración 15 Gráfica 15 Vías de comunicación padre-profesor	44
Ilustración 16 Gráfica 16 Vías de comunicación padre-profesor	45
Ilustración 17 Gráfica 17 Vías de comunicación padre-profesor	46
Ilustración 18 Gráfica 18 Vías de comunicación padre-profesor	47
Ilustración 19 Gráfica 19 Vías alternativas de comunicación	48
Ilustración 20 Diagrama inicial de clases.....	61
Ilustración 21 Diagrama de la arquitectura del proyecto	82
Ilustración 22 Diagrama del caso de uso general	84
Ilustración 23 Diagrama de caso de uso acceso	85
Ilustración 24 Diagrama de caso de uso gestión escolar.....	86
Ilustración 25 Diagrama de los casos de uso Encuestas	92
Ilustración 26 Diagrama de casos de uso del módulo de eventos	96
Ilustración 27 Diagrama de casos de uso del módulo de mensajería.....	99
Ilustración 28 Diagrama de casos de uso del módulo de notificaciones.....	101
Ilustración 29 Diagrama de clases para Edutor.Web.Api.Models.....	113
Ilustración 30 Diagrama de clases para Edutor.Web.Api	114
Ilustración 31 Diagrama de paquetes general.....	115

Ilustración 32 Diagrama de paquetes para Edutor.Common.....	115
Ilustración 33 Diagrama de paquetes para Edutor.Web.Common	116
Ilustración 34 Diagrama de paquetes para Edutor.Data.....	116
Ilustración 35 Diagrama de paquetes para Edutor.Data.SqlServer	117
Ilustración 36 Diagrama de paquetes para Edutor.Web.Api.Models	117
Ilustración 37 Diagrama de paquetes para Edutor.Web.Api.....	118
Ilustración 38 Diagrama de secuencia para Acceso.....	119
Ilustración 39 Diagrama de secuencia para ingresar alumno	120
Ilustración 40 Diagrama de secuencia para Modificar Alumno.....	120
Ilustración 41 Diagrama de secuencia para Eliminar un Alumno	121
Ilustración 42 Diagrama de secuencia para Añadir un Profesor	121
Ilustración 43 Diagrama de secuencia para Modificar un Profesor.....	122
Ilustración 44 Diagrama de secuencia para Eliminar un Profesor.....	123
Ilustración 45 Diagrama de secuencia para Añadir un Grupo.....	123
Ilustración 46 Diagrama de secuencia para Modificar un Grupo.....	124
Ilustración 47 Diagrama de secuencia para Eliminar un Grupo.....	124
Ilustración 48 Diagrama de secuencia para Vincular un Profesor con un Grupo	125
Ilustración 49 Diagrama de secuencia para Desvincular un Profesor de un Grupo	126
Ilustración 50 Diagrama de secuencia para Vincular un Alumno con un Grupo.....	126
Ilustración 51 Diagrama de secuencia para Desvincular un Alumno de un Grupo.....	127
Ilustración 52 Diagrama de secuencia para Generar Encuesta.....	127
Ilustración 53 Diagrama de secuencia para Asignar Duración a una Encuesta	128
Ilustración 54 Diagrama de secuencia para Añadir Respuesta.....	128
Ilustración 55 Diagrama de secuencia para Elegir un Grupo.....	129
Ilustración 56 Diagrama de secuencia para Consultar Resultados	129
Ilustración 57 Diagrama de secuencia para Responder una Pregunta	130
Ilustración 58 Diagrama de secuencia para Crear un Evento.....	130
Ilustración 59 Diagrama de secuencia para Asignar Fecha a Evento	131
Ilustración 60 Diagrama de secuencia para Elegir Tipo de Evento.....	131
Ilustración 61 Diagrama de secuencia para Elegir Invitados al Evento.....	132
Ilustración 62 Diagrama de secuencia para Responder Invitación a Evento	132
Ilustración 63 Diagrama de secuencia para Enviar mensaje a Tutor.....	133

Ilustración 64 Diagrama de secuencia para Enviar mensaje a un Profesor.....	133
Ilustración 65 Diagrama de secuencia para elegir a un tutor receptor.....	134
Ilustración 66 Diagrama de secuencia para elegir a un profesor receptor	134
Ilustración 67 Diagrama de secuencia para establecer comunicación	135
Ilustración 68 Diagrama de secuencia para notificar mensaje	135
Ilustración 69 diagrama de secuencia para enviar notificación como escuela	136
Ilustración 70 Diagrama de secuencia para enviar notificación como profesor.....	136
Ilustración 71 Diagrama de secuencia para elegir un receptor.....	137
Ilustración 72 Diagrama de secuencia para notificaciones recibidas	137
Ilustración 73 Diagrama de actividades para Acceso	138
Ilustración 74 Diagrama de actividades para alta de alumno.....	139
Ilustración 75 Diagrama de actividades para modificar alumno	140
Ilustración 76 Diagrama de actividades para eliminar alumno	141
Ilustración 77 Diagrama de actividades para alta de profesor.....	142
Ilustración 78 Diagrama de actividades para modificar profesor.....	143
Ilustración 79Diagrama de Actividades para Eliminar profesor	144
Ilustración 80Diagrama de Actividades para Añadir un Grupo	145
Ilustración 81Diagrama de Actividades para Modificar un Grupo	146
Ilustración 82Diagrama de Actividades para Eliminar Grupo	147
Ilustración 83Diagrama de Actividades para Vincular un Profesor con un Grupo	148
Ilustración 84Diagrama de Actividades para Desvincular un Profesor de un Grupo.....	149
Ilustración 85Diagrama de Actividades para Vincular un Alumno a un Grupo	150
Ilustración 86Diagrama de Actividades para Desvincular un Alumno de un Grupo	151
Ilustración 87Diagrama de Actividades para Generar Encuestas.....	152
Ilustración 88Diagrama de Actividades para Asignar Duración a Encuesta	153
Ilustración 89Diagrama de Actividades para Añadir Respuesta	154
Ilustración 90Diagrama de Actividades para Elegir un Grupo	155
Ilustración 91Diagrama de Actividades para Consulta de Resultados	156
Ilustración 92Diagrama de Actividades para Contestar Encuesta.....	157
Ilustración 93Diagrama de Actividades para Crear Evento	158
Ilustración 94Diagrama de Actividades para Asignar Fecha al Evento	159
Ilustración 95Diagrama de Actividades para Elegir Tipo de Evento	160

Ilustración 96	Diagrama de Actividades para Elegir Invitados al Evento	161
Ilustración 97	Diagrama de Actividades para Responder Invitación a Evento	162
Ilustración 98	Diagrama de Actividades para Enviar Mensaje a Tutor	163
Ilustración 99	Diagrama de Actividades para Enviar Mensaje a Profesor	164
Ilustración 100	Diagrama de Actividades para Notificar Mensaje.....	165
Ilustración 101	Diagrama de Actividades para Enviar Notificación como Escuela.....	166
Ilustración 102	Diagrama de Actividades para Enviar Notificación como Profesor	167
Ilustración 103	Diagrama de Actividades para Notificaciones Recibidas	168
Ilustración 104	Proyectos que componen la aplicación	171
Ilustración 105	Clase Constants.cs	171
Ilustración 106	Ejemplo de mapeo ORM.....	173
Ilustración 107	Código para manejar los errores.....	174
Ilustración 108	Composición del proyecto Models	175
Ilustración 109	Comparativa entre clases	175
Ilustración 110	Fragmento de la clase NewUser	177
Ilustración 111	Fragmento de la clase NewTutor.....	178
Ilustración 112	Composición del controlador	179
Ilustración 113	Fragmento de la clase QuestionsController	180
Ilustración 114	Código del método AddQuestion	180
Ilustración 115	Código del método GetQuestion	181
Ilustración 116	Código del método CanHandleAuth.....	183
Ilustración 117	Fragmento de código para la asignación de roles.....	183
Ilustración 118	Fragmento de código para la creación de paginación.....	184
Ilustración 119	Creación de un servidor de base de datos	189
Ilustración 120	Configuración del servidor de base de datos.....	190
Ilustración 121	Creación de un servidor de aplicaciones web	191
Ilustración 122	Configuración del servidor de aplicaciones web.....	191
Ilustración 123	Script de información post-creación de base de datos.....	192
Ilustración 124	Publicación del proyecto de base de datos.....	193
Ilustración 125	Tablas Users y Students	193
Ilustración 126	Obtención del perfil de publicación.....	195
Ilustración 127	Publicación del proyecto de aplicación web	195

Ilustración 128 Importar perfil de publicación.....	195
Ilustración 129 Solicitud de la ruta de los archivos para carga	216
Ilustración 130 Peticiones al extremo Tutors	216
Ilustración 131 Peticiones al extremo SchoolUsers.....	217
Ilustración 132 Peticiones al extremo Students	217
Ilustración 133 Peticiones al extremo Groups.....	218
Ilustración 134 Peticiones al extremo Groups para vincular alumnos.....	219
Ilustración 135 Solicitud para cargar datos de muestra	221
Ilustración 136 Peticiones al extremo Events.....	222
Ilustración 137 Peticiones el extremo Notifications	222
Ilustración 138 Peticiones al extremo Questions.....	222
Ilustración 139 Pantalla de ingreso de token.....	223
Ilustración 140 Pantalla de ingreso de token, con vista del teclado nativo.....	224
Ilustración 141 Confirmación de estudiante añadido	224
Ilustración 142 Lista de notificaciones	225
Ilustración 143 Pantalla de notificación.....	225
Ilustración 144 Lista de eventos	226
Ilustración 145 Pantalla de evento	226
Ilustración 146 Control de responder a evento.....	227
Ilustración 147 Lista de preguntas.....	227
Ilustración 148 Pantalla de pregunta.....	228

Índice de tablas

Tabla 1. Requerimientos funcionales.....	63
Tabla 2 Requerimiento Funcional 1.....	64
Tabla 3 Requerimiento Funcional 2.....	64
Tabla 4 Requerimiento Funcional 3.....	64
Tabla 5 Requerimiento Funcional 4.....	65
Tabla 6 Requerimiento Funcional 5.....	65
Tabla 7 Requerimiento Funcional 6.....	65
Tabla 8 Requerimiento Funcional 7.....	66
Tabla 9 Requerimiento Funcional 8.....	66
Tabla 10 Requerimiento Funcional 9.....	66
Tabla 11 Requerimiento Funcional 10.....	67
Tabla 12 Requerimiento Funcional 11.....	67
Tabla 13 Requerimiento Funcional 12.....	67
Tabla 14 Requerimiento Funcional 13.....	68
Tabla 15 Requerimiento Funcional 14.....	68
Tabla 16 Requerimiento Funcional 15.....	68
Tabla 17 Requerimiento Funcional 16.....	69
Tabla 18 Requerimiento Funcional 17.....	69
Tabla 19 Requerimientos no funcionales.....	70
Tabla 20 Requerimiento no funcional 1.....	70
Tabla 21 Requerimiento no funcional 2.....	71
Tabla 22 Requerimiento no funcional 3.....	71
Tabla 23 Requerimiento no funcional 4.....	71
Tabla 24 Requerimiento no funcional 5.....	72
Tabla 25 Requerimiento no funcional 6.....	72
Tabla 26 Puntos de función en el sistema.....	74
Tabla 27 Descripción del factor de ajuste.....	74
Tabla 28 Cálculo del factor de ajuste.....	75
Tabla 29 Riesgos.....	77
Tabla 30 Rangos de probabilidades.....	78
Tabla 31 Descripción de los efectos de los riesgos.....	78

Tabla 32 Riesgos, probabilidades y efectos	79
Tabla 33 Estrategias para la gestión de riesgos	81
Tabla 34 Caso de uso acceso	86
Tabla 35 Caso de uso Gestionar alumnos	88
Tabla 36 Caso de uso Gestionar profesores	89
Tabla 37 Caso de uso Gestionar grupos	90
Tabla 38 Caso de uso Gestionar enlaces profesor-grupo	91
Tabla 39 Caso de uso Gestionar enlaces alumno-grupo	92
Tabla 40 Caso de uso Genera encuesta	93
Tabla 41 Caso de uso Asignar duración a encuesta	93
Tabla 42 Caso de uso Añadir respuestas	94
Tabla 43 Caso de uso Elegir grupo	94
Tabla 44 Caso de uso Consultar resultados	95
Tabla 45 Caso de uso Contestar encuesta	95
Tabla 46 Caso de uso Crear evento	97
Tabla 47 Caso de uso Asignar fecha a evento	97
Tabla 48 Caso de uso Elegir tipo de evento	97
Tabla 49 Caso de uso Elegir invitados al evento	98
Tabla 50 Caso de uso Responder invitación a evento	98
Tabla 51 Caso de uso Enviar mensaje	99
Tabla 52 Caso de uso Elegir receptor	100
Tabla 53 Caso de uso Establece comunicación	100
Tabla 54 Caso de uso Notificación mensaje	101
Tabla 55 Caso de uso Enviar notificación	102
Tabla 56 Caso de uso Elegir receptor	102
Tabla 57 Caso de uso Generar notificación	103
Tabla 58 Caso de uso Notificación recibida	103
Tabla 59 Diccionario de la tabla Users	106
Tabla 60 Diccionario de la tabla Students	107
Tabla 61 Diccionario de la tabla Groups	107
Tabla 62 Diccionario de la tabla Enrollments	108
Tabla 63 Diccionario de la tabla TeacherGroup	108

Tabla 64 Diccionario de la tabla Notifications.....	108
Tabla 65 Diccionario de la tabla NotificationDetails	109
Tabla 66 Diccionario de la tabla Events	109
Tabla 67 Diccionario de la tabla Invitations	110
Tabla 68 Diccionario de la tabla Answers	111
Tabla 69 Diccionario de la tabla Conversations.....	111
Tabla 70 Diccionario de la tabla Messages.....	112
Tabla 71 Nomenclatura de los diagramas	118
Tabla 72 Características del servidor de pruebas	186
Tabla 73 Características del servidor de base de datos.....	187
Tabla 74 Características del servidor web para la etapa de desarrollo	188
Tabla 75 Características del servidor de base de datos durante el desarrollo	188

Introducción

La actitud de un niño hacia los estudios se ve altamente influenciada por la actitud que percibe hacia ella por parte de las personas que lo rodean en su entorno. Si el menor percibe que su trabajo se ve valorado, reconocido, hay interés en su aprendizaje y se le hace saber mediante estímulos apropiados, su desempeño académico mejorará de manera significativa [1].

Por una parte tenemos que del lado de la institución escolar es el profesor el encargado de ser el referente para el alumno, mientras que de lado familiar son los tutores o persona al cargo del niño, que suelen ser en muchas ocasiones los padres.

En el ámbito académico, hay labores relativamente sencillas que los tutores ayudados por los profesores, pueden realizar para fomentar el aprendizaje del menor y así mostrar interés en él. Sin embargo, además de la implicación académica también hay otras actividades en las que los tutores se deben ver envueltos como miembros activos de la comunidad escolar.

La teoría nos dicta que lo ideal sería que las acciones, tanto académicas como no académicas, se realizaran de manera natural correcta y constante, pero en la práctica no suele ser así, y es en gran parte debido a la falta de comunicación entre tutores y profesores, situación que se puede dar por una variedad de motivos, como la falta de tiempo de los tutores, restricciones del plantel escolar, etcétera...

En el trabajo que se presenta a continuación, se expone la información referente al análisis del sistema propuesto como Trabajo Terminal, el cual busca generar un sistema que facilitará la comunicación entre profesores y tutores con la finalidad de lograr una mejora en aprovechamiento estudiantil a nivel educación básica

Capítulo I. Antecedentes

Problemática

Es común que las acciones en conjunto que se deben realizar entre tutores y profesores no se lleven a cabo por una variedad de motivos, siendo la falta de comunicación uno de los motivos principales del rezago y abandono escolar, quedándose muchas de las responsabilidades conjuntas sin ser tomadas en cuenta por ninguna de las dos partes. Varias iniciativas se han tomado, pero sin un marco de colaboración y comunicación adecuada, la puesta en marcha de un programa de este tipo solamente sirve para abrumar a directivos, profesores, tutores e inclusive alumnos.

El alejamiento de los tutores y la escuela se puede explicar en estos días debido a la cantidad de cambios que ha habido en el entorno económico y social. Las relaciones y la comunicación se ven limitadas por el tiempo disponible que tienen los tutores después de los días de trabajo, situación que afecta directamente el tiempo que estos prestan atención a la educación de sus hijos [2].

Otro aspecto que produce el distanciamiento tutor–profesor es el hecho de que la estructura tradicional de la familia se ha transformado últimamente, siendo que ahora existe una gran cantidad de variantes, como familias monoparentales o familias separadas [3]. Cambios como este resultan difíciles de adaptar a la estructura de comunicación de las escuelas al no tener conocimiento de a quién acercarse para tratar asuntos relacionados con la educación de sus alumnos.

Estado del arte

El interés por incluir a nuevos sujetos sociales dentro de la educación es reciente. Maestros y maestras siempre han sido los más favorecidos, pues se les considera actores principales. Sin embargo, un nuevo interés por dar cuenta de las particularidades y semejanzas en la construcción social de las escuelas, ha permitido entrever la actividad de los padres en estos espacios. Aun así la participación de los padres en educación no es asunto sencillo de resolver en términos históricos [7].

Desde la segunda mitad del siglo XVIII las ideas en torno a la escuela como lugar de renovación social comenzaron a circular, a su vez, se dejó de ver al niño como un mini adulto y se le comenzó a ver como un ser desprotegido. Fue entonces cuando la pregunta “¿A quién pertenece el derecho de formar a los niños?” surgió. Con las revoluciones surgidas en el siglo XIX fueron las parroquias, particulares y ayuntamientos quienes se erigieron como los actores principales en la educación.

Para el caso de las escuelas particulares, los padres desempeñaban un papel más activo que en las escuelas gratuitas, ya que al pagar de manera directa el salario del maestro ponían mayor atención en la instrucción otorgada a sus hijos, los maestros se sentían en mayor medida obligados a satisfacer las expectativas de los padres ofreciendo un servicio “extra”.

Para la mitad del siglo XIX, se integraron juntas de vigilancia integradas por políticos, representantes administrativos, recaudadores de rentas y el representante de la comunidad a los cuales se les unían dos o tres padres de familia distinguidos por su comportamiento. Estas juntas fueron el preámbulo de la Unión Nacional de Padres de Familia. Sin embargo estas no dejan de ser esfuerzos más en lo colectivo que de manera individual, durante todo ese tiempo se crearon organismos, se organizaron consejos y se realizaron reuniones para recaudar las peticiones de los padres para la educación.

Situación actual

Un rápido análisis nos permite afirmar que, hace unos pocos años, las familias contaban con elementos de solidez propios muy superiores a los actuales: mayor estabilidad, menor estrés, más miembros y mayores oportunidades de interacción entre ellos, etc. En la actualidad, las familias, a pesar de sus mejores niveles de formación y educación, están más afectadas por influencias sociales negativas propias de la sociedad occidental y son más débiles en su estructura, encontrándose inmersas, en muchos casos, en problemas reales que afectan a su estabilidad. Carencia de ideales claros de vida, dificultades de convivencia o ruptura del matrimonio, etc. Esas familias necesitan más que nunca ayuda en su acción educativa profunda, y deben encontrar colaboración en el ámbito escolar, dentro de un marco de confianza.

La peculiar relación existente entre escuela y familia, exige de ellas una exquisita coordinación. Del mismo modo, la necesidad de personalización para una verdadera formación, y la reciprocidad de la relación establecida, solicitan crecientes grados de participación y comunicación entre ambas instituciones.

Se pueden identificar algunas características de la relación Familia-Escuela y es el principio de subsidiariedad el que marca esta relación. Así mismo debemos recalcar que es la familia quien tiene el derecho-deber de la educación de los alumnos.

- Son los padres quienes tienen la posibilidad de decidir acerca de las cuestiones esenciales: más, a medida que los hijos son menores.
- Son los padres quienes eligen el centro educativo, sobre todo en las etapas de Educación Primaria y Secundaria. Ayudan a los hijos también a elegir los amigos al situarles en determinados contextos sociales, donde se entablan las relaciones de amistad.
- Son los padres quienes, como consecuencia de su estilo de vida, relaciones, conversaciones, juicios, etc., van creando una cultura familiar que es clave en todo el proceso de maduración de la persona, de tal manera que muchos de los referentes en la toma de decisiones de las personas adultas se basan en actitudes y valores adquiridos en los primeros años de vida.
- Son los padres quienes gozan de esa relación de intimidad única que exclusivamente se da en el seno de una familia y que permite todo tipo de interrelaciones personales: de afecto,

ayuda, orientación, soporte, etc., que influyen y modifican los comportamientos de todos sus miembros. Suele decirse que en una familia todos educan y son educados.

- Son, asimismo, los padres quienes están en mejores condiciones, a causa de su cariño desinteresado, de conseguir el crecimiento en autonomía de sus hijos y, por tanto, la madurez: un crecimiento en libertad y responsabilidad que solamente es posible, de manera armónica, cuando la familia soporta las decisiones personales, con su mezcla de aciertos y errores.

Capítulo II. Propuesta de solución

Propuesta

Se propone crear una plataforma basada en servicios Web mediante la cual se realizará el intercambio de información entre tutores y profesores sobre asuntos relacionados con los alumnos, a dicha plataforma se podrá acceder a través de una interfaz diseñada para ser consumida especialmente por aplicaciones desde equipos móviles con el uso de una conexión a internet.

Mediante la realización de este trabajo se pretende atacar, haciendo uso efectivo de los medios tecnológicos, varios de los puntos que dificultan la interacción profesor-tutor, los cuales se dividen en 3 niveles esenciales, según Rodrigo Vázquez Neira [4]:

- Nivel de información
- Nivel de formación
- Nivel de interacción

Edutor pretende ser una plataforma que elimine estas distancias, facilitando la interacción directa del profesor con el tutor, permitiendo la comunicación entre ambas partes, contemplando la información que algunos autores mencionan es fundamental que se debe saber del alumno y su entorno [5] [6].

De entre las características que se contemplan para Edutor se extraen algunas para explicar la vinculación que tienen con los usuarios finales:

1. **Mensajería directa profesor-tutor** – Ya sea para la realización de preguntas directas, alguna conversación acerca de algún problema con el alumno o para consultas acerca de temas académicos específicos.

2. **Realización de encuestas o sondeos** – Se puede usar este medio para conocer la opinión de los tutores acerca de algún tema de interés general, como los métodos de enseñanza, pedagógicos y hasta las instalaciones del plantel.

3. **Notificaciones** – Las notificaciones se pueden emplear para que el tutor de seguimiento a la realización de las tareas o para informar acerca de algún tema nuevo visto en clase y este a su vez lo pueda comentar con su tutorado para reforzar el conocimiento.

4. **Planeación de eventos** – El uso de Edutor no suprimirá por completo la interacción física entre las partes, por lo tanto se contempla también que a través de él se puedan planear eventos, como juntas de información general o para apoyar directamente en las labores del plantel.

Objetivos

Desarrollar una plataforma basada en servicios Web a través de la cual fluya la comunicación tutor-profesor para contribuir al fortalecimiento de esta relación gracias al vínculo directo entre ellos, consiguiendo con esto una mejoría en la educación impartida en las escuelas de nivel básico.

Objetivos particulares

- Creación de una plataforma Web operable a través de una capa de abstracción.
- Desarrollo de una aplicación cliente básica que haga uso de la API de Edutor.
- Generación del reporte técnico de Trabajo Terminal.

Análisis del mercado

Hoy día con la cantidad de actividades extraescolares, reuniones, juntas, entre otras cosas, suele resultar complicado que la comunicación entre padres y profesores se realice de una manera fluida y eficaz. Información tan importante como los libros que deben comprar, las reglas de la escuela, horarios de las clases y de las actividades de educación física, las calificaciones de los alumnos, etc. Son cuestiones que todos los padres desean conocer en un momento dado y que resulta complicado atender por parte de los profesores por el trabajo del día a día. Por ello, las nuevas tecnologías se han encontrado con educación para facilitar y propiciar este tipo de comunicación bidireccional al binomio padres – profesores y que esta sea sencilla, eficaz y confiable.

Es un hecho innegable que a estas alturas de nuestra existencia vivimos rodeados de tanta tecnología que apenas lo notamos, pero fue hace tiempo cuando vivimos la transición de las tecnologías de la información de los ámbitos militar y empresarial hacia los centros educativos y la familia.

Por un tiempo el internet y las páginas web estuvieron a la vanguardia y ofrecían llenar las necesidades de información, conectar a las personas y comunicarlas, sin embargo hoy en día las aplicaciones móviles se están convirtiendo en la principal fuente de información para los usuarios de smartphones.

Una vez realizada la investigación de las soluciones actuales, encontramos que en la actualidad, existen muy pocas soluciones dedicadas de lleno a atacar el problema que se busca afrontar con el desarrollo de este trabajo terminal. En general la mayoría simplemente son plataformas que podemos calificar como *aula en línea* que ofrecen un canal de comunicación maestro – alumno con opciones de participación limitadas para el tutor, que más bien pasa a ser un espectador. En las líneas siguientes resumimos lo que encontramos:

- **Escolapp**, es una aplicación pensada para los padres y para las escuelas. Se seleccionan los detalles de la app y esta se hace a medida. Así se intenta atender las necesidades que presentan la escuela y los padres, que reciben el máximo de información que la escuela está

en disposición de dar. A la vez, se pretende fomentar el ahorro del papel, ya que toda la información se puede consultar en el teléfono. Con esta aplicación, los padres que la descarguen recibirán en su teléfono móvil todas las notificaciones enviadas por la escuela, como un cambio de horario, una fotografía tomada en una función escolar, la fecha de una excursión y la autorización para asistir, la fecha de los exámenes y la consulta de notas o los deberes que han de hacer los hijos, entre otros datos [8].

- **Clickedu**, surgió como una plataforma en la nube para la gestión de los centros educativos, pero se ha adaptado al formato de aplicación para facilitar su uso por parte de padres, alumnos y centros en el entorno móvil. En cuanto a las familias cuyos hijos estudian en centros que utilizan Clickedu, reciben información sobre noticias, álbumes de fotos, exámenes, información del día a día en clase, calendarios, mensajes de los tutores, fechas para concertar entrevistas, boletines de notas, actividades extraescolares, menús y comedor, entre otras. La aplicación reúne datos sobre el calendario escolar, horarios, notas y fichas de alumnos, además de permitir el acceso a la versión web para disponer del resto de información mencionada [9].
- **KinderClose**, presume de ser una aplicación sencilla e intuitiva, debe ser descargada tanto por el profesor como por el padre del alumno. El profesor registra cada acción que el niño realice dentro del centro, tras lo cual el padre recibe una notificación mostrándole lo que el profesor registró. Tanto padres como profesores tienen la capacidad de ver la ficha del día de cada alumno (con la restricción de que solo los padres de un niño pueden ver lo relacionado con él). Implementa funcionalidades como mensajes, avisos permanentes además de que se almacena toda la información que en ella se registra [10].

Estudio del impacto

Para documentar este apartado se tomó en cuenta una escuela, en ella se aplicaron dos encuestas a profesores y padres de familia, para saber los medios de comunicación que utilizan actualmente así como cuales podrían utilizar si es que se tuvieran al alcance, lo que las encuestas nos arrojan como resultados en cuanto a profesores es lo siguiente:

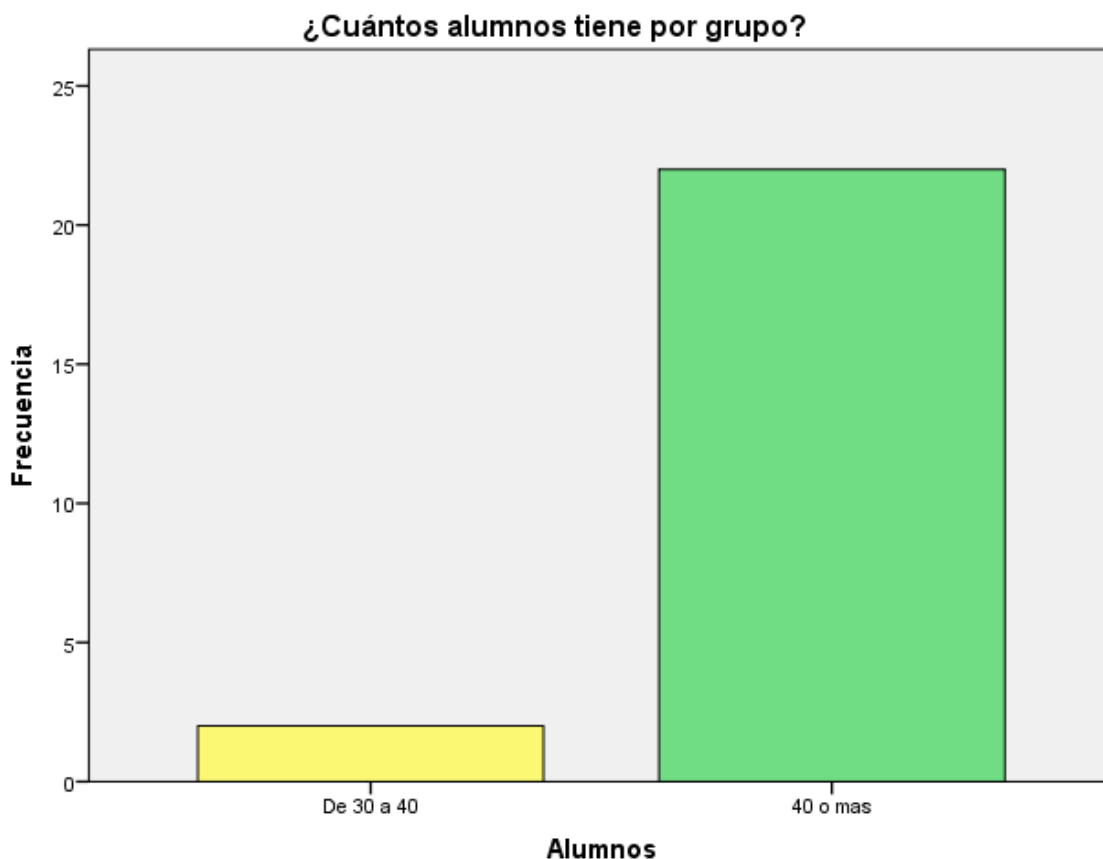


Ilustración 1 Gráfica 1 Cantidad de alumnos por grupo

Como se observa en la gráfica 1, la mayoría de los profesores entrevistados manejan grupos de más de cuarenta alumnos lo que nos indica que el profesor estará interactuando y teniendo comunicación con cuarenta tutores de un solo grupo, aunado a ello podemos decir que un maestro puede o no tener asignados varios grupos, si es así es evidente que los profesores necesitan un medio de comunicación con el cual sea más directa y personalizada dicha comunicación. Y así de esta forma poder tener un mejor aprovechamiento por parte de los alumnos y abatir el rezago y abandono escolar.

Las siguientes gráficas (gráfica 2, gráfica 3), muestran los problemas más comunes en los alumnos, esto con la finalidad de saber qué factores son los que influyen en el día a día de los alumnos y así los profesores tengan una herramienta que les ayude a mantener informados a los padres y así ellos ayuden a los profesores a combatir dichas conductas.

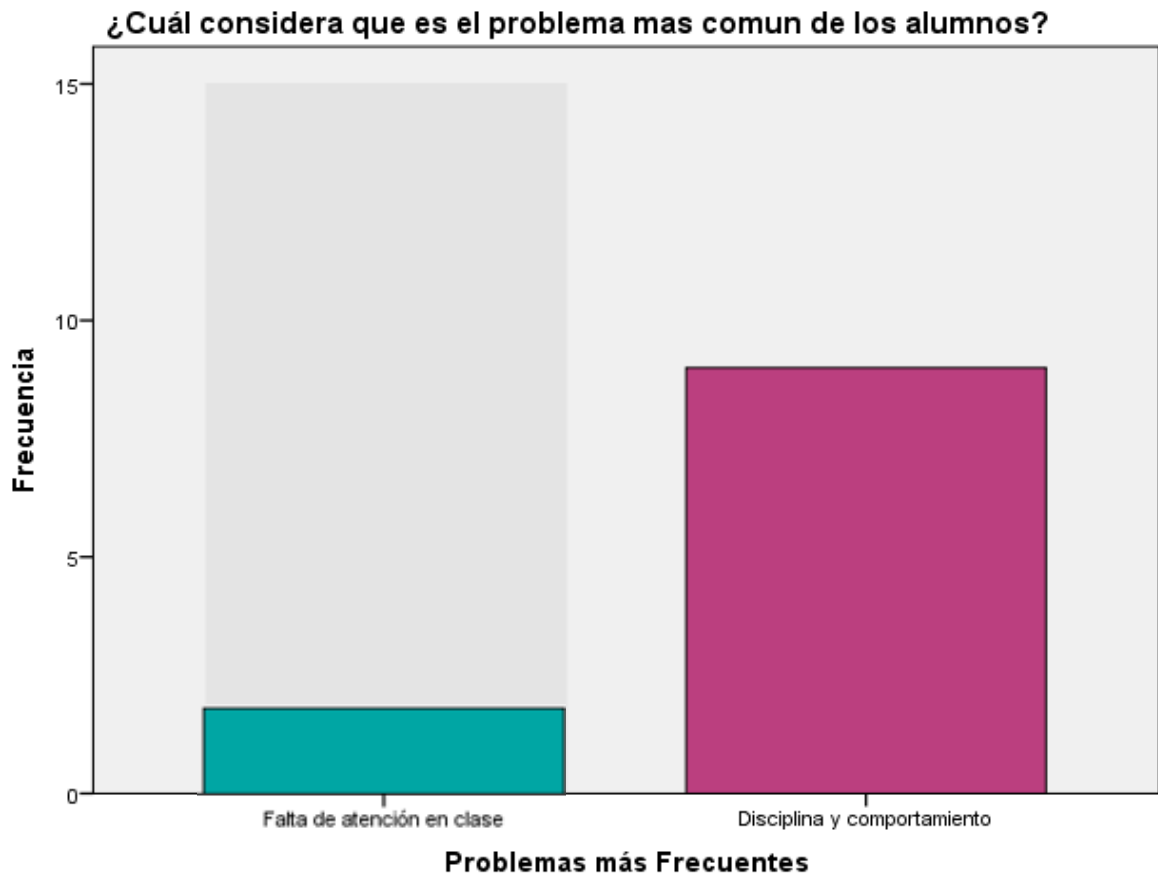


Ilustración 2 Gráfica 2 Problemáticas más frecuentes con los alumnos

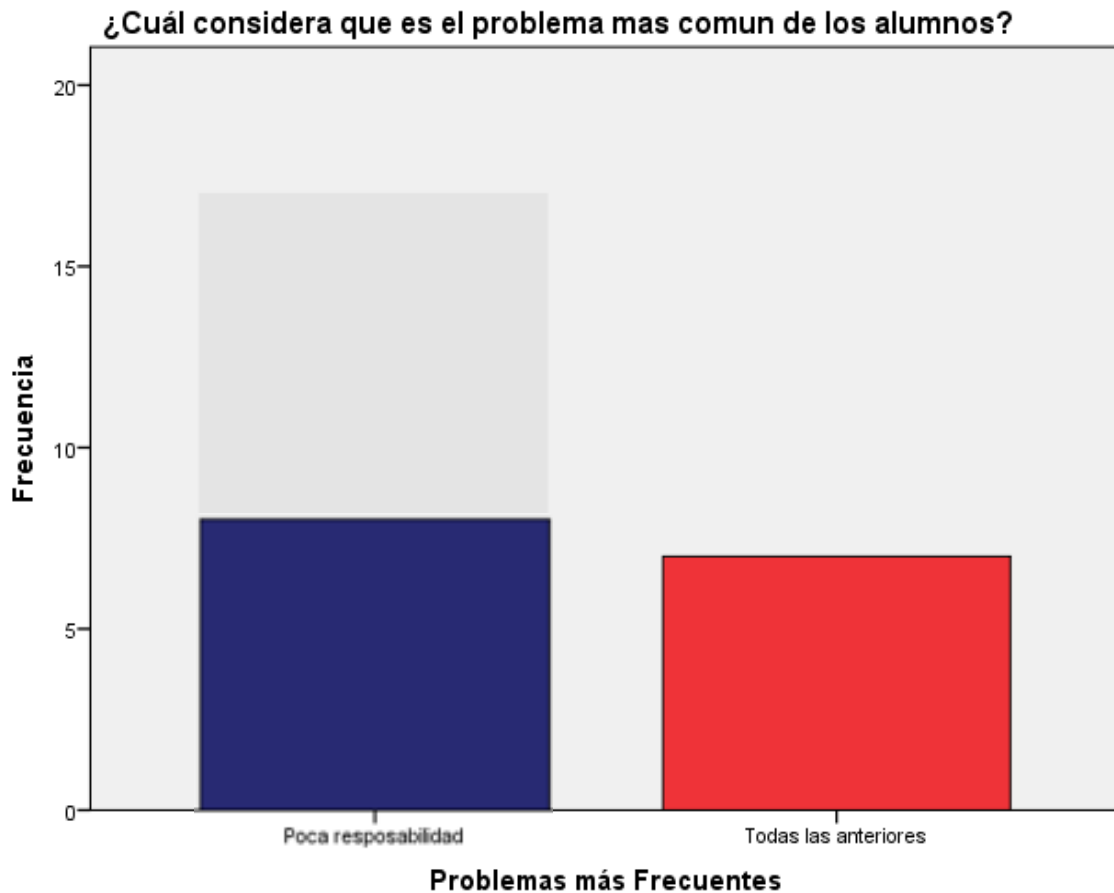


Ilustración 3 Gráfica 3 Problemáticas más frecuentes con los alumnos

Así mismo nos dimos a la tarea de investigar con qué frecuencia se cita a los padres de familia para dar a conocer este tipo de eventualidades y tratar algunos otros temas. Como se muestra en la siguiente gráfica (gráfica 4), en su mayoría los profesores citan a los padres de familia de tres a seis veces por año, estos son periodos largos tomando en cuenta que diariamente se tienen acontecimientos con los alumnos, si bien puede ser por disciplina, incumplimiento, falta de atención, entre muchas otras, es casi imposible poder dar un seguimiento teniendo citas esporádicamente como lo hacen la mayoría de las escuelas.

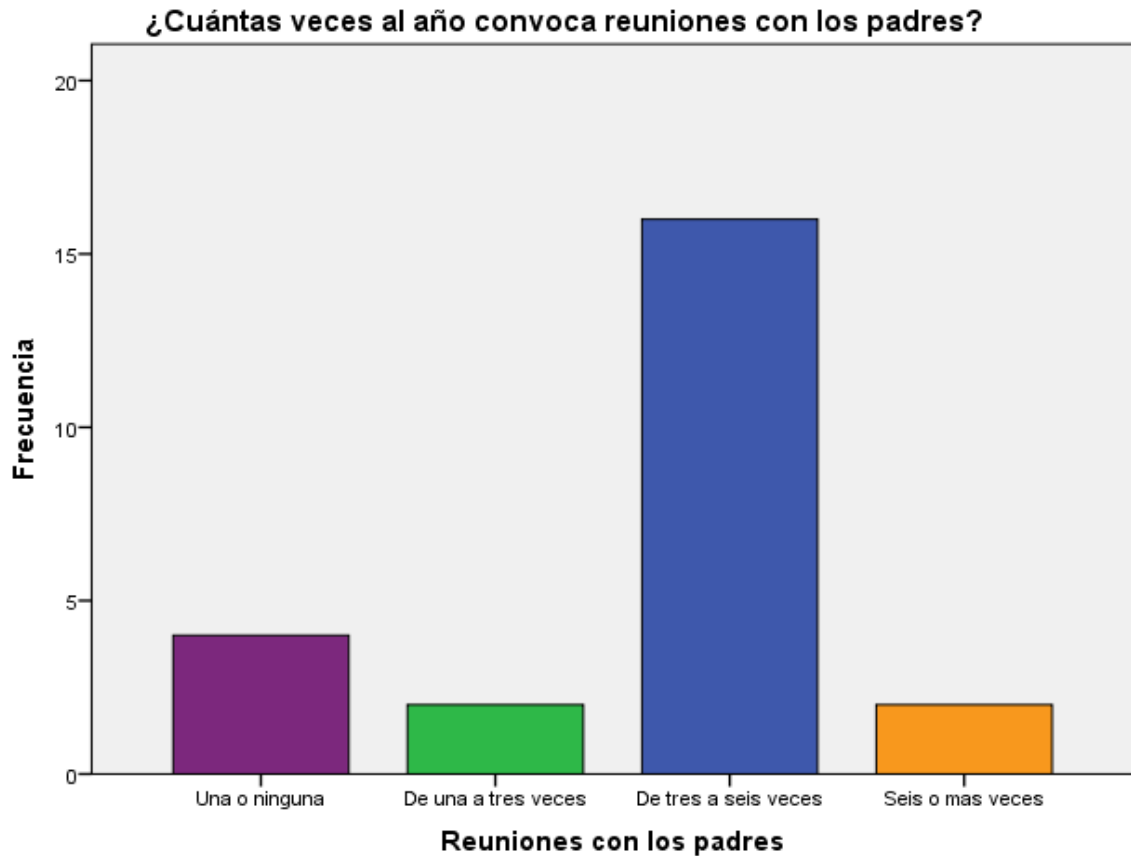


Ilustración 4 Gráfica 4 Cantidad de reuniones padre-profesor al año

Así mismo es muy difícil poder tratar estos temas por alumno en una sola reunión puesto que como se muestra en la gráfica siguiente (gráfica 5), las reuniones se realizan en su mayoría en grupo, tomemos en cuenta que son grupos grandes y muchas veces el tiempo no alcanza para tratar los asuntos de cada uno de los alumnos que pertenecen a dicho grupo, por ello sería de gran ayuda tener un medio de comunicación en el que se pueda comunicar el profesor directamente con el padre de familia, e informar la eventualidad en el instante.

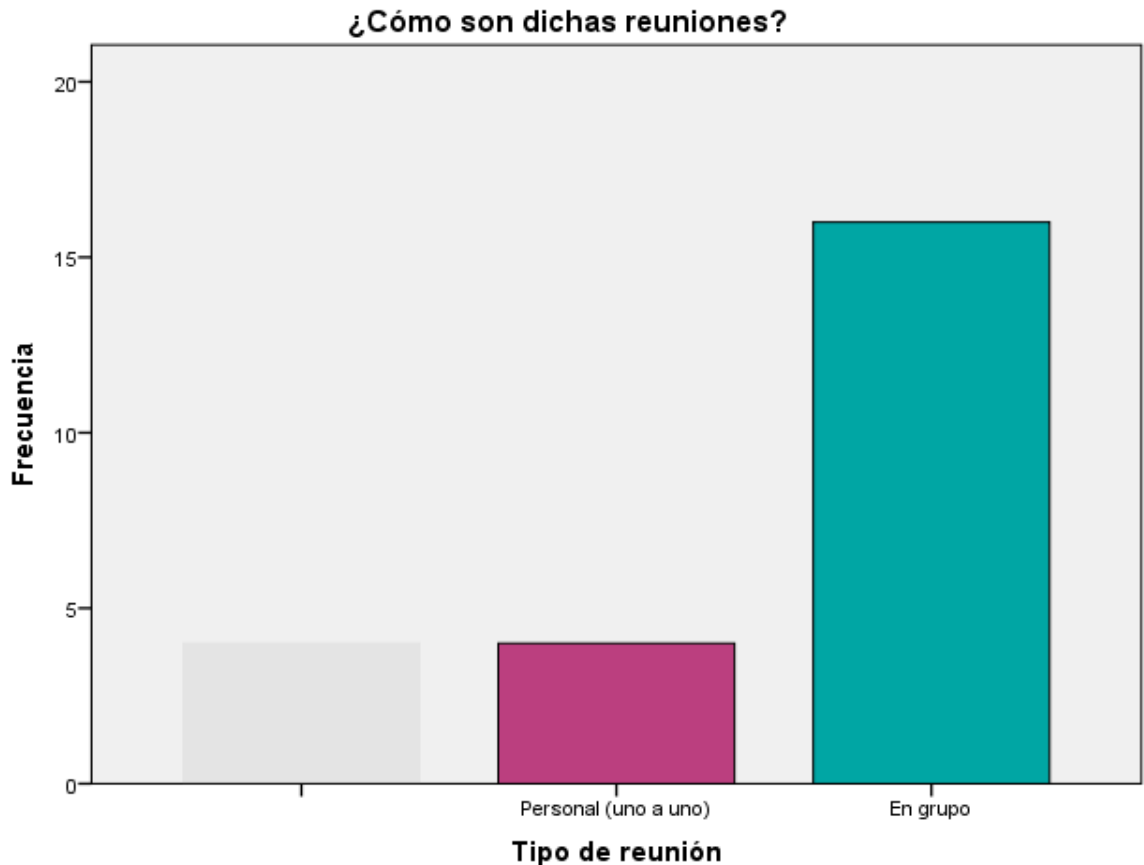


Ilustración 5 Gráfica 5 Formato de las reuniones padre-profesor

Como ya se mencionó el principal objetivo de las reuniones con los padres es informar sobre diversos temas como lo muestran las siguientes gráficas (gráfica 6, gráfica 7). Entre los temas que más destacan son:

- Comportamiento
- Calificación
- Aprovechamiento

Que coinciden con algunos de los problemas del día a día de los alumnos, como se mostró en gráficas anteriores, también se puede apreciar que los tres temas más relevantes tienen una frecuencia bastante alta lo que nos indica que la mayoría de los profesores realiza las juntas para tratar específicamente estos temas, con un medio de comunicación que les permitiera informar día a día sobre el comportamiento, aprovechamiento, temas y tareas, entre otras cosas se podría aprovechar más el tiempo y se tendría más informado a los padres de familia acerca del día a día de sus hijos y así se evitaría la fatiga de liderar con muchos padres en una sola exhibición.

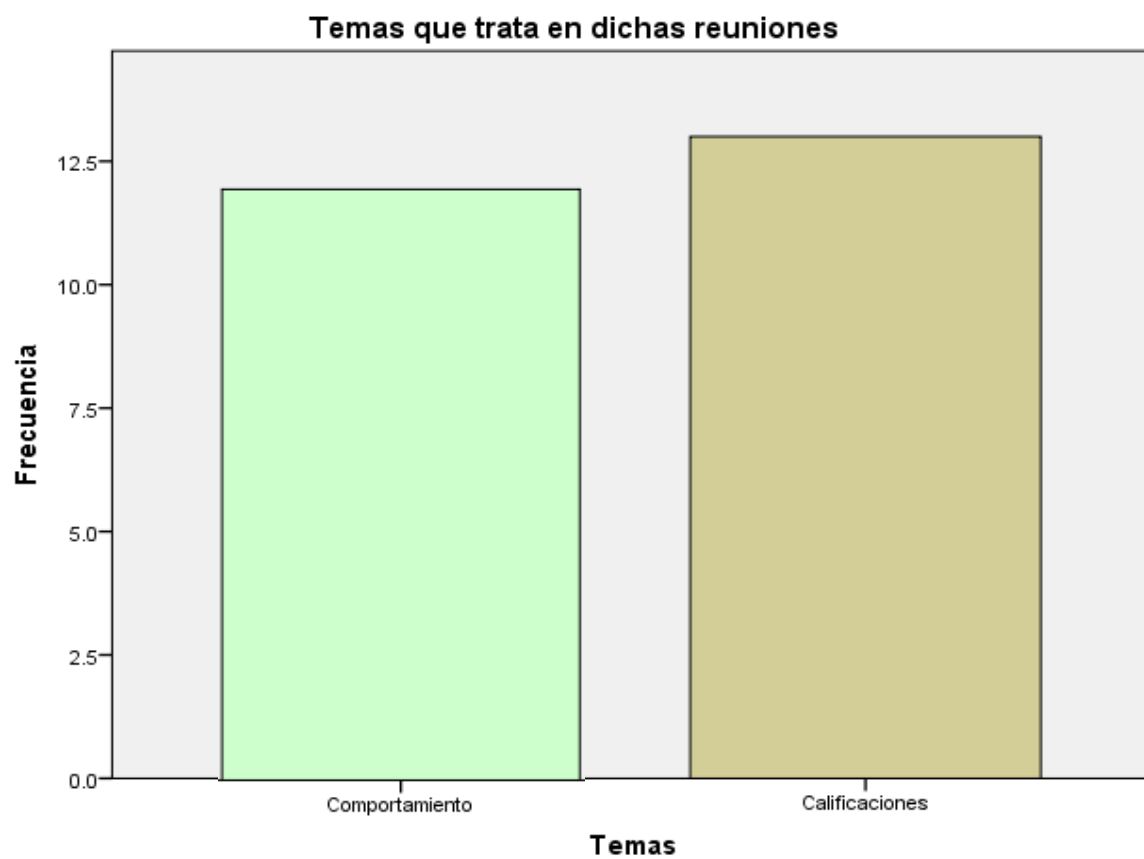


Ilustración 6 Gráfica 6 Temática de las reuniones padre-profesor

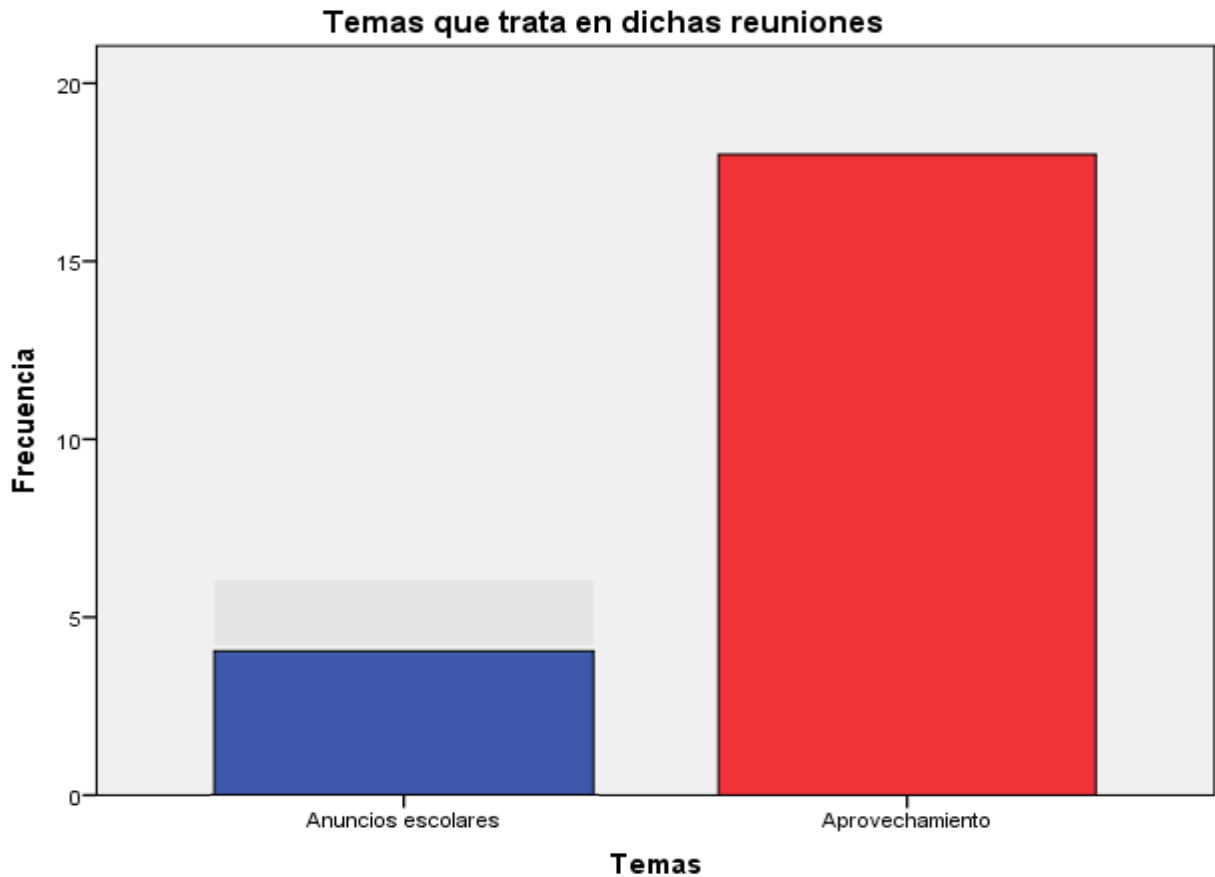


Ilustración 7 Gráfica 7 Temática de las reuniones padre-profesor

Hubo casos en los que nos encontramos que había profesores que no convocaban a reuniones, esto porque no alcanzaba el tiempo o simplemente porque no lo creían necesario, sin embargo la mayoría su contestación fue no aplica por lo cual damos por entendido que realizan reuniones con frecuencia. Tal y como lo vimos en gráficas anteriores.

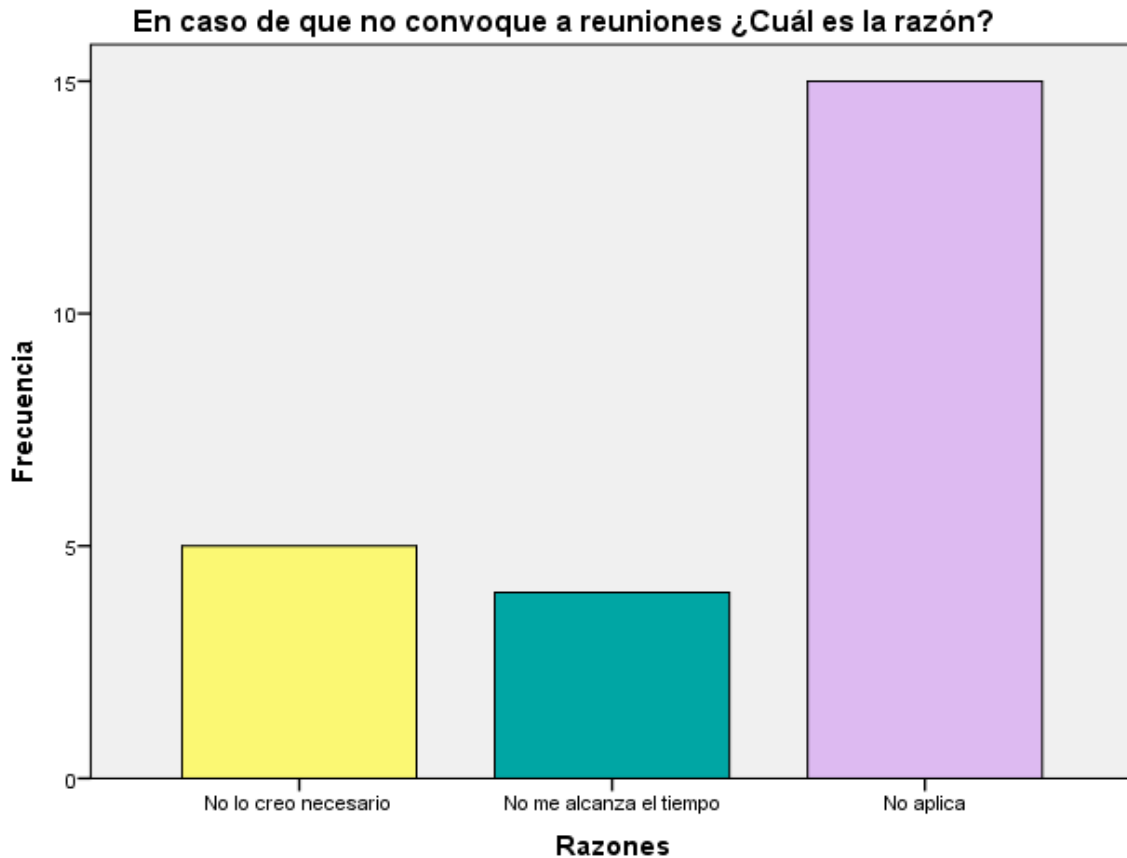


Ilustración 8 Gráfica 8 Razones por las que no convoca a reuniones

Inclusive a la mayoría de los profesores les gustaría convocar más seguido a juntas pero individualmente, esto con el objetivo de dar una atención personalizada y tratar asuntos específicamente de un alumno, esto se facilitaría con *Edutor* puesto que se podría informar el día a día de los alumnos y si es necesario agendar una cita para tratar algún punto específico pero los padres estarían enterados de la conducta, tareas, trabajos, etc. y tendría una comunicación directa con los profesores del alumno y así estar enterado del aprovechamiento de su hijo.

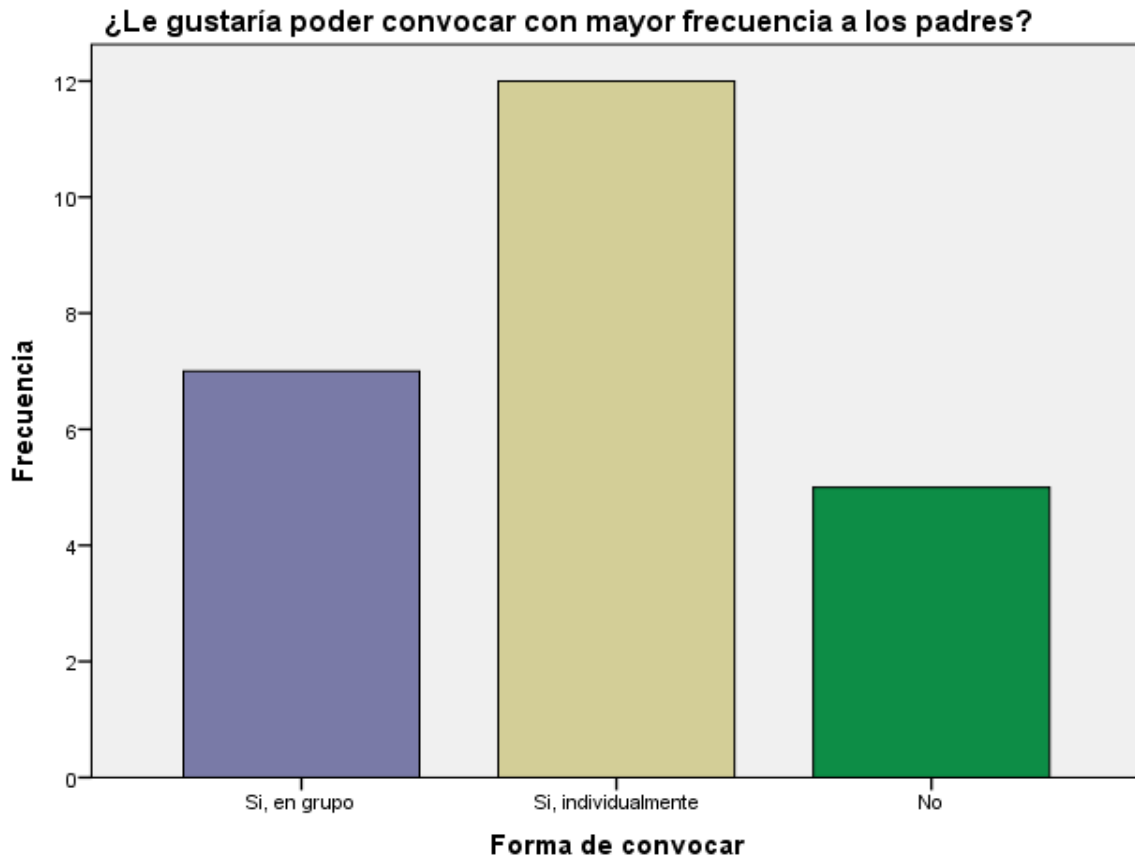


Ilustración 9 Gráfica 9 Interés en convocar más reuniones

La ventaja que se ve ante esta situación de poder convocar a juntas si es necesario es que a pesar de que existe un calendario de reuniones, se les permite a los profesores para organizar otras, como se muestra en la gráfica 10. Y como ya se mencionó previamente *Eduator* tendrá un módulo en que se podrán agendar estas reuniones extras a las establecidas en el calendario y así poder tener esa comunicación profesor-tutor.

¿Cuál es la flexibilidad de la escuela en la que trabaja con respecto a reunirse con padres?

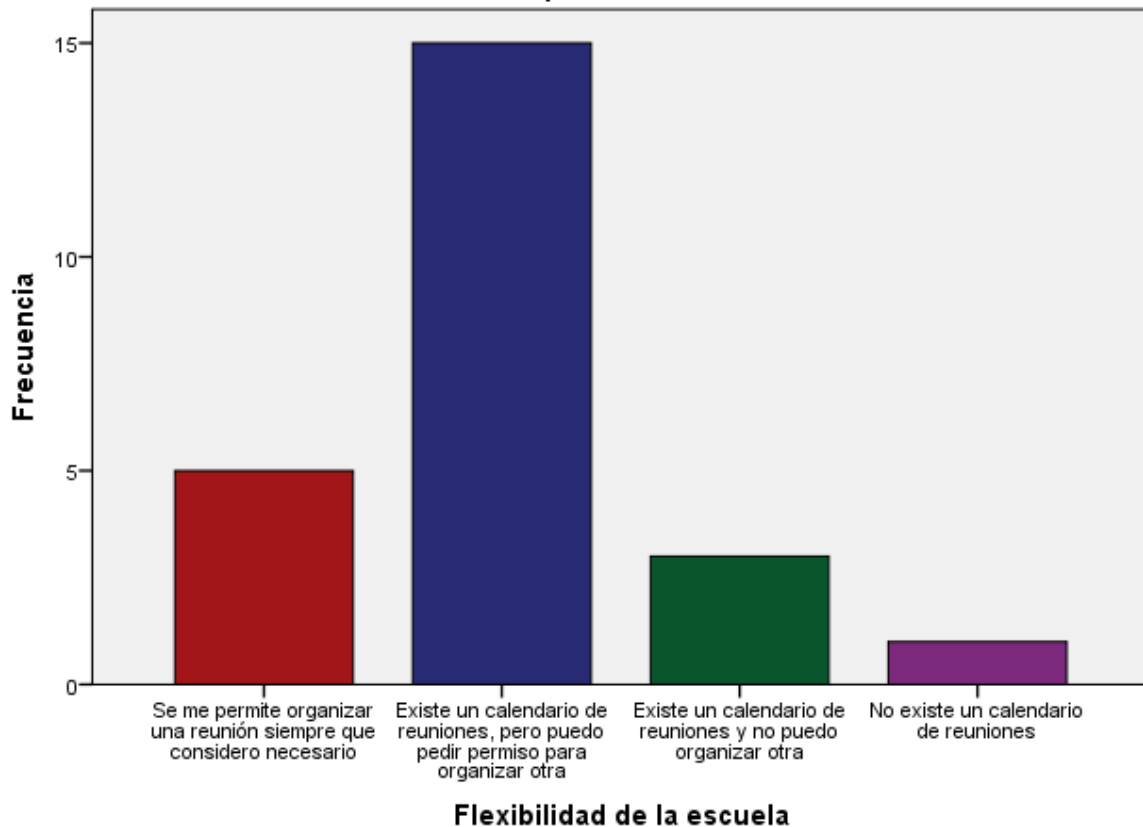


Ilustración 10 Gráfica 10 Flexibilidad para convocar reuniones

Como vemos en la gráfica 11, el trabajo en clase es uno de los aspectos más importantes del día a día de un alumno así como la asistencia, si bien sabemos que estos dos aspectos van de la mano puesto que si un alumno no asiste a clase se perderá del trabajo que se realice ese día. Muchas veces por las actividades cotidianas que tienen los padres de familia como es el ir a trabajar, no se dan cuenta si en realidad los alumnos asisten a clases o si realizan las tareas que se dejan, *Edutor* ayudara a los padres a estar en constante comunicación con los profesores y a su vez los profesores podrán informar a sus respectivos tutores si el alumno falta, si es indisciplinado, inclusive llevar una relación de los trabajos y tareas que se han realizado.

¿Qué información del día a día considera esencial para el aprendizaje de un alumno?

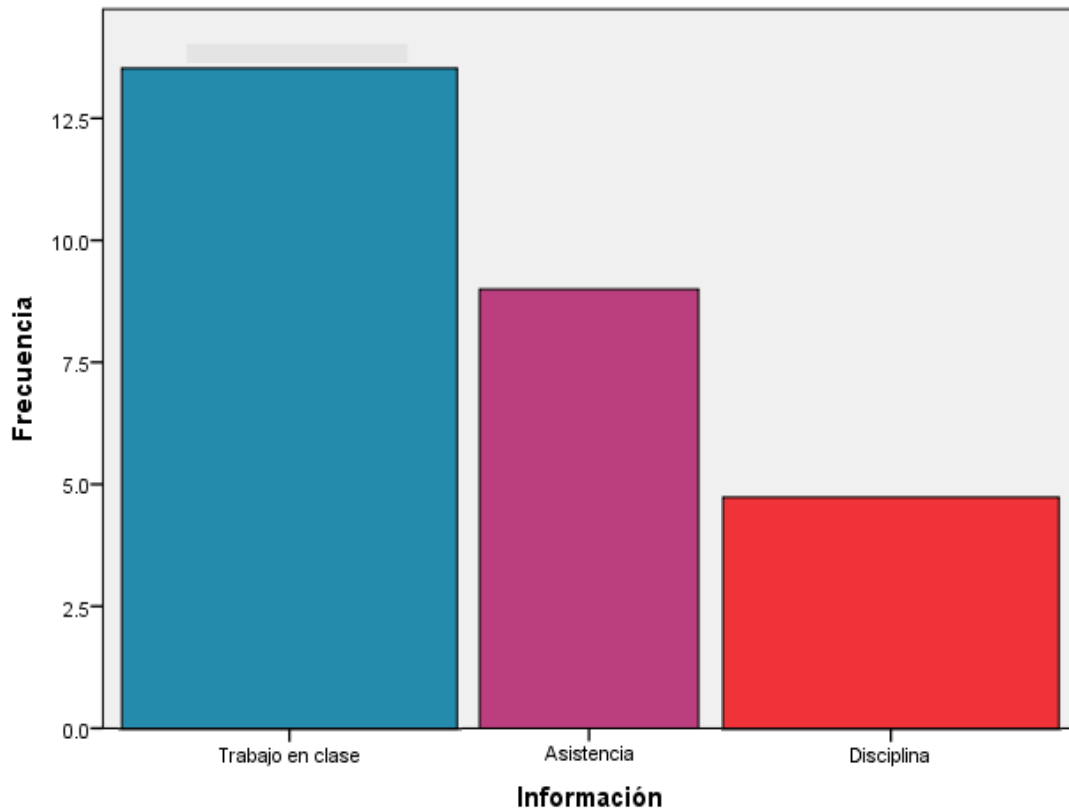


Ilustración 11 Gráfica 11 Información crucial sobre el alumno en el aprendizaje

En las siguientes gráficas (gráfica 12, gráfica 13) se observa la frecuencia con la que la mayoría de los profesores dejan tarea y las características de entrega de las mismas, esto con la finalidad de que los padres de familia sepan cada cuando se deja tarea y como deben ser entregadas.

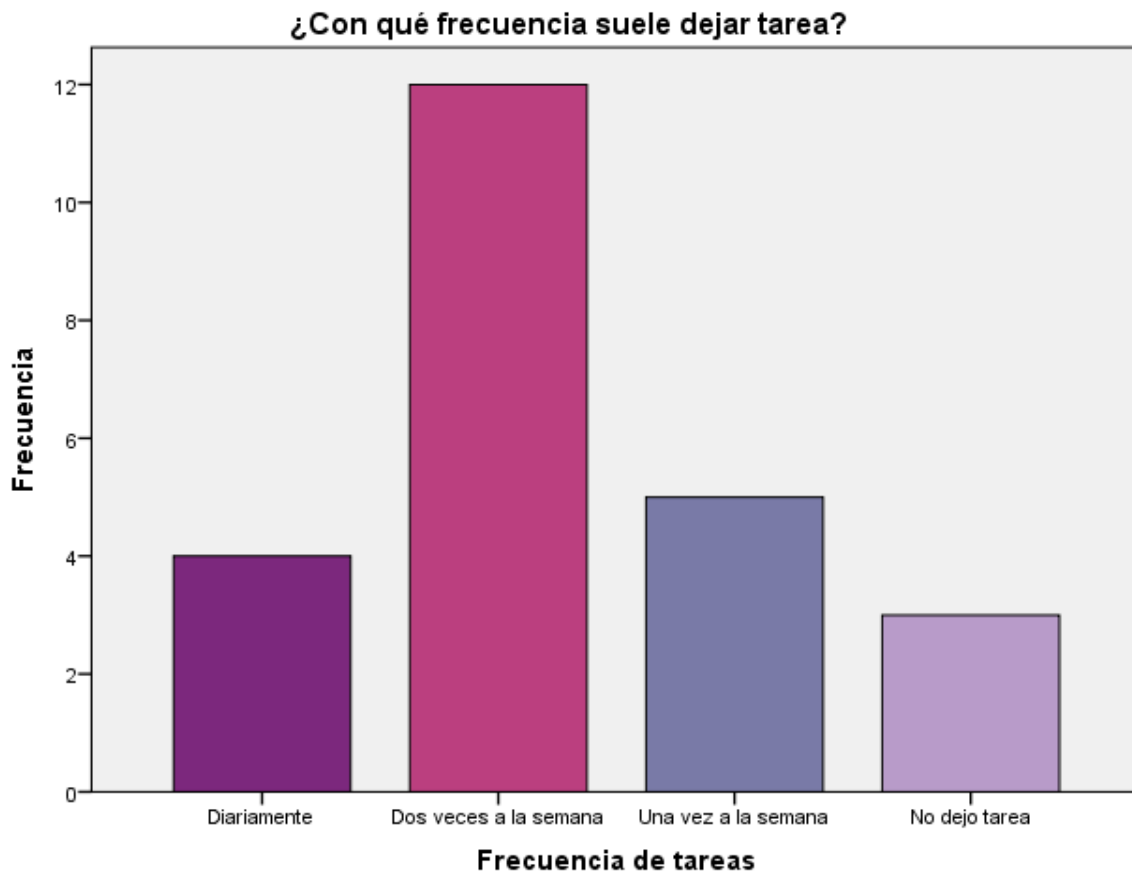


Ilustración 12 Gráfica 12 Frecuencia de tareas

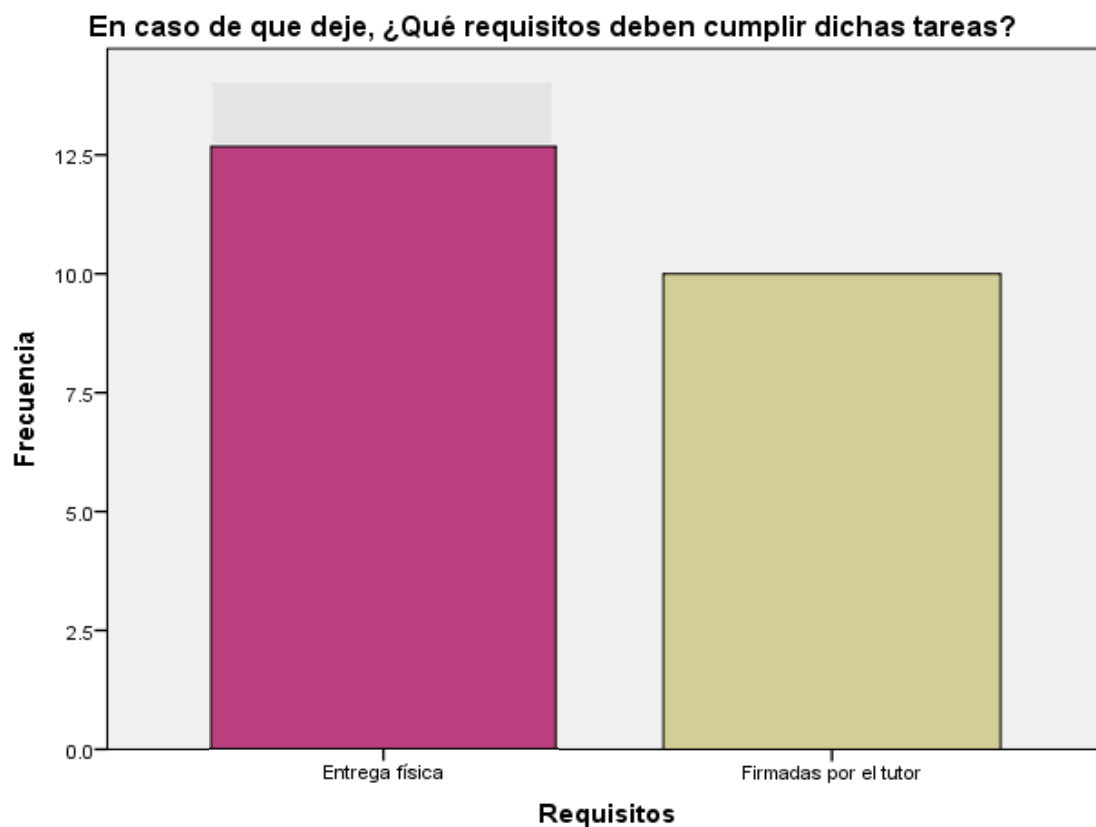


Ilustración 13 Gráfica 13 Requisitos de las tareas

Si bien es importante saber que opinan los profesores acerca de la relación que llevan con los padres de familia, lo que en su mayoría respondieron que es buena como se muestra en la gráfica 14 , lo que se podría mejorar con ayuda de *Edutor* ya que como lo dice páginas atrás su objetivo principal es: “*Desarrollar una plataforma basada en servicios Web a través de la cual fluya la comunicación tutor–profesor para contribuir al fortalecimiento de esta relación gracias al vínculo directo entre ellos, consiguiendo con esto una mejoría en la educación impartida en las escuelas de nivel básico.*”

¿Cómo considera que es la relación padre-profesor con los padres de sus alumnos?

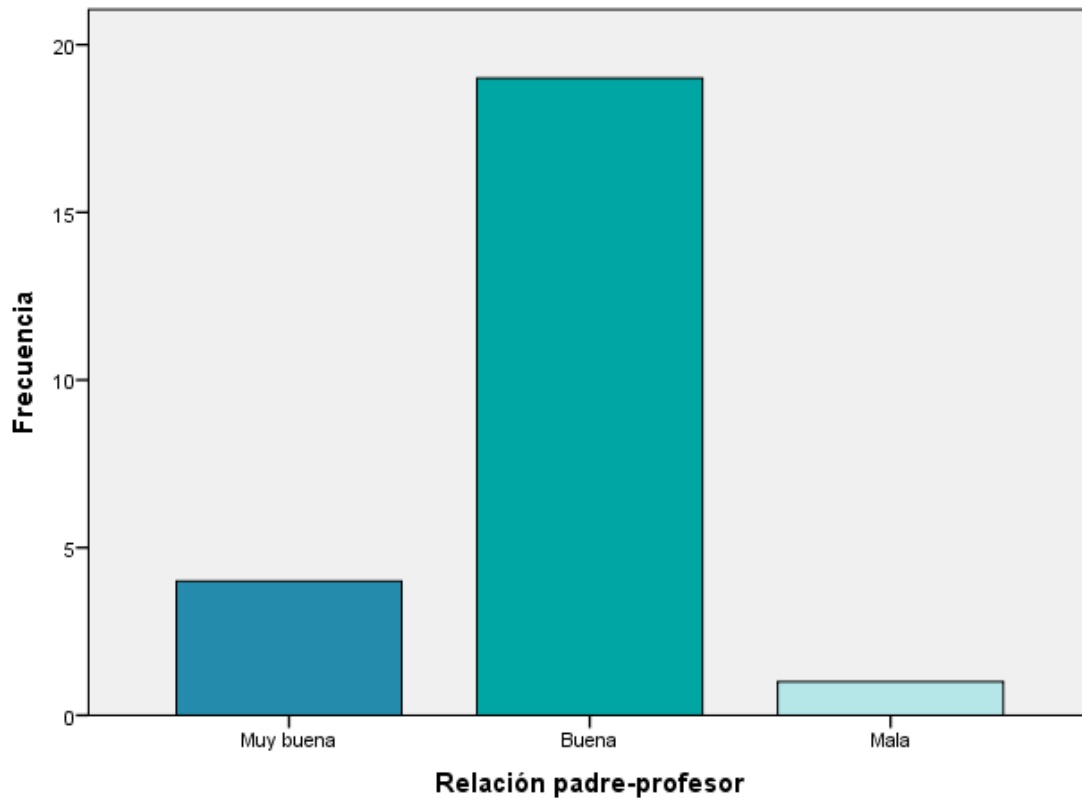


Ilustración 14 Gráfica 14 Situación de la relación padre-profesor

Se preguntaron también por los medios alternos que tenían para comunicarse con los padres aparte de las juntas que se realizan, se plantearon algunas posibles respuestas y se obtuvieron los siguientes resultados mostrados en las gráficas 15, 16, 17,18. La mayoría de los profesores respondieron

¿Aparte de las posibles reuniones organizadas por la escuela, tiene otra vía de comunicación con los padres?

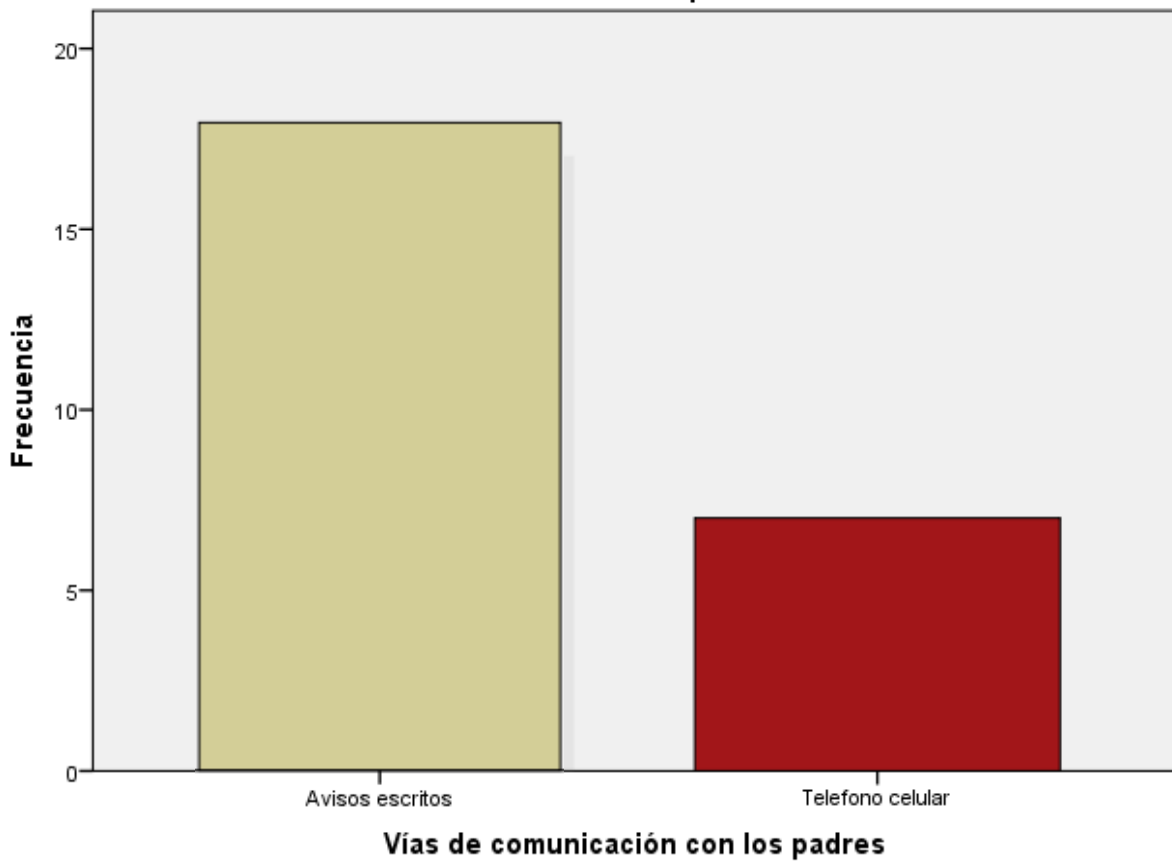
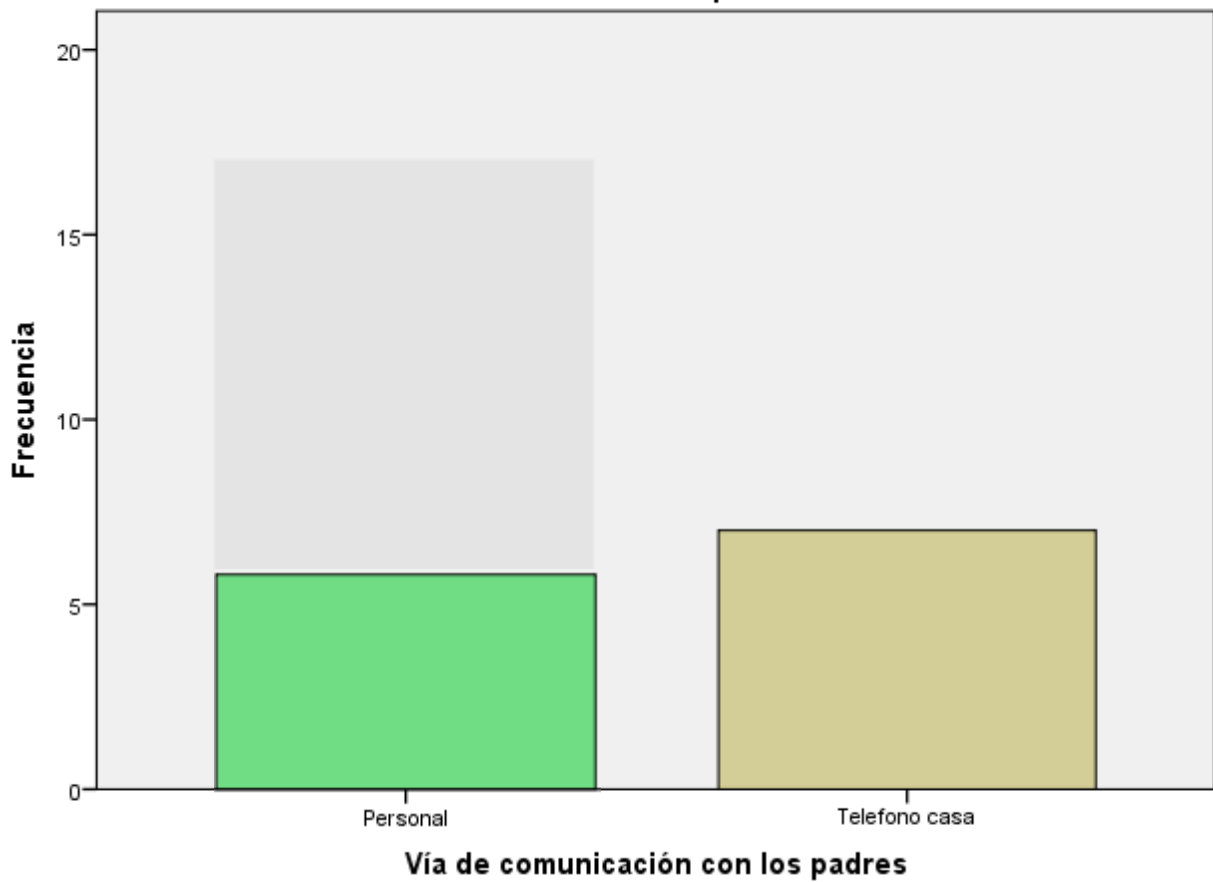


Ilustración 15 Gráfica 15 Vías de comunicación padre-profesor

¿Aparte de las posibles reuniones organizadas por la escuela, tiene otra vía de comunicación con los padres?



Gráfica 16

Ilustración 16 Gráfica 16 Vías de comunicación padre-profesor

¿Aparte de las posibles reuniones organizadas por la escuela, tiene otra vía de comunicación con los padres?

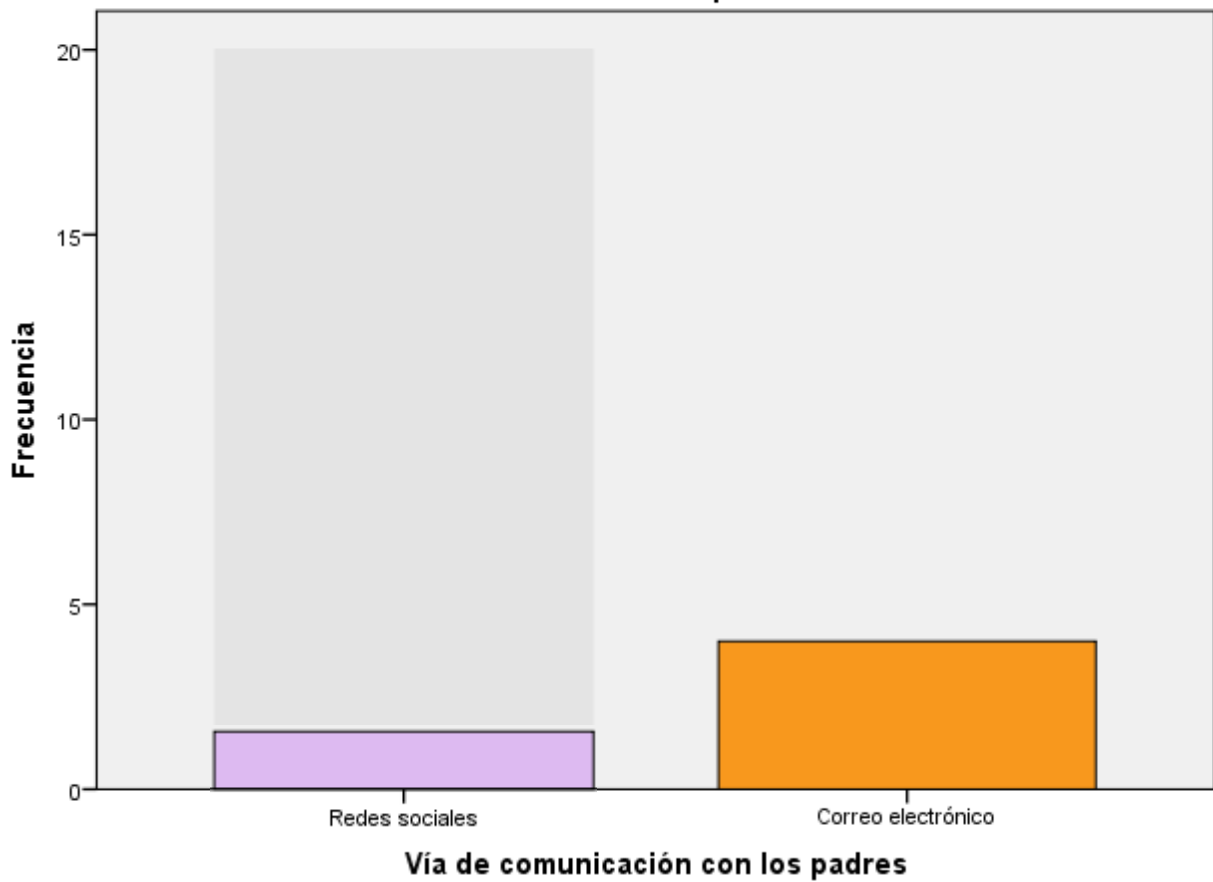


Ilustración 17 Gráfica 17 Vías de comunicación padre-profesor

¿Aparte de las posibles reuniones organizadas por la escuela, tiene otra vía de comunicación con los padres?

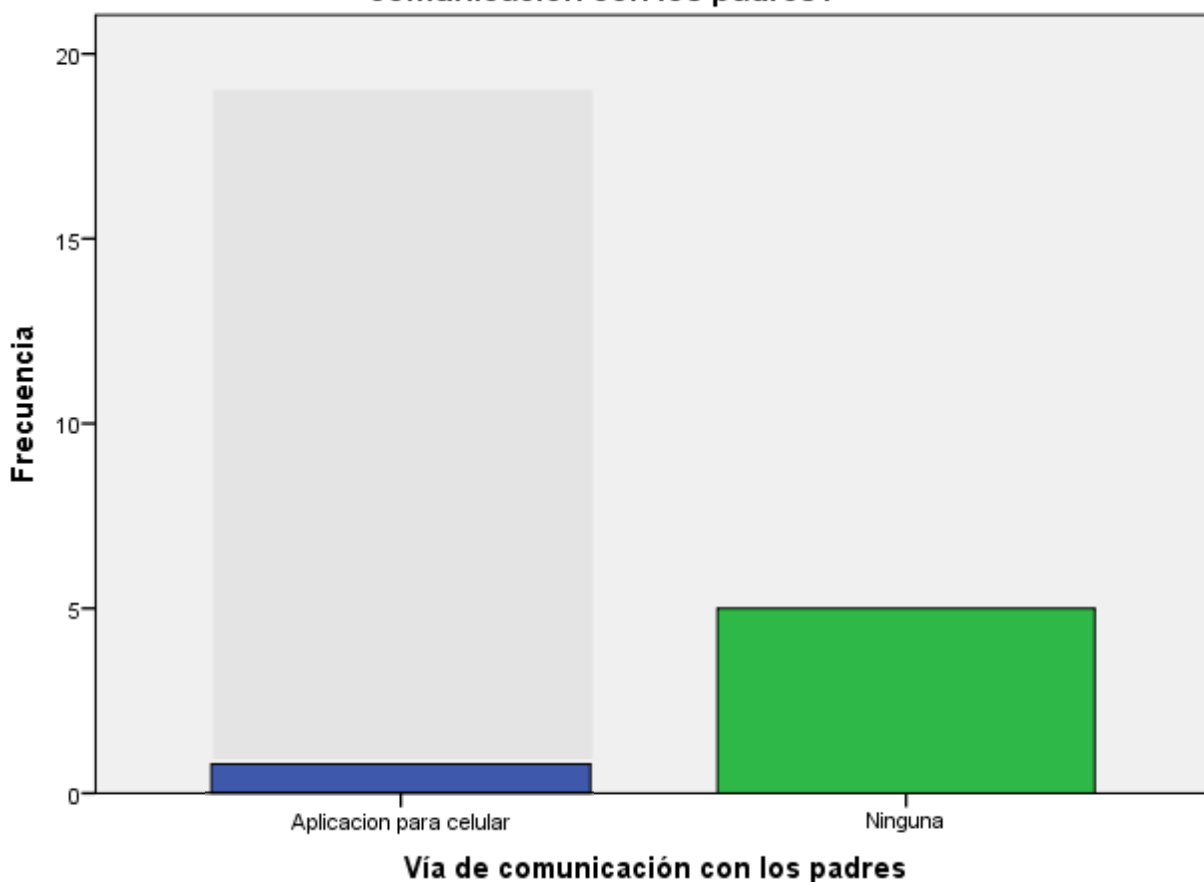


Ilustración 18 Gráfica 18 Vías de comunicación padre-profesor

La mayoría de los profesores indicaron que les gustaría tener otra forma de comunicarse con los padres como se muestra en la siguiente gráfica (gráfica 19), por medio de correo o aplicaciones para celular, esto quiere decir que los profesores no se oponen al uso de la tecnología lo que favorece en gran parte a este trabajo terminal y por lo que nos atrevemos a decir que tendría un gran impacto en las escuelas y sería de gran utilidad para estar en comunicación con los padres de familia. Ofreciendo esto como un servicio de la escuela.

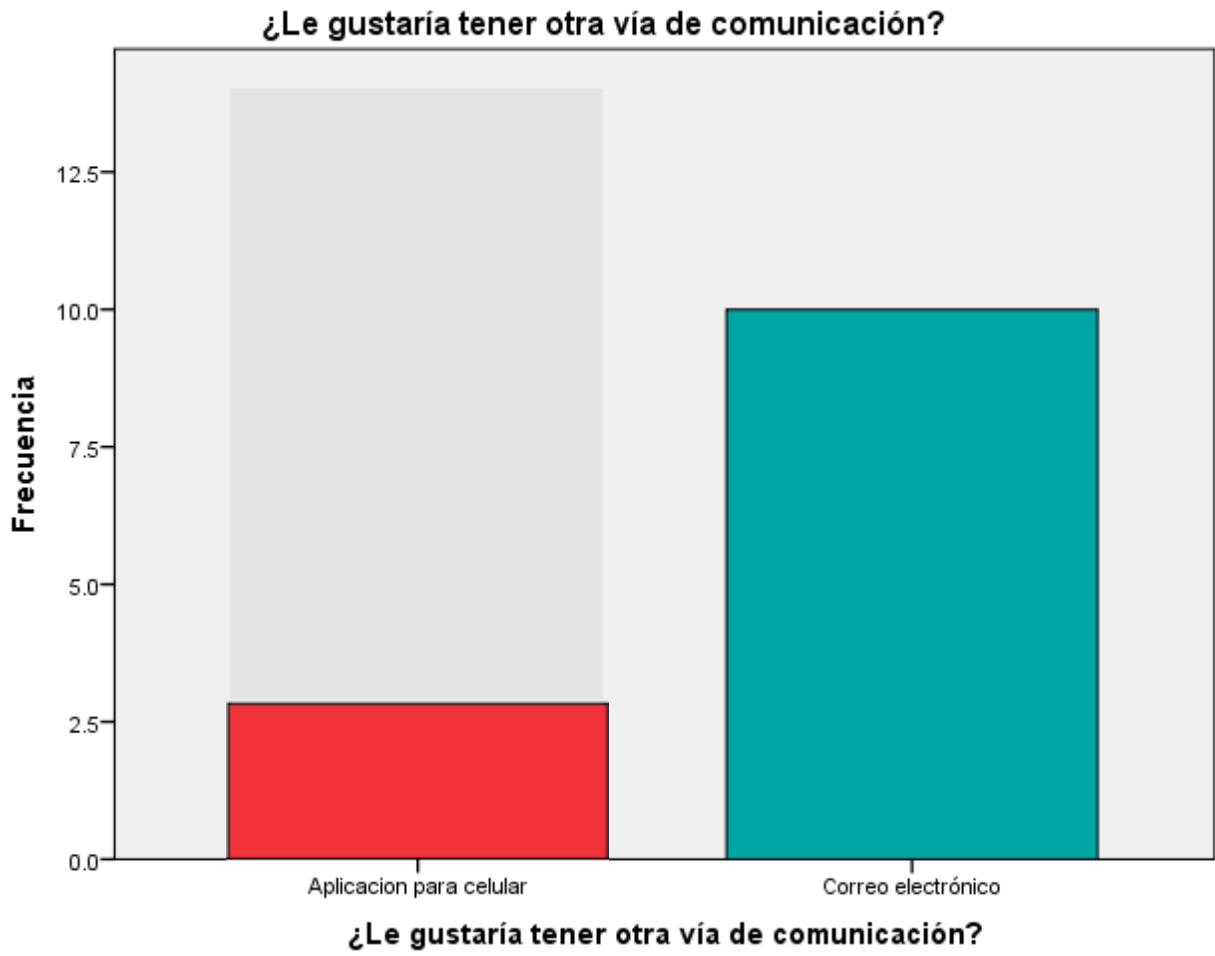


Ilustración 19 Gráfica 19 Vías alternativas de comunicación

Capítulo III. Marco teórico

Educación básica

También conocida como educación general básica, es el proceso que tiene como principal objetivo capacitar al alumno en los principios de su formación académica. En ella se le enseña a leer, escribir, realizar operaciones matemáticas básicas, así como algunos temas culturales. Mediante la impartición de ella se pretende que todos los alumnos partan de una formación en común y a partir de ahí se desarrollen de acuerdo a sus gustos y capacidades.

En México, la educación básica precede a la educación inicial y tiene una duración de 12 años y está dividida en tres niveles:

- Educación preescolar. Se atiende a niños de 3 a 5 años con 11 meses de edad, con el propósito de que vivan experiencias que contribuyan a sus procesos de desarrollo y aprendizaje, y que paulatinamente desarrollen su afectividad, adquieran confianza para expresarse, desarrollen el gusto por la lectura, utilicen el razonamiento matemático, se interesen en la observación de fenómenos naturales, adquieran valores y principios indispensables para la convivencia, usen su imaginación, sean creativos, posean iniciativa para expresarse con lenguajes artísticos, y mejoren sus habilidades de coordinación, desplazamiento [11].
- Educación primaria. Es el segundo nivel de la Educación Básica, donde los niños y niñas aprenden a leer y escribir para comunicarse, desarrollan sus habilidades matemáticas, aprenden a convivir, a explorar el mundo, comprenderlo y desarrollarse como personas. Se atienden niños de 6 a 14 años de edad, a población de 9 a 14 años en situación de riesgo y a jóvenes y adultos de más de 15 años que no han iniciado o concluido su Educación Primaria [12].
- Educación secundaria. Es el tercer nivel de la Educación Básica. Se cursa en tres años en los que se busca que los adolescentes adquieran herramientas para aprender a lo largo de la vida, a través del desarrollo de competencias relacionadas con lo afectivo, lo social, la naturaleza y la vida democrática. Los jóvenes que egresan de Secundaria tienen capacidad de reflexión y análisis, ejercen sus derechos, producen e intercambian conocimientos,

cuidan de la salud y del ambiente. La educación secundaria es básica y obligatoria desde 1993 [13].

Arquitectura Orientada a Servicios (SOA)

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular [14].

Se caracteriza por la utilización de servicios para dar soporte a los requerimientos de software de los usuarios, proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes de la red como servicios independientes a los que tienen acceso de modo estandarizado [15].

Servicio Web

Un servicio web es cualquier componente de software que está disponible a través del protocolo HTTP (internet) y que implementa un lenguaje estandarizado para intercambiar información con otros componentes o aplicaciones para consultar datos, invocar procesos o notificar eventos.

Uno de los pilares de la comunicación a través de servicios web es la implementación del lenguaje estandarizado, el cual usualmente suele ser XML pero que sin embargo está siendo reemplazado por JSON al ser este último más compacto y más entendible para el ser humano. La estandarización permite la interoperabilidad entre componentes montados bajo distintas arquitecturas de hardware o sistemas operativos [16].

Los servicios web son usados para dos tipos de tareas; una de ellas es generar componentes reusables de software, muchos son los ejemplos como servicios que convierten divisas, dan reportes del clima e inclusive ofrecen traducciones entre idiomas. La otra tarea es conectar aplicaciones existentes que de un momento a otro debido a las reglas de negocio deben cooperar para cumplir con los requerimientos.

API

Una API (abreviación de *Application Programming Interface*), es la interfaz a través de la cual un programa de aplicación accede a los recursos de un servicio u otro programa. Dicha API debe estar definida a nivel de código fuente y proveer un nivel de abstracción suficiente para facilitar la portabilidad de las llamadas a ese servicio [17].

Para lograr esto, el programa que requiere comunicarse con otro emplea un conjunto de peticiones estandarizadas y definidas a través de la interfaz. Las APIs son buena idea para el desarrollador que desea exponer las funciones de su aplicación sin perder control sobre su funcionamiento ni perder la propiedad intelectual del mismo, puesto que a través de ellas puede definir el comportamiento del usuario dentro de su sistema.

Existen diversos tipos de API, ya sean para sitios web, sistemas operativos, componentes de hardware, sin embargo, las que pertenecen a servicios web son las más populares, algunos ejemplos son:

- Google Maps
- Twitter
- YouTube
- Facebook
- Foursquare

Aplicación móvil

También conocidas como *apps* móviles o simplemente *apps* son aplicaciones para computadora diseñadas especialmente para funcionar en tablets, phablets, smartphones y otros dispositivos móviles. Por tanto, deben ser diseñadas tomando en consideración las restricciones de dichos dispositivos, tales como la resolución de pantalla, capacidad de almacenamiento o duración de la batería.

Mientras que algunas toman aplicaciones comunes en una PC y portan su funcionalidad a un dispositivo móvil, más comúnmente las *apps* son usadas para conectarse a servicios de internet y ofrecer los recursos que estos proveen tomando las ventajas que estos ofrecen [18].

Las aplicaciones móviles pueden dividirse en tres categorías de acuerdo con su forma de desarrollo, siendo estas las categorías en las que podemos encontrarlas:

- Nativas. Que son aplicaciones escritas por completo en el lenguaje nativo del dispositivo, está pensada para explotar al máximo las características del sistema operativo que la hospeda
- Basadas en web. Funciona de manera similar a una página web, ya que se accede a ella mediante una URL y está escrita en principalmente en HTML, JavaScript y CSS, la experiencia de usuario es igual sin importar la plataforma que este use.
- Híbridas. Similares a las basadas en web, solo que estas si tienen integración con algunas de las *APIs* del dispositivo que las hospeda, inclusive pueden integrar experiencias completamente nativas y totalmente web dentro de la misma app.

Capítulo IV. Análisis de tecnologías de desarrollo

Arquitectura del servicio web

Existen diversas maneras de construir servicios web, sin embargo la primera decisión a tomar radica en qué tipo de arquitectura/plataforma. Para la realización de este trabajo se evaluarán dos tecnologías recientes y muy conocidas: SOAP y REST.

SOAP

SOAP es el acrónimo de *Simple Object Access Protocol*. Y como su nombre lo indica, es un protocolo para el intercambio de mensajes a través de la red empleando el uso del protocolo HTTP. Está basado en XML y es relativamente ligero, lo integran diversas partes. Una “envoltura (*envelope*)” que describe algunos atributos del contenido que envuelve. La segunda es el encabezado, que es empleado como un mecanismo de extensión que indica la manera en que el paquete debe ser procesado. Y por último el cuerpo del mensaje, es donde se contiene toda la información relativa a la llamada y a la respuesta [19].

SOAP fue originalmente pensado para ser una versión de CORBA. Como muestra está su predecesor: XML-RPC. Y es que en verdad funciona para entornos controlados y perfectamente conocido por aquellos que consumirán el servicio y es que los servicios web basados en SOAP emplean un lenguaje que los describe (*WSDL: Web Service Definition Language*), los servicios suelen estar acompañados de un archivo en ese lenguaje que permite definir claramente cualquier detalle de las funciones de nuestro servicio.

REST

REST es el acrónimo de *Representational State Transfer*. Este es una arquitectura para el diseño de sistemas distribuidos en internet, se centra en el uso de protocolo HTTP suprimiendo la necesidad de contar con una capa adicional. Las operaciones se ayudan a través de los verbos (mejor conocidos como métodos HTTP) para ser distinguidas unas con otras ya que a los recursos relacionados se puede acceder a través de una misma URLy es el verbo la que los diferencia.

Adicionalmente, REST también se encarga de gestionar otros aspectos de HTTP, como la falta de estado, el cacheo, Al no basarse más que en el protocolo HTTP resulta ser una forma ligera y sencilla de consumir un servicio.

Este término es utilizado en su mayoría para describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional, como lo hace SOAP, en tal sentido éstos dos significados pueden chocar o incluso solaparse. Es de esta forma que es importante remarcar el hecho de que REST no es un estándar, ya que es tan solo un estilo de arquitectura, pero también está basado en los siguientes estándares:

- HTTP
- URL
- Representación de los recursos: XML/HTML/GIF/JPEG/...
- Tipos MIME: text/xml, text/html,

En la actualidad existen varios proyectos que pueden verse beneficiados de arquitecturas como REST donde la idea principal está en la manera en la que se hacen las peticiones al Servidor desde el cliente, las peticiones se hacen basadas en el recurso que te interesa, por ejemplo si quieres ver los productos de una tienda de la categoría lácteos, debería de ser suficiente con ir a la página <http://latienda/productos/lacteos> y el servidor es el que se encarga de encontrar el recurso adecuado [20].

La arquitectura de REST tiene que cumplir con estos 6 principios.

- **Cliente-servidor** Esta define que deben de estar separados el cliente del servidor a través de interfaces uniformes, es decir el cliente no sabe nada de cómo se almacena la información, ni como se está obteniendo. Por otro lado los servidores no saben la manera en la que se está presentado la información.

- **Capaces de almacenarse en caché** En el WWW los clientes no tienen un mecanismo de almacenar las respuestas en caché. Las respuestas deben de estar implícitas o en su defecto explícitamente deben definirse a sí mismas como almacenables en caché o no, para evitar que los clientes hagan uso inapropiado de información regresada por una solicitud.
- **Sistemas en capas** El cliente no debe de saber si está conectado directamente a un servidor final o a un intermediario. Un servidor intermediario te puede ayudar a balancear las cargas y la escalabilidad de la aplicación.
- **Código bajo demanda** Los servidores pueden ser capaces de extender la funcionalidad de un cliente transfiriéndole lógica que puedan ejecutar, por ejemplo Java Applets o JavaScript.
- **Interface Uniforme** Son recursos individuales que deben de estar incluidos dentro de la solicitud.

Framework y lenguaje de desarrollo

Una vez elegida la arquitectura REST para el desarrollo del sistema, se debe elegir el lenguaje de desarrollo, decisión que estará fuertemente ligada a los frameworks que estén disponibles para cada uno de ellos, a continuación, se presenta una lista con descripciones de cada uno de los considerados:

Recess / PHP

Recess es un framework de desarrollo para PHP que provee una experiencia de desarrollo excelente para principiantes. Recess es rápido, ligero y con poco impacto en el desempeño de la aplicación. Es ideal para el desarrollo en una plataforma tipo LAMP. Está basado en el patrón modelo – vista – controlador y está optimizado y diseñado para funcionar específicamente con PHP 5.

Recess provee además de:

- Un componente de ORM.
- Herramientas de diagnóstico y búsqueda de errores.
- Una interfaz gráfica para acelerar el desarrollo.

Recess fue creado por Kris Jordan, un científico computacional que lo liberó bajo la licencia MIT en GitHub [21].

Play / Java

Otro framework de desarrollo que sigue el patrón modelo – vista – controlador, escrito en Scala y en Java, por lo tanto funciona para ambos lenguajes. Como ventajas resaltan su capacidad de optimizar la productividad del desarrollador al utilizar el paradigma de programación *Convención sobre configuración*, recarga rápida de código y la capacidad de mostrar errores en el navegador.

Es completamente REST, facilita las pruebas con JUnit y Selenium, arquitectura modular, soporte para Scala y Java. Un módulo ORM, un componente simple de SMTP, un framework de testing, entre otras características.

Al momento de la escritura de este documento la última versión estable de *Play* es la 2.3.4 liberada en septiembre de 2014, cuyo código se encuentra en GitHub bajo la licencia Apache 2 [22].

Nancy / C#

Nancy es un framework (ligero en desempeño y en cantidad de código escrito) para construir servicios basados en el protocolo HTTP en las plataformas de .NET y Mono. Tiene como objetivo declarado mantenerse fuera del camino del desarrollador lo más que se pueda para que este pueda trabajar enfocándose solamente en el código de la aplicación.

Está diseñado para manejar peticiones DELETE, GET, HEAD, OPTIONS, POST, PUT y PATCH. Una de las grandes ventajas que presenta es la no dependencia de otros frameworks ya que está construido con *.NET framework client profile* lo cual hace que pueda ser usado en donde sea.

Nancy comenzó a ser desarrollado en 2010 por Andreas Håkansson, y su última versión estable fue liberada bajo la licencia MIT y su código está disponible en GitHub [23].

Tornado / Python

Es un framework para Python para desarrollo web y *networking* asíncrono que provee escalabilidad debido a sus operaciones no bloqueantes de entrada y salida. Además de que facilita la creación de APIs REST de manera rápida.

Tornado es una herramienta de código libre creada por Facebook, cuya versión más reciente es la 4.0.2 y el código se puede encontrar en GitHub [24].

ASP.NET Web API / C#

ASP.NET es un framework gratuito de desarrollo de aplicaciones y servicios web. Entre sus componentes se encuentra Web API, que simplifica la construcción de servicios basados en HTTP, Web API fue introducido en ASP.NET MVC 4, lo cual significa que sigue un patrón modelo – vista – controlador. Provee de todas las características de enrutamiento, controladores y action results. Entre sus características se destacan: manejo de tipos complejos como imágenes, PDFs, tiene soporte para OData y además el ya mencionado soporte para todas las características de MVC.

ASP.NET Web API fue diseñado inicialmente por Microsoft pero ahora es una herramienta de código libre bajo la licencia Apache 2.0 cuyo código está disponible a través de CodePlex [25].

Base de datos

Es innegable que dada la cantidad de datos y la rapidez con la que esperamos que estos sean producidos debemos contar con un fuerte soporte para almacenarlos, al haber elegido C# como lenguaje principal pareciera que es SQL Server la opción ideal, sin embargo también se consideraron otras opciones:

Oracle

Oracle es un sistema gestor de base de datos relacional, es otro de los más usados a nivel mundial dadas sus características entre las que se incluyen su capacidad de ejecutarse en distintos tipos de

sistemas operativos, su lenguaje PL/SQL que permite la implementación de diseños con *triggers* y procedimientos almacenados, además de las características que los otros sistemas ofrecen.

MySQL

Sistema gestor de base de datos relacional, multi-hilo y multiusuario es desarrollada como software libre (bajo la licencia GPL) por Oracle. Es una de las opciones más elegidas por todos los desarrolladores alrededor del mundo. Es muy usado en múltiples aplicaciones web funcionando en línea, pero su popularidad en la web se liga principalmente a PHP [26].

MySQL resulta ideal para entornos de baja concurrencia para actualizar y alta concurrencia para recuperación de datos, en cuanto a la compatibilidad con lenguajes de programación, existen conectores específicos para C# a MySQL [27] una ventaja destacable de MySQL es que además de ser software libre (con un costo de licencia para aplicaciones empresariales) es compatible con una gran cantidad de sistemas operativos.

SQL Server

Este sistema gestor de bases de datos es producido y desarrollado por Microsoft, este provee soporte para transacciones, procedimientos almacenados, modo cliente-servidor. Cuenta con dos lenguajes para trabajar con él T-SQL y ANSI SQL, cuenta con alto rendimiento para aplicaciones de Big Data y BI, soporte para modo local, en la nube o híbrido.

Para el desarrollo de aplicaciones multicapa cuenta con interfaces de acceso a distintas plataformas, siendo C# la más conocida por ser desarrollados por el mismo productor. No se puede dejar de mencionar como desventaja que no es multiplataforma ya que solo funciona bajo sistemas operativos Windows [28].

Proveedor de plataforma web

Al ser *Edutor* una plataforma basada en servicios web debemos considerar también la manera en la que mantendremos los servidores de aplicación y base de datos funcionando con alta disponibilidad. Analizamos diversos proveedores de *hosting* de aplicaciones, incluyendo la posibilidad de almacenar nosotros la aplicación y la base de datos.

Hosting propio

Una opción que podría resultar válida a modo de pruebas, sin embargo nos consumiría tiempo de diseñar e implementar la infraestructura, compra de equipo, realización de contratos con proveedores de internet así como el mantenimiento de los equipos y sistemas. Lo cual derivaría en un incremento de tareas para los miembros del equipo pero otorgaría control total sobre los datos y la aplicación a nivel físico.

Microsoft Azure

Es la plataforma en la nube de Microsoft: una colección de servicios integrados — proceso, almacenamiento, datos, redes y aplicación — que ayudan al desarrollador a crear e implementar soluciones en nube más rápido. Ofrece soluciones de *IaaS* y *PaaS*.

Se precia de ser abierto y flexible al soportar cualquier sistema operativo, lenguaje, herramienta y marco, ya sea Windows, Linux, SQL Server, Oracle, C# o Java. Además, pone a su alcance lo mejor de los ecosistemas de Windows y Linux, por lo que puede crear excelentes aplicaciones y servicios que funcionan con cualquier dispositivo [29].

Amazon Web Services

Amazon Web Services ofrece varios servicios de informática basada en la nube, incluida una amplia selección de instancias informáticas cuyo escalado se puede aumentar y reducir automáticamente para satisfacer las necesidades de su aplicación, un servicio de equilibrio de carga gestionado y escritorios totalmente gestionados en la nube.

Es una plataforma que soporta múltiples tecnologías, entre las que se incluyen Windows, Linux, C#, PHP, Ruby, entre muchos otros. Un punto destacable son sus paquetes a medida que ofrecen cierta cantidad de segundos de cómputo, número de peticiones y ancho de banda [30].

Rackspace

Rackspace es un proveedor de lo que se conoce como Nube Administrada ayuda en el diseño, construcción y operación de cargas de trabajo en la nube liberando así a la empresa de los costos temporales de gestionar la plataforma en nube. Ofrecen dos líneas de negocio: servidores en nube y servidores dedicados [31].

Capítulo V. Análisis y diseño

Diagrama inicial de clases

De un análisis rápido del negocio y de la tecnología y arquitectura elegida se conformó el siguiente diagrama de clases, que sufrirá modificaciones conforme se avance en el análisis y diseño del sistema.

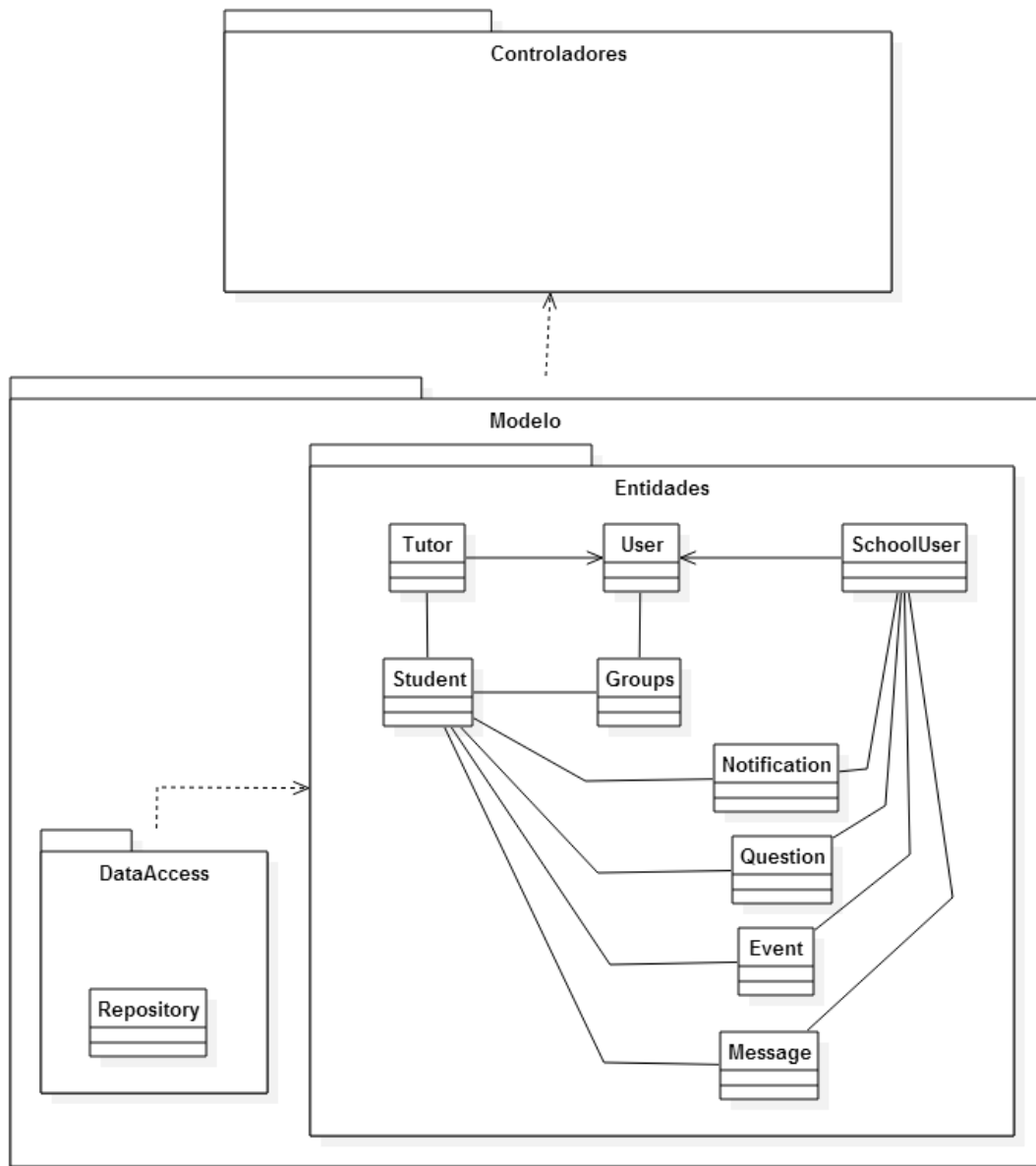


Ilustración 20 Diagrama inicial de clases

Requerimientos funcionales

En la tabla siguiente se puede observar los requerimientos extraídos, con la nomenclatura la columna **I** que representa la importancia del requerimiento y la columna **C** que representa la complejidad de llevarlo a cabo.

ID	Nombre	Descripción	Módulo	I	C
RF1	Gestión de alumnos y grupos	La escuela gestionará a los alumnos y deberá asignarlos a grupos.	General	A	A
RF2	Gestión de profesores	La escuela gestionará a su planta docente y les asignará grupos.	General	A	A
RF3	Autenticación en el sistema	Las credenciales del usuario deben ser válidas para conectarse a la plataforma.	General	A	M
RF4	Privacidad	Garantizar que todos los usuarios tengan acceso únicamente a la información que les corresponde.	General	A	M
RF5	Generación del código de emparejamiento	Para cada tutor la aplicación deberá generar un código único de emparejamiento.	General	A	B
RF6	Emparejamiento	El tutor ingresa el código generado por el profesor.	General	A	B
RF7	Alta de preguntas	El maestro genera una pregunta para un grupo. Las preguntas deben ser de opción múltiple.	Encuestas	A	M
RF8	Asignar duración a pregunta	Cada pregunta debe tener asignada una fecha de expiración.	Encuestas	M	B
RF9	Responder pregunta	El tutor elige la respuesta y la envía, este envío es anónimo.	Encuestas	M	M
RF10	Alta de eventos	El maestro genera un evento dentro de la aplicación.	Eventos	A	M
RF11	Asignar tipo de evento	El maestro puede elegir si el evento será grupal o personal.	Eventos	M	B

RF12	Elegir invitados	El maestro elige invitados: un grupo o un solo tutor.	Eventos	M	B
RF13	Responder a evento	El tutor confirma o rechaza su asistencia al evento.	Eventos	M	M
RF14	Enlace profesor-tutor	El profesor deberá seleccionar un tutor con el que desea entablar una conversación.	Mensajería	A	A
RF15	Enlace tutor-profesor	El tutor deberá seleccionar un profesor con el que desea entablar una conversación.	Mensajería	A	A
RF16	Enlace profesor-grupo	El profesor podrá enviar notificaciones como nuevos temas y tareas.	Notificaciones	A	M
RF17	Marcar notificaciones como vistas	Una vez que el tutor consulte una notificación, esta no debe aparecer como nueva.	Notificaciones	B	B

Tabla 1. Requerimientos funcionales

Se describen los requerimientos de manera detallada.

Nombre	
RF1. Gestión de alumnos y grupos.	
Objetivo	Contar con un área de administración para gestionar alumnos y asignarlos a grupos.
Descripción	Recibir una lista de alumnos y grupos y contar con una forma de vincularlos o desvincularlos.
Entradas	No Aplica.
Salidas	No Aplica.
Precondiciones	El usuario que lo realiza debe contar con las credenciales con privilegios de administrador.
Postcondiciones	Los alumnos (y por tanto sus tutores) quedarán vinculados hasta que explícitamente sean desvinculados por otro usuario.
Errores	Los alumnos y grupos que intente vincular no existen. Un vínculo ya existe entre alumno y grupo.

Tabla 2 Requerimiento Funcional 1

Nombre		RF2. Gestión de profesores.
Objetivo		Contar con un área de administración para profesores.
Descripción		Recibir una lista de grupos y una de profesores para vincularlos o desvincularlos.
Entradas		No Aplica.
Salidas		No Aplica.
Precondiciones		El usuario que lo realiza debe contar con las credenciales con privilegios de administrador.
Postcondiciones		Los profesores quedarán vinculados a uno o varios grupos hasta que sean desvinculados por un usuario administrador.
Errores		Los profesores y grupos que intente vincular no existen. Un vínculo ya existe entre profesor y grupo.

Tabla 3 Requerimiento Funcional 2

Nombre		RF3. Autenticación en el sistema.
Objetivo		Que los usuarios cuenten con las credenciales válidas de acuerdo a su rol para ingresar al sistema.
Descripción		Validar contra los registros en la base las credenciales o el token.
Entradas		Token de seguridad.
Salidas		Mensaje de error o confirmación de credenciales válidas.
Precondiciones		[No aplica]
Postcondiciones		El usuario estará conectado al sistema.
Errores		Credenciales invalidas/ Token incorrecto

Tabla 4 Requerimiento Funcional 3

Nombre		RF4. Privacidad.
Objetivo		Proteger la información de todos los usuarios, especialmente de los alumnos y sus tutores.
Descripción		Verificar que la información presentada a los usuarios sea la correcta de acuerdo a su rol.
Entradas		[No aplica]
Salidas		[No aplica]
Precondiciones		[No aplica]

Postcondiciones	[No aplica]
Errores	[No aplica]

Tabla 5 Requerimiento Funcional 4

Nombre RF5. Generación del código de emparejamiento.	
Objetivo	Proveer de un código que identifique de manera única a un alumno.
Descripción	Genera un código alfanumérico para realizar el emparejamiento que identifique de manera única a un alumno. Dicho código deberá ser otorgado a cada tutor.
Entradas	El identificador de alumno.
Salidas	Un código alfanumérico.
Precondiciones	Que el alumno esté dado de alta.
Postcondiciones	Se genera un código único.
Errores	[No aplica]

Tabla 6 Requerimiento Funcional 5

Nombre RF6. Emparejamiento.	
Objetivo	Realizar el emparejamiento tutor-profesor mediante el código único generado previamente.
Descripción	El tutor ingresa el código único dentro de la aplicación
Entradas	El código único generado por <i>Edutor</i> .
Salidas	Una bandera indicando si el código fue o no válido.
Precondiciones	Que el tutor tenga un código valido.
Postcondiciones	Se le otorgan los privilegios a la aplicación del tutor para consultar la información acerca de su alumno.
Errores	El tutor ingresa un código inválido.

Tabla 7 Requerimiento Funcional 6

Nombre RF7. Alta de preguntas.	
Objetivo	Permitir que el profesor ingrese preguntas para ser contestadas por los tutores de alumnos pertenecientes a sus grupos.
Descripción	El maestro genera una pregunta y la envía para ser contestada anónimamente por los tutores.
Entradas	La pregunta y las opciones posibles para contestarla.
Salidas	La confirmación de si la pregunta fue enviada correctamente o no.
Precondiciones	Que el profesor haya ingresado al sistema.

Postcondiciones	La pregunta queda registrada y es enviada a los tutores para ser respondida
Errores	[No aplica]

Tabla 8 Requerimiento Funcional 7

Nombre	RF8. Asignar duración a pregunta.
Objetivo	Permitir que el profesor asigne una fecha de expiración a una pregunta creada por él.
Descripción	Cada pregunta debe tener asignada una fecha de expiración.
Entradas	La pregunta seleccionada y una fecha de expiración.
Salidas	La confirmación de si la fecha fue asignada o no.
Precondiciones	[No aplica]
Postcondiciones	La pregunta es modificada con la fecha de expiración.
Errores	Que la pregunta no exista.

Tabla 9 Requerimiento Funcional 8

Nombre	RF9. Responder pregunta.
Objetivo	Permitir que el tutor responda a una pregunta enviada por el profesor eligiendo de un conjunto de respuestas predefinidas.
Descripción	El tutor responde la pregunta y esta respuesta es registrada por el sistema.
Entradas	La respuesta a la pregunta.
Salidas	[No aplica]
Precondiciones	Que el tutor tenga preguntas para ser contestadas.
Postcondiciones	La respuesta queda registrada y es añadida a las estadísticas que el profesor puede contestar.
Errores	Que la pregunta ya haya expirado.

Tabla 10 Requerimiento Funcional 9

Nombre	RF10. Alta de eventos.
Objetivo	Permitir que el profesor registre eventos a realizarse con relación a la educación de sus alumnos.
Descripción	El profesor genera un evento dentro del sistema.
Entradas	Descripción del evento, hora, día y requisitos del evento.
Salidas	[No aplica]
Precondiciones	Que el profesor haya ingresado al sistema.

Postcondiciones	Un evento es registrado dentro del sistema y es enviada una notificación a los tutores elegidos por el profesor.
Errores	Que otro evento esté agendado a la misma hora-fecha.

Tabla 11 Requerimiento Funcional 10

Nombre	RF11. Asignar tipo de evento.
Objetivo	Permitir que el profesor elija si el evento creado será individual o grupal.
Descripción	El profesor elige de entre un evento para un solo tutor o para todo un grupo.
Entradas	El evento al que se desea asignar tipo y el tipo de evento.
Salidas	[No aplica]
Precondiciones	Que el profesor haya ingresado al sistema y que haya elegido un evento para editar.
Postcondiciones	El evento seleccionado es modificado.
Errores	[No aplica]

Tabla 12 Requerimiento Funcional 11

Nombre	RF12. Elegir invitados.
Objetivo	En caso de que el evento sea individual, el profesor deberá poder elegir el tutor invitado.
Descripción	El profesor elige de entre la lista de tutores de sus alumnos de los diversos grupos.
Entradas	El evento y el tutor invitado.
Salidas	[No aplica]
Precondiciones	Que el profesor haya ingresado al sistema y que haya elegido un evento al cual se desea elegir invitado.
Postcondiciones	El evento seleccionado es modificado y la invitación es enviada al tutor.
Errores	[No aplica]

Tabla 13 Requerimiento Funcional 12

Nombre	RF13. Responder invitación a evento.
Objetivo	El tutor debe poder confirmar o rechazar su asistencia al evento.
Descripción	El profesor elige de entre la lista de tutores de sus alumnos de los diversos grupos.
Entradas	El evento y la respuesta del tutor.
Salidas	[No aplica]
Precondiciones	[No aplica]

Postcondiciones	El evento seleccionado es modificado y su respuesta queda registrada en el sistema.
Errores	Que el evento al que se desea confirmar no exista.

Tabla 14 Requerimiento Funcional 13

Nombre RF14. Enlace profesor-tutor	
Objetivo	Tener un enlace uno a uno para una mejor comunicación.
Descripción	El profesor pueda elegir el tutor con el que desea establecer una comunicación. No se podrán agregar personas a la conversación, para ello existe otro caso de uso.
Entradas	[No aplica]
Salidas	[No aplica]
Precondiciones	El profesor deberá acceder al grupo en donde se encuentra el tutor con el que desea establecer la comunicación.
Postcondiciones	La comunicación entre profesor-tutor será uno a uno garantizando la privacidad de la misma.
Errores	[No aplica]

Tabla 15 Requerimiento Funcional 14

Nombre RF15. Enlace tutor-profesor	
Objetivo	Tener un enlace uno a uno para una mejor comunicación.
Descripción	El tutor pueda elegir el profesor con el que desea establecer una comunicación, obviando que los profesores deben estrictamente impartirles clases a sus hijos. No se podrán agregar personas a la conversación, para ello existe otro caso de uso.
Entradas	[No aplica]
Salidas	[No aplica]
Precondiciones	El tutor deberá seleccionar el profesor con el que desea establecer una comunicación.
Postcondiciones	La comunicación entre tutor-profesor será uno a uno garantizando la privacidad de la misma.
Errores	[No aplica]

Tabla 16 Requerimiento Funcional 15

Nombre RF16. Enlace profesor-grupo	
Objetivo	Tener un enlace uno a muchos para informar o notificar.
Descripción	El profesor podrá seleccionar el grupo al que desea notificarle o informar algún evento.
Entradas	[No aplica]

Salidas	[No aplica]
Precondiciones	El profesor deberá seleccionar el grupo.
Postcondiciones	La comunicación entre profesor-grupo será uno a muchos garantizando que la información sea visible solo para el grupo seleccionado.
Errores	El grupo no este asignado al profesor.

Tabla 17 Requerimiento Funcional 16

Nombre	
RF17. Marcar notificaciones como vistas	
Objetivo	Tener un control sobre las notificaciones recibidas por parte de los profesores.
Descripción	Una vez que el tutor consulte una notificación, esta no debe aparecer como nueva.
Entradas	[No aplica]
Salidas	[No aplica]
Precondiciones	Haber revisado la notificación.
Postcondiciones	La notificación se descartara como nueva.
Errores	Que el profesor elimine la notificación.

Tabla 18 Requerimiento Funcional 17

Requerimientos no funcionales

Estos son los requisitos que no fueron encontrados como parte de las funciones específicas que debe ofrecer el sistema, si no funciones necesarias para cumplir con las expectativas que se tienen sobre este.

ID	Nombre	Descripción	Módulo	I	C
RNF1	Disponibilidad	La aplicación debe ser accesible mediante internet.	General	A	B
RNF2	Seguridad de los datos	Protección de los datos de todos los usuarios.	General	A	M
RNF3	Inmediatez	Los datos deben ser propagados por la aplicación tan pronto como sean actualizados o creados.	General	M	A
RNF4	Integridad	Los mensajes, notificaciones y cualquier información generada	General	M	M

		debe permanecer sin modificaciones una vez que ha sido creada			
RNF5	Interfaz	En las aplicaciones cliente de <i>Eduator</i> se debe contar con una interfaz gráfica fluida, intuitiva y ergonómica.	Aplicaciones cliente	M	M
RNF6	Interoperabilidad	En las aplicaciones cliente de <i>Eduator</i> se deben contar con mecanismos de interoperabilidad para interactuar con las notificaciones, calendario, etc. dentro del dispositivo.	Aplicaciones cliente	B	B

Tabla 19 Requerimientos no funcionales

Nombre		RNF1. Disponibilidad
Objetivo	Que la aplicación esté disponible todo el tiempo posible, en especial durante los periodos escolares.	
Descripción	La aplicación se mantiene totalmente funcional durante los periodos escolares y más tiempo, de ser posible.	
Entradas	[No aplica]	
Salidas	[No aplica]	
Precondiciones	[No aplica]	
Postcondiciones	[No aplica]	
Errores	[No aplica]	

Tabla 20 Requerimiento no funcional 1

Nombre		RNF2. Seguridad de los datos
Objetivo	Proteger la información almacenada en el sistema de posibles intrusos y personas ajenas a la institución con el fin de garantizar la confidencialidad de los datos proporcionados.	
Descripción	Los datos proporcionados por parte de los tutores hacia la institución deberán manejarse con delicadeza, por ello se debe garantizar que dicha información no estará disponible para el público en general.	
Entradas	[No aplica]	

Salidas	[No aplica]
Precondiciones	Tener almacenados los datos en nuestros servidores de base de datos.
Postcondiciones	Solo la institución y el profesor podrán tener acceso a dicha información.
Errores	[No aplica]

Tabla 21 Requerimiento no funcional 2

Nombre	RNF3. Inmediatez
Objetivo	Que la información esté disponible para los receptores tan pronto como alguien la cree.
Descripción	Los datos deben ser propagados por la aplicación tan pronto como sean actualizados o creados.
Entradas	Cualquier información creada dentro del sistema, por el profesor o el tutor.
Salidas	[No aplica]
Precondiciones	[No aplica]
Postcondiciones	[No aplica]
Errores	[No aplica]

Tabla 22 Requerimiento no funcional 3

Nombre	RNF4. Integridad
Objetivo	Que la información sea entregada a sus receptores tal y como el emisor la envió, sin modificaciones de tamaño o contenido.
Descripción	Los mensajes, notificaciones y cualquier información generada deben permanecer sin modificaciones una vez que ha sido creada.
Entradas	Cualquier información creada dentro del sistema por el profesor o tutor.
Salidas	[No aplica]
Precondiciones	[No aplica]
Postcondiciones	[No aplica]
Errores	[No aplica]

Tabla 23 Requerimiento no funcional 4

Nombre	RNF5. Interfaz
Objetivo	Que las aplicaciones cliente que hagan uso de <i>Edutor</i> otorguen una facilidad de uso a los usuarios.
Descripción	En las aplicaciones cliente de <i>Edutor</i> se debe contar con una interfaz gráfica fluida, intuitiva y ergonómica.

Entradas	[No aplica]
Salidas	[No aplica]
Precondiciones	[No aplica]
Postcondiciones	La aplicación presenta una experiencia de usuario suficientemente buena como para agradar a quien la usa.
Errores	[No aplica]

Tabla 24 Requerimiento no funcional 5

Nombre	RNF6. Interoperabilidad
Objetivo	Que las aplicaciones cliente que hagan uso de <i>Edutor</i> otorguen conectividad con los recursos del dispositivo en el que están montadas.
Descripción	En las aplicaciones cliente de <i>Edutor</i> se deben contar con mecanismos de interoperabilidad para interactuar con las notificaciones, calendario, etc. dentro del dispositivo.
Entradas	[No aplica]
Salidas	[No aplica]
Precondiciones	[No aplica]
Postcondiciones	[No aplica]
Errores	[No aplica]

Tabla 25 Requerimiento no funcional 6

Métricas

Puntos de función

Estimación inicial

Entradas

1. Alta de alumnos
2. Alta de profesores
3. Alta de grupos
4. Alta de vinculo grupo-profesor
5. Alta de vinculo grupo-tutor
6. Credenciales de autenticación
7. Token de emparejamiento
8. Alta de encuestas
9. Alta de eventos
10. Alta de notificaciones
11. Alta de mensajes en una conversación
12. Respuesta a evento
13. Respuesta a encuesta
14. Aviso de mensaje visto

Salidas

1. Mensaje de entrega correcta de información consultada
2. Mensaje de operación completada correctamente
3. Mensaje de recurso no encontrado
4. Mensaje de error interno
5. Mensaje de error en la autenticación
6. Mensaje de error en la validación de los datos ingresados
7. Mensaje de notificación push

Peticiones

1. Consulta alumnos
2. Consulta profesores
3. Consulta tutores
4. Consulta grupos
5. Consulta mensajes
6. Consulta notificaciones
7. Consulta eventos
8. Consulta encuestas

Archivos

1. Base de datos

Interfaces externas

2. Servicios de notificaciones push

De la información anterior se obtiene esta tabla siguiente:



Descripción	Baja		Media		Alta		Total
	Valor	Cantidad	Valor	Cantidad	Valor	Cantidad	
Entradas	3	0	4	0	6	14	84
Salidas	4	0	5	0	7	7	49
Peticiones	3	0	4	8	6	0	24
Archivos	7	0	10	1	15	0	10
Interfaces	5	1	7	0	10	0	5
UPF Totales							172

Tabla 26 Puntos de función en el sistema

Factor de ajuste

La estimación anterior debe ajustarse valorando la complejidad del proceso, que varía en función de las características de la aplicación y de las herramientas requeridas para realizarla.

Se deben tomar en cuenta catorce características, para las cuales se ayuda de preguntas que deben ser respondidas en una escala definida de grados de influencia:

Valor	Descripción
0	Sin influencia
1	Incidental
2	Moderado
3	Medio
4	Significativo
5	Escencial

Tabla 27 Descripción del factor de ajuste

Pregunta	Valor
¿Requiere el sistema de copias de seguridad y recuperación fiables?	5
¿Requiere comunicación de datos?	3
¿Existen funciones de procesamiento distribuido?	1
¿Es crítico el rendimiento?	3
¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	3
¿Requiere el sistema de entrada de datos interactiva?	5

¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	3
¿Se actualizan los archivos maestros de forma interactiva?	1
¿Son complejas las entradas, las salidas, los archivos o las peticiones?	3
¿Es complejo el procesamiento interno?	3
¿Se ha diseñado el código para ser reutilizable?	4
¿Es fácil la instalación?	5
¿Soporta múltiples instalaciones?	2
¿Tiene facilidad de cambios y amigabilidad?	5
Total de grados de influencia	43

Tabla 28 Cálculo del factor de ajuste

Para el cálculo del factor de ajuste (FA), se emplea la siguiente fórmula:

$$FA = 0.65 + 0.01 * \sum_{i=1}^{14} F_i$$

Empleando los datos obtenidos de las preguntas:

$$FA = 0.65 + 0.01 * (43)$$

$$FA = 1.08$$

Puntos de función totales

$$PF = UPF * FA$$

$$PF = 172 * 1.08$$

$$PF = 185.76$$

Reconciliación de métricas

Para poder hacer uso de otras técnicas de métricas, se realizó la reconciliación de métricas, para convertir los Puntos de Función en miles de líneas de código (KLDC). Partiendo del hecho de que la mayoría del sistema estará escrito en C# (lenguaje orientado a objetos), nos queda la siguiente fórmula:

$$1 PF = 54 LDC [32]$$

$$LDC_{totales} = 185.76 * 54 = 10031.04$$

10KLDC

Estimación empleando COCOMO Básico

Emplearemos el nivel básico, que nos otorga una aproximación rápida aunque no tan certera del proyecto, así como el modo orgánico, ya que nuestros requisitos no son tan rígidos y que nuestro equipo es pequeño.

Esfuerzo:

$$E = 2.4 * (10^{1.05}) \approx 26$$

Tiempo:

$$D = 2.5 * (26^{0.38}) \approx 8.6$$

Personas:

$$N = \frac{26}{8.6} \approx 3$$

Costo de desarrollo

Para el cálculo del costo de desarrollo del sistema, se tomaron en cuenta los salarios de un desarrollador C# al mes, el cual para 2014 es de \$24,000 [33] en México.

Salario mensual desarrollador C#: $S = 24000$

Desarrolladores: $D = 3$

Meses desarrollo $M = 9$

$$C = S * D * M$$

$$C = 24000 * 3 * 9$$

$$C = 648000$$

Lo que nos deja con un costo total de desarrollo de \$648,000

Análisis de riesgos

Se usa el proceso de análisis de riesgos para valorar los riesgos de un proyecto, en él se consideran las probabilidades de que ocurra un suceso inusual que afecte el desarrollo del proyecto, desde su planteamiento hasta la implementación exitosa del mismo.

Adjunto a la identificación de riesgos y la probabilidad de que estos se conviertan en realidades que nos afecten, se proponen estrategias para mitigar los efectos negativos de estos.

Podemos clasificar los riesgos en: Escolares, Personales, Diseño, Análisis, Tecnológicos

Clasificación	Riesgo
Escolares	Cambio en las fechas de presentación Cambio en los requisitos para la presentación
Personales	Enfermedad de algún integrante Salida de algún integrante del equipo Dificultad en encontrar tiempos en común para trabajar conjuntamente Falta de conocimiento sobre algún tema por parte de algún integrante del equipo
Análisis	Que no se obtenga la información suficiente por parte de los usuarios Que se encuentren requerimientos ocultos en una fase posterior al análisis Que alguna normativa cambie y afecte el análisis realizado
Tecnológicos	Que no se consigan los apoyos necesarios para llevar a cabo el proyecto Que las escuelas no cuenten con la tecnología necesaria para implantar el sistema.

Tabla 29 Riesgos

Partiremos de la siguiente clasificación para definir las probabilidades de que los riesgos ocurran:

Rangos	Expresión
0 a 10 %	Baja
11 a 25 %	Poco probable
26 a 55 %	Media
56 a 80 %	Altamente probable
81 a 100 %	Casi seguro

Tabla 30 Rangos de probabilidades

Descripción	Expresión
Es mínimo y se puede atender al finalizar las fases críticas del trabajo	Despreciable
Es mínimo pero debe atenderse tan pronto como sea posible	Tolerable
Es considerable y debe atenderse de inmediato para evitar retrasos en las entregas	Moderado
Pone en riesgo las fechas de entrega y los productos finales	Serio
El efecto es grande y pone en riesgo la conclusión del trabajo terminal	Grave

Tabla 31 Descripción de los efectos de los riesgos

Riesgo	Probabilidad	Efecto
Cambio en las fechas de presentación	Baja	Serio
Cambio en los requisitos de presentación	Poco probable	Serio
Enfermedad de algún integrante	Media	Serio
Salida de algún integrante del equipo	Baja	Grave

Dificultad en encontrar tiempos en común para trabajar conjuntamente	Altamente probable	Tolerable
Falta de conocimiento sobre algún tema por parte de algún integrante del equipo	Media	Serio
Que no se obtenga la información suficiente por parte de los usuarios	Media	Moderado
Que se encuentren requerimientos ocultos en una fase posterior al análisis	Poco probable	Moderado
Que alguna normativa cambie y afecte el análisis realizado	Baja	Tolerable
Que no se consigan los apoyos necesarios para llevar a cabo el proyecto	Poco probable	Moderado
Que las escuelas no cuenten con la tecnología necesaria para implantar el sistema	Altamente probable	Moderado

Tabla 32 Riesgos, probabilidades y efectos

Una vez realizado el análisis de probabilidades de que ocurran y el efecto que estos representan, se elaboraron las siguientes estrategias para mitigar los efectos:

Riesgo	Estrategia
Cambio en las fechas de presentación	Realizar la recalendarización de las actividades y las modificaciones necesarias en la documentación

	generada. Valorar las horas necesarias para completar el proyecto.
Cambio en los requisitos de presentación	Realizar los ajustes a las presentaciones, y obtener la documentación necesaria en caso de que así se requiera.
Enfermedad de algún integrante	Organizar el trabajo de forma que la interacción física del equipo sea la mínima, facilitar el teletrabajo.
Salida de algún integrante del equipo	Realizar la recalendarización de las actividades y las modificaciones necesarias en la documentación generada. Valorar las horas necesarias para completar el proyecto.
Dificultad en encontrar tiempos en común para trabajar conjuntamente	Organizar el trabajo de forma que la interacción física del equipo sea la mínima, facilitar el teletrabajo. Fijar un horario para realizar sesiones presenciales.
Falta de conocimiento sobre algún tema por parte de algún integrante del equipo	Encontrar cursos, libros, tutoriales que ayuden al integrante a superar la carencia de conocimiento. Buscar aprender lo estrictamente necesario para el cumplimiento de las metas del proyecto.
Que no se obtenga la información suficiente por parte de los usuarios	Tratar de intuir la información faltante, documentando de manera suficiente su falta para que en un posterior momento se pueda rectificar.
Que se encuentren requerimientos ocultos en una fase posterior al análisis	Realizar las adecuaciones necesarias procurando no afectar significativamente el trabajo previamente realizado.
Que alguna normativa cambie y afecte el análisis realizado	Realizar las adecuaciones necesarias procurando no afectar significativamente el trabajo previamente realizado.
Que no se consigan los apoyos necesarios para llevar a cabo el proyecto	Buscar herramientas que suplan a las no conseguidas, favoreciendo los costos bajos sin sacrificar la calidad del trabajo.

Que las escuelas no cuenten con la tecnología necesaria para implantar el sistema	Es necesario que los usuarios tengan conocimiento previo de los requisitos. Para llevar a cabo las pruebas e implementación se emplearan recursos del equipo que presenta el TT.
--	--

Tabla 33 Estrategias para la gestión de riesgos

Arquitectura del proyecto

En la ilustración se muestra a grandes rasgos la arquitectura ideal del sistema.

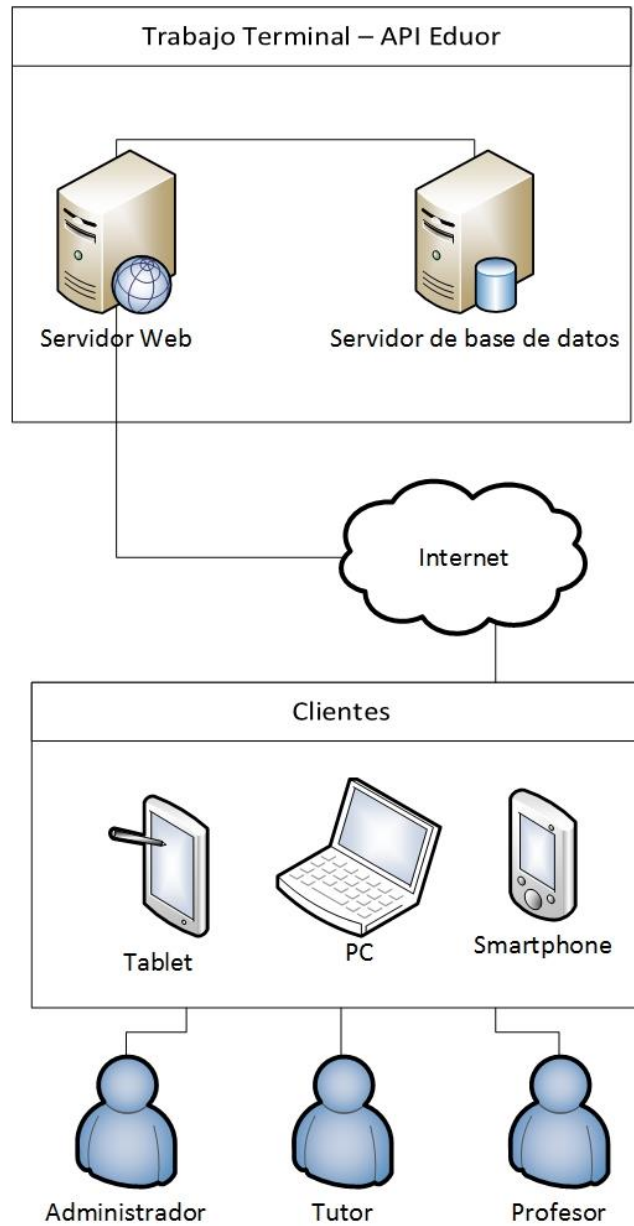


Ilustración 21 Diagrama de la arquitectura del proyecto

Usuarios

Se describen todos los usuarios de *Edutor*

- **Tutor.** Son todos aquellos encargados fuera de la escuela de los alumnos, con ellos se realizará la comunicación. Su acceso será controlado a través de un código único generado dentro de *Edutor*.
- **Profesor.** Son los encargados de impartir la educación dentro de la escuela, para acceder a la aplicación necesitarán un nombre de usuario y contraseña fijos, determinados por la escuela.
- **Administrador.** Cumplen la función de dar de alta estudiantes y profesores, genera grupos y los asigna a los profesores. Su acceso es controlado mediante un nombre de usuario y contraseña.

Dispositivos

El *API* de *Edutor* está diseñada para ser consumida por clientes móviles, sin embargo en el diagrama se pueden observar dos tipos de dispositivos:

- **Cliente móvil.** Es el dispositivo recomendado para tutores y profesores, se accederá a *Edutor* a través de una aplicación móvil.
- **Cliente de escritorio.** Dispositivo recomendado para llevar a cabo la administración de una escuela, el acceso será a través de un sitio web.

Casos de uso

General

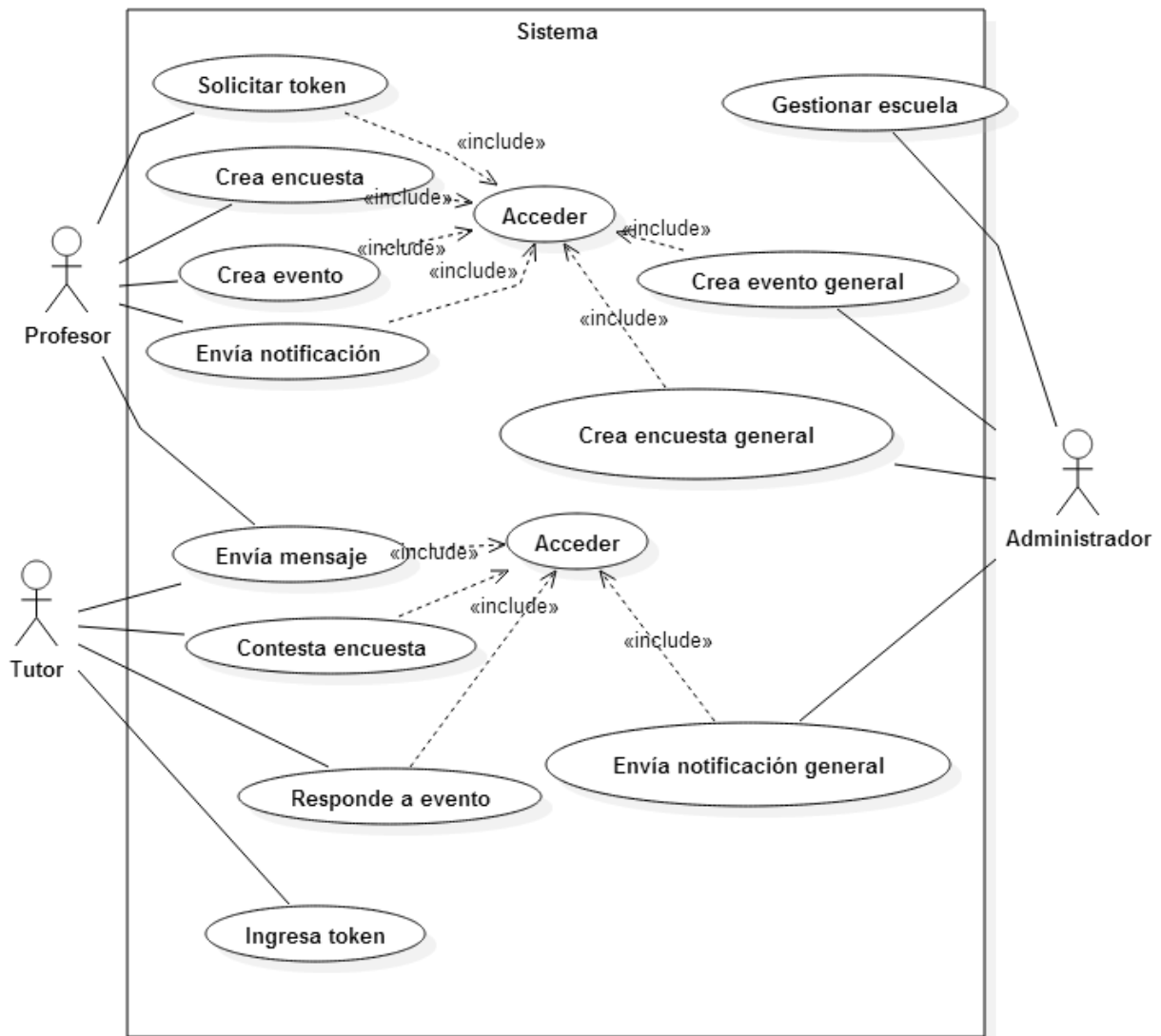


Ilustración 22 Diagrama del caso de uso general

En el diagrama general se aprecia a grandes rasgos todos los actores y las interacciones que estos tienen con el sistema. Cada conjunto de casos de uso pertenecen a un módulo, que se detalla en los diagramas que siguen a este.

Actores

- **Administrador:** Cuando se hace referencia a este actor, en realidad se hace referencia a uno o varios administradores de la escuela. Este usuario tiene el privilegio de administrar grupos, alumnos y profesores.

- **Profesor:** Este usuario es uno de los que más interacciones tiene dentro del sistema,
- **Tutor:** Este usuario es quien tendrá la interacción directa con el actor profesor.

Casos de uso acceso

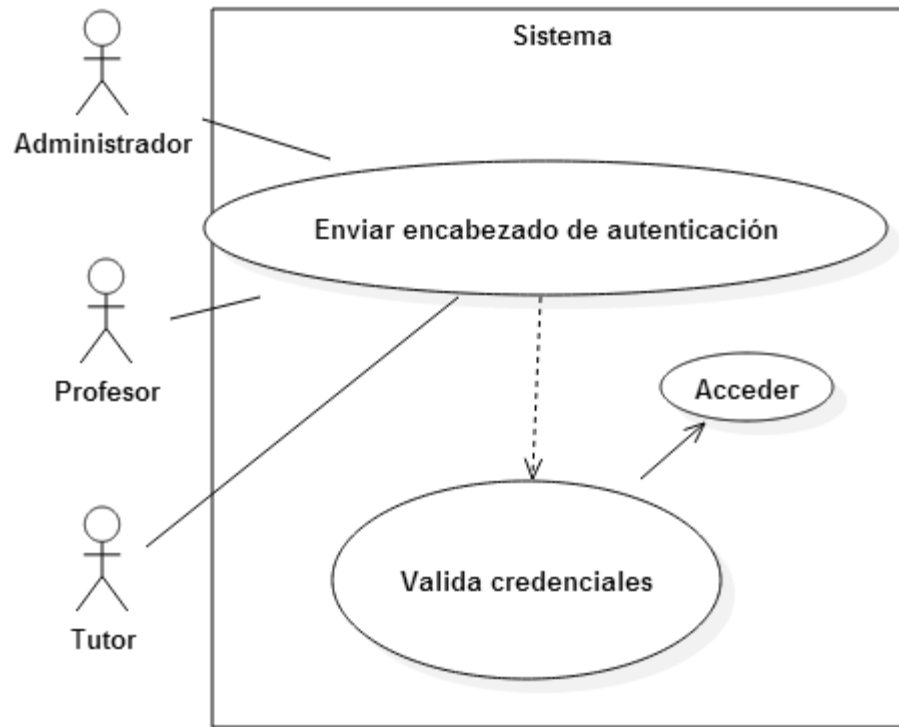


Ilustración 23 Diagrama de caso de uso acceso

Nombre	Acceso
Actores	Administrador / Profesor / Tutor
Descripción	Los clientes que hagan uso de <i>Eduator</i> deberán enviar en el encabezado de la petición HTTP ciertos tokens de autorización para poder usar las acciones que le fueron asignadas dentro del sistema.
Precondiciones	El usuario debe existir en la base de datos, es decir, debe estar registrado.
Postcondiciones	El usuario podrá interactuar con el sistema.
Flujo normal	El usuario ingresa los datos que le fueron otorgados El sistema valida los datos

	En caso de ser correctos, se otorgará acceso al sistema y a las operaciones que desee realizar.
Flujos alternativos	El usuario ingresa los datos que le fueron otorgados El sistema valida los datos El sistema responde con un mensaje de error dado que los datos son inválidos
Excepciones	Los datos ingresados no son correctos
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 34 Caso de uso acceso

Casos de uso gestión escolar

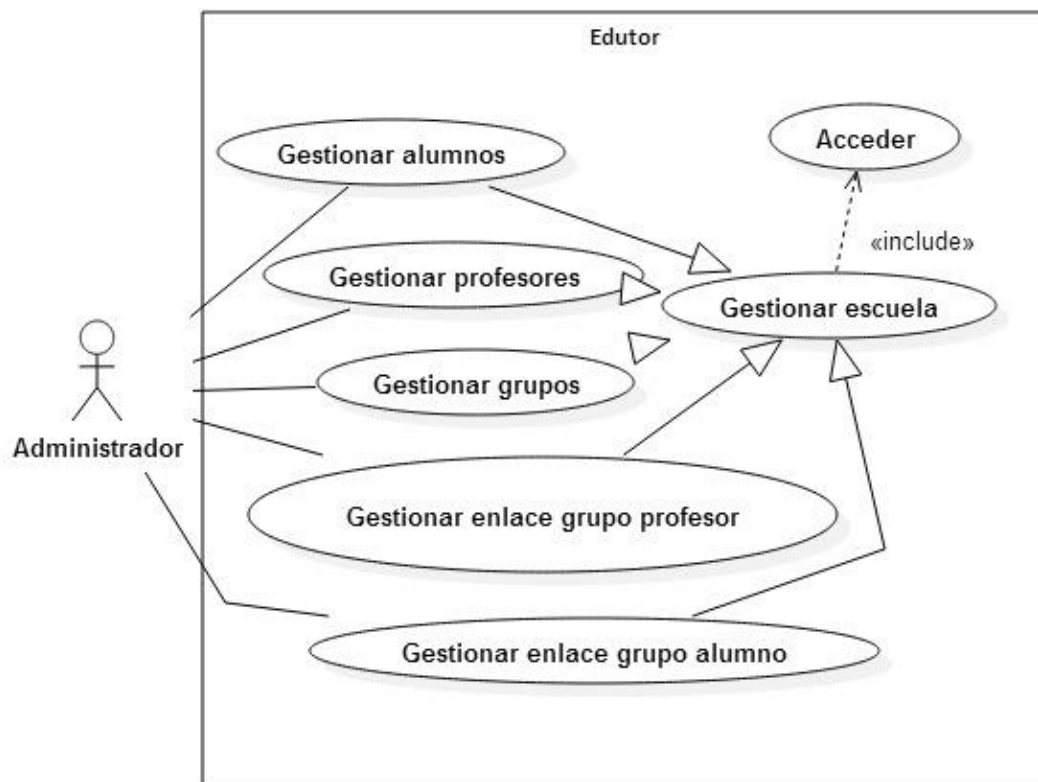


Ilustración 24 Diagrama de caso de uso gestión escolar

Nombre **Gestionar alumnos**

Actores	Administrador
Descripción	Permite al administrador añadir, modificar y eliminar alumnos
Precondiciones	El usuario tiene autorización para gestionar los alumnos
Postcondiciones	Los datos dentro del sistema se modifican de acuerdo a las peticiones del usuario
Flujo normal	<p>Existen varios flujos a seguir:</p> <p>Añadir un alumno:</p> <p>Se ingresan los datos del alumno</p> <p>Se envían al sistema</p> <p>Se validan los datos</p> <p>Si los datos válidos y no existe en el sistema, se da de alta</p> <p>Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido</p> <p>Modificar un alumno</p> <p>Se indica al sistema el alumno que se desea modificar</p> <p>El sistema regresa los datos modificables</p> <p>El usuario modifica los datos</p> <p>Se envían los datos al sistema</p> <p>Se validan los datos</p> <p>Si los datos son válidos, las modificaciones se llevan a cabo</p> <p>Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido</p> <p>Eliminar un alumno</p> <p>Se indica al sistema el alumno que se desea eliminar</p> <p>El sistema valida la posibilidad de eliminar al alumno</p> <p>Si es válido eliminar un alumno, se elimina</p> <p>Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido</p>
Flujos alternativos	[No aplica]
Excepciones	Que los datos no sean válidos, que un alumno esté ligado a un grupo y se intente eliminar.

Prioridad	Alta
Frecuencia de uso	Media

Tabla 35 Caso de uso Gestionar alumnos

Nombre	Gestionar profesores
Actores	Administrador
Descripción	Permite al administrador añadir, modificar y eliminar profesores
Precondiciones	El usuario tiene autorización para gestionar los profesores
Postcondiciones	Los datos dentro del sistema se modifican de acuerdo a las peticiones del usuario
Flujo normal	<p>Existen varios flujos a seguir:</p> <ul style="list-style-type: none"> • Añadir un profesor: <ul style="list-style-type: none"> ○ Se ingresan los datos del profesor ○ Se envían al sistema ○ Se validan los datos ○ Si los datos válidos y no existe en el sistema, se da de alta ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido • Modificar un profesor <ul style="list-style-type: none"> ○ Se indica al sistema el profesor que se desea modificar ○ El sistema regresa los datos modificables ○ El usuario modifica los datos ○ Se envían los datos al sistema ○ Se validan los datos ○ Si los datos son válidos, las modificaciones se llevan a cabo ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido • Eliminar un profesor <ul style="list-style-type: none"> ○ Se indica al sistema el profesor que se desea eliminar

	<ul style="list-style-type: none"> ○ El sistema valida la posibilidad de eliminar al profesor ○ Si es válido eliminar al profesor, se elimina ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido
Flujos alternativos	[No aplica]
Excepciones	Que los datos no sean válidos, que un profesor esté ligado a un grupo y se intente eliminar.
Prioridad	Alta
Frecuencia de uso	Media

Tabla 36 Caso de uso Gestionar profesores

Nombre	Gestionar grupos
Actores	Administrador
Descripción	Permite al administrador añadir, modificar y eliminar grupos
Precondiciones	El usuario inició sesión con su cuenta
Postcondiciones	Los datos dentro del sistema se modifican de acuerdo a las peticiones del usuario
Flujo normal	<p>Existen varios flujos a seguir:</p> <ul style="list-style-type: none"> ○ Añadir un grupo: <ul style="list-style-type: none"> ○ Se ingresan los datos del grupo ○ Se envían al sistema ○ Se validan los datos ○ Si los datos válidos y no existe en el sistema, se da de alta ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido ○ Modificar un grupo <ul style="list-style-type: none"> ○ Se indica al sistema el grupo que se desea modificar ○ El sistema regresa los datos modificables ○ El usuario modifica los datos

	<ul style="list-style-type: none"> ○ Se envían los datos al sistema ○ Se validan los datos ○ Si los datos son válidos, las modificaciones se llevan a cabo ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido ○ Eliminar un grupo <ul style="list-style-type: none"> ○ Se indica al sistema el grupo que se desea eliminar ○ El sistema valida la posibilidad de eliminar el grupo ○ Si es válido eliminar el grupo, se elimina ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido
Flujos alternativos	[No aplica]
Excepciones	Que los datos no sean válidos
Prioridad	Alta
Frecuencia de uso	Media

Tabla 37 Caso de uso Gestionar grupos

Nombre	Gestionar enlaces profesor-grupo
Actores	Administrador
Descripción	Permite al administrador agregar y eliminar relaciones entre un grupo y un profesor
Precondiciones	Que el usuario esté autenticado, tenga autorización para manipular los datos y que tanto el grupo y el profesor que desea manipular existan
Postcondiciones	Los datos dentro del sistema se modifican de acuerdo a las peticiones del usuario
Flujo normal	Existen varios flujos a seguir: <ul style="list-style-type: none"> ● Vincular: <ul style="list-style-type: none"> ○ Se elige el ID del grupo ○ Se elige el ID del profesor ○ Se envían los datos al sistema ○ Se validan los datos ○ Si los datos son válidos, las modificaciones se llevan a cabo ○ Se envía un mensaje de respuesta al usuario dependiendo de lo sucedido

	<ul style="list-style-type: none"> • Desvincular <ul style="list-style-type: none"> ○ Se elige el ID del grupo ○ Se elige el ID del profesor ○ Se envían los datos al sistema ○ Se validan los datos ○ Si los datos son válidos, las modificaciones se llevan a cabo ○ Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Media

Tabla 38 Caso de uso Gestionar enlaces profesor-grupo

Nombre	
Actores	Escuela
Descripción	Permite al administrador agregar y eliminar relaciones entre un grupo y un profesor
Precondiciones	Que el usuario esté autenticado y que tanto el grupo y el alumno que desea manipular existan
Postcondiciones	Los datos dentro del sistema se modifican de acuerdo a las peticiones del usuario
Flujo normal	<p>Existen varios flujos a seguir:</p> <ul style="list-style-type: none"> • Vincular: <ul style="list-style-type: none"> ○ Se elige el ID del grupo ○ Se elige el ID del alumno ○ Se envían los datos al sistema ○ Se validan los datos ○ Si los datos son válidos, las modificaciones se llevan a cabo ○ Se envía un mensaje de respuesta al usuario dependiendo de lo sucedido • Desvincular <ul style="list-style-type: none"> ○ Se elige el ID del grupo ○ Se elige el ID del alumno ○ Se envían los datos al sistema ○ Se validan los datos ○ Si los datos son válidos, las modificaciones se llevan a cabo <p>Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido</p>
Flujos alternativos	[No aplica]
Excepciones	[No aplica]

Prioridad	Alta
Frecuencia de uso	Media

Tabla 39 Caso de uso Gestionar enlaces alumno-grupo

Casos de uso módulo de encuestas

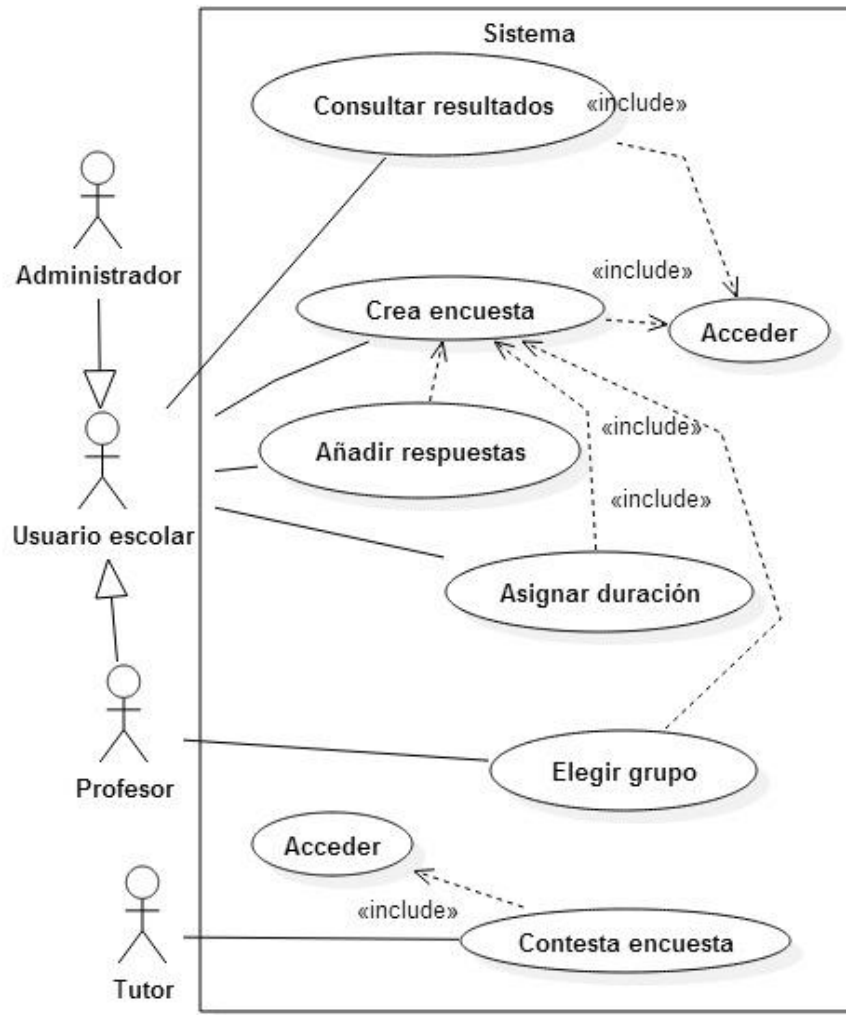


Ilustración 25 Diagrama de los casos de uso Encuestas

Nombre	Genera encuesta
Actores	Usuario escolar
Descripción	El sistema permitirá crear encuestas con respuestas predefinidas para conocer la opinión de los padres
Precondiciones	El usuario inició sesión con su cuenta
Postcondiciones	Una encuesta es enviada a los usuarios seleccionados

Flujo normal	El usuario elige crear una nueva encuesta Ingresa la pregunta a realizar Se envía al sistema Se validan los datos de la pregunta y en caso de ser válidos, se da de alta. Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido
Flujos alternativos	[No aplica]
Excepciones	El usuario ingresó datos incorrectos sobre su pregunta
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 40 Caso de uso Genera encuesta

Nombre Asignar duración a encuesta	
Actores	Usuario escolar
Descripción	Todas las preguntas deben tener asignada una fecha de duración, o se le colocará una por default.
Precondiciones	El usuario inició sesión en el sistema y tiene una pregunta para asignar duración.
Postcondiciones	La duración de una pregunta queda asignada con carácter de inmodificable.
Flujo normal	El usuario ha añadido una pregunta. Se elige si desea asignar una duración o dejar la predefinida por el sistema. Se introducen los datos y se envían al sistema. El sistema valida los datos. Si son válidos modifica la duración de la pregunta. Se envía un mensaje de respuesta de acuerdo a lo sucedido.
Flujos alternativos	El usuario decide no modificar la duración de la encuesta. La encuesta queda con duración por default.
Excepciones	Los datos enviados no son válidos como fecha.
Prioridad	Media
Frecuencia de uso	Alta

Tabla 41 Caso de uso Asignar duración a encuesta

Nombre Añadir respuestas	
--	--

Actores	Usuario escolar
Descripción	Las preguntas deben tener un conjunto de respuestas asignadas, a manera de opciones.
Precondiciones	El usuario inició sesión en el sistema y tiene una pregunta recién cread sin posibles respuestas asignadas.
Postcondiciones	Se le asignan las posibles respuestas a una pregunta.
Flujo normal	El usuario ha añadido una pregunta. Se introducen las posibles opciones de respuesta, una por una. El sistema valida cada respuesta posible. Si son válidas, se asigna la posible respuesta a la pregunta. Se envía un mensaje de acuerdo a lo sucedido.
Flujos alternativos	[No aplica]
Excepciones	La cantidad máxima de respuestas ha sido excedida. Las posibles respuestas no son válidas.
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 42 Caso de uso Añadir respuestas

Nombre Elegir grupo	
Actores	Profesor
Descripción	Las preguntas deben tener un grupo que será quién deba responder la pregunta
Precondiciones	El usuario inició sesión en el sistema y tiene una pregunta recién creada sin un grupo asignado.
Postcondiciones	La pregunta queda asignada a un grupo.
Flujo normal	El usuario ha añadido una pregunta. Se elige uno de los grupos que el profesor tiene asignados. Se envía un mensaje de acuerdo a lo sucedido.
Flujos alternativos	[No aplica]
Excepciones	El profesor no tiene asignado el grupo elegido.
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 43 Caso de uso Elegir grupo

Nombre Consultar resultados	
Actores	Usuario escolar
Descripción	La finalidad de generar una encuesta radica en observar los resultados de esta.
Precondiciones	El usuario inició sesión en el sistema y tiene una pregunta creada por el de la cual desea conocer el resultado.
Postcondiciones	El profesor tiene conocimiento de las respuestas que los tutores han contestado a una pregunta.

Flujo normal	El usuario consulta sus preguntas válidas Elige la que desea revisar El sistema envía los resultados para que el tutor lo revise
Flujos alternativos	[No aplica]
Excepciones	La pregunta indicada no es válida o no existe.
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 44 Caso de uso Consultar resultados

Nombre	Contestar encuesta
Actores	Tutor
Descripción	La parte esencial de las encuestas levantadas por el Profesor radica en que los padres las respondan.
Precondiciones	El Profesor ha creado una encuesta para el grupo del tutor.
Postcondiciones	La respuesta es almacenada y no puede ser cambiada por el tutor ni el profesor.
Flujo normal	El tutor elige de entre todas las preguntas válidas que uno o varios profesores han registrado para él. El sistema entrega las posibles respuestas El tutor elige una El sistema registra su respuesta
Flujos alternativos	[No aplica]
Excepciones	La pregunta no existe o ha expirado.
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 45 Caso de uso Contestar encuesta

Casos de uso módulo de eventos

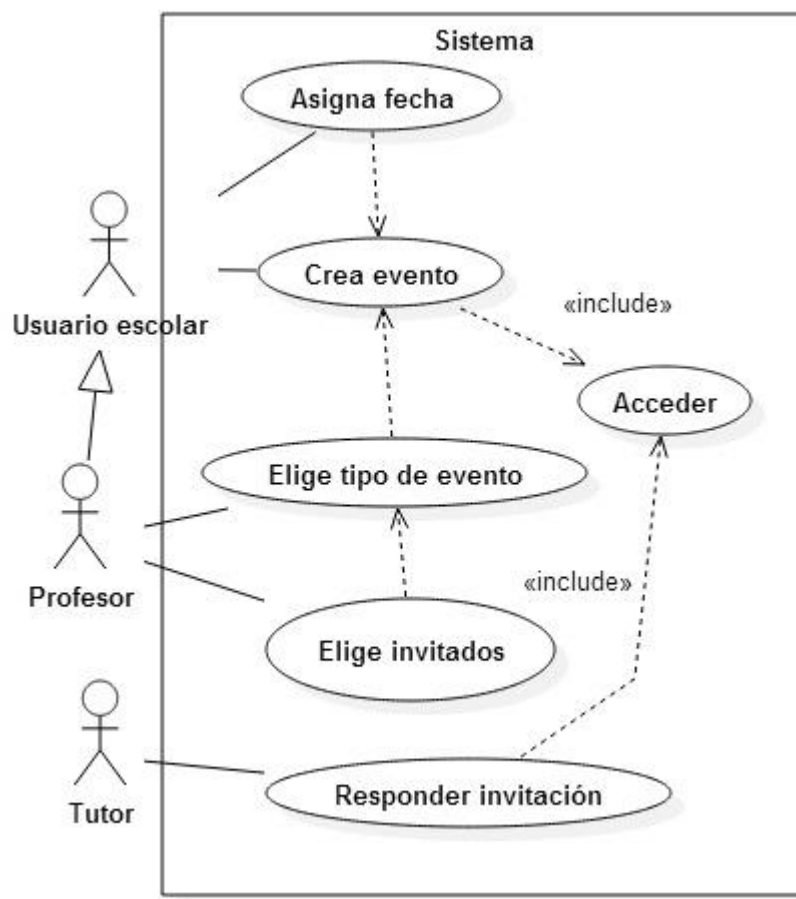


Ilustración 26 Diagrama de casos de uso del módulo de eventos

Nombre	Crear evento
Actores	Usuario escolar
Descripción	El sistema permitirá crear eventos para que los profesores se reúnan personalmente con los padres.
Precondiciones	El usuario inició sesión con su cuenta
Postcondiciones	El evento es creado y notificaciones sobre el evento son enviadas a todos los invitados informándoles del evento.
Flujo normal	El usuario elige crear un nuevo evento Ingresa en título del evento y una descripción Se envía al sistema Se validan los datos del evento y en caso de ser válidos, se da de alta. Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido
Flujos alternativos	[No aplica]
Excepciones	Que los datos no sean válidos
Prioridad	Alta

Frecuencia de uso	Alta
--------------------------	------

Tabla 46 Caso de uso Crear evento

Nombre	Asignar fecha a evento
Actores	Usuario escolar
Descripción	Todos los eventos deben tener una fecha de realización.
Precondiciones	El usuario inició sesión en el sistema y tiene un evento recién creado para asignar una fecha de realización.
Postcondiciones	La fecha de realización de un evento queda asignada.
Flujo normal	El usuario ha añadido un evento. Se introducen los datos y se envían al sistema. El sistema valida los datos. Si son válidos asigna la fecha de realización del evento. Se envía un mensaje de respuesta de acuerdo a lo sucedido.
Flujos alternativos	[No aplica]
Excepciones	Los datos enviados no son válidos como fecha.
Prioridad	Media
Frecuencia de uso	Alta

Tabla 47 Caso de uso Asignar fecha a evento

Nombre	Elegir tipo de evento
Actores	Profesor
Descripción	Un evento puede ser de dos tipos: Individual o Grupal
Precondiciones	El usuario inició sesión y tiene un evento recién creado
Postcondiciones	El tipo de evento queda asignado.
Flujo normal	El usuario ha añadido un evento. Se elige entre los dos tipos de evento. Se envía un mensaje de respuesta de acuerdo a lo sucedido.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Media
Frecuencia de uso	Alta

Tabla 48 Caso de uso Elegir tipo de evento

Nombre	Elegir invitados al evento
Actores	Profesor
Descripción	Todos los eventos deben contar con invitados para que tengan razón de ser
Precondiciones	El usuario inició sesión y tiene un evento recién creado

Postcondiciones	El evento queda cerrado y las notificaciones son enviadas a los Tutores marcados como invitados.
Flujo normal	El usuario ha añadido un evento. Según el tipo de evento el sistema entrega una lista de posibles invitados a él. Se envían los invitados. El sistema los registra y envía las notificaciones.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 49 Caso de uso Elegir invitados al evento

Nombre	Responder invitación a evento
Actores	Tutor
Descripción	Una vez que los tutores han recibido una invitación pueden confirmar o rechazar su asistencia al evento.
Precondiciones	El tutor recibió una invitación para participar en un evento.
Postcondiciones	La respuesta del tutor es registrada por el sistema.
Flujo normal	El tutor elige uno de entre todos los eventos a los que ha sido invitado. El tutor envía una confirmación sobre su participación en el evento El sistema registra su respuesta
Flujos alternativos	[No aplica]
Excepciones	El evento no existe o ha expirado.
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 50 Caso de uso Responder invitación a evento

Casos de uso módulo de mensajería

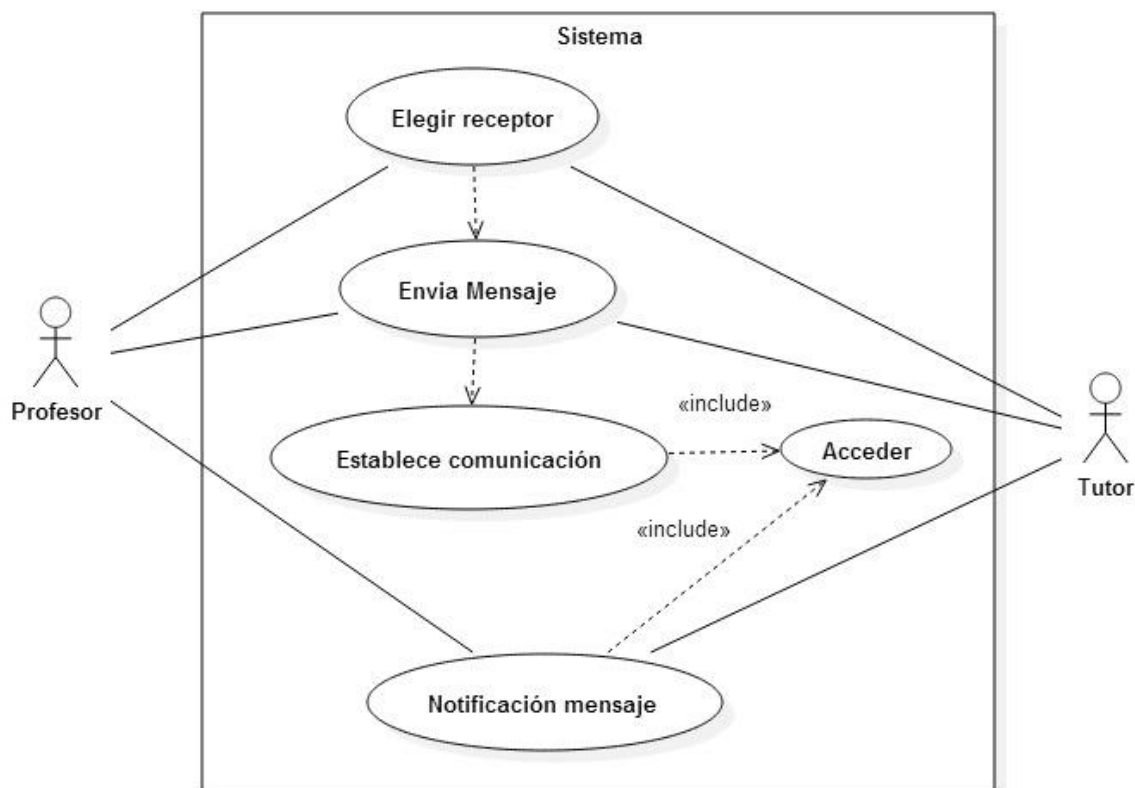


Ilustración 27 Diagrama de casos de uso del módulo de mensajería

Nombre	Enviar mensaje
Actores	Profesor/Tutor
Descripción	El sistema permitirá envía mensajería directa entre profesores y tutores.
Precondiciones	Elegir receptor.
Postcondiciones	El usuario podrá comunicarse directamente con el profesor/tutor que haya elegido.
Flujo normal	El usuario ingresa al sistema. Selecciona al profesor/tutor con el que dese establecer comunicación. Envía mensaje.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 51 Caso de uso Enviar mensaje

Nombre	Elegir receptor
--------	-----------------

Actores	Profesor/Tutor
Descripción	Todo mensaje debe tener un único receptor.
Precondiciones	El usuario deberá acceder al sistema mediante su usuario y contraseña y elegir el tutor/profesor con el que desea iniciar una comunicación.
Postcondiciones	El usuario podrá comunicarse directamente con el profesor/tutor que haya elegido.
Flujo normal	El usuario ingresa al sistema. Si es el caso del profesor selecciona el grupo, el nombre del alumno y establecerá comunicación con el tutor. Si es el caso del tutor se enlistaran únicamente los profesores que den clase a su hijo, selecciona al profesor con el que dese establecer comunicación.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 52 Caso de uso Elegir receptor

Nombre Establece comunicación	
Actores	Profesor/Tutor
Descripción	Alguna de las dos partes debe envía un mensaje para que se establezca la comunicación.
Precondiciones	Enviar un mensaje.
Postcondiciones	El usuario podrá comunicarse directamente con el profesor/tutor que haya elegido.
Flujo normal	El usuario selecciona un receptor. Envía mensaje. Se estable la comunicación.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 53 Caso de uso Establece comunicación

Nombre Notificación mensaje	
Actores	Profesor/Tutor
Descripción	Una vez establecida la comunicación y enviado el mensaje al receptor le llegara una notificación de un mensaje nuevo.
Precondiciones	Haber enviado un mensaje.
Postcondiciones	Recibir notificación de mensaje nuevo.
Flujo normal	Se envía mensaje. Se establece comunicación.

	Notificación de mensaje recibido.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 54 Caso de uso Notificación mensaje

Casos de uso módulo de notificaciones

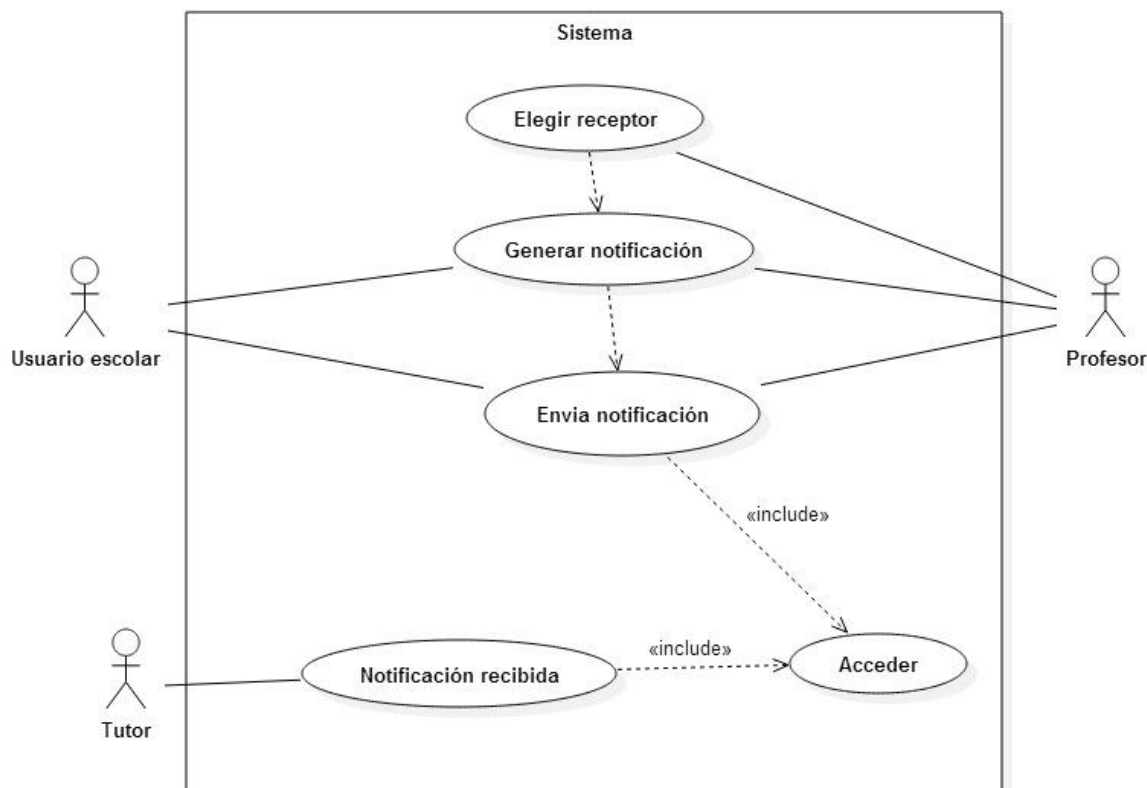


Ilustración 28 Diagrama de casos de uso del módulo de notificaciones

Nombre	Enviar notificación
Actores	Usuario escolar/Profesor
Descripción	El sistema permitirá envía notificaciones por parte de la escuela la cual les llegara a todos los tutores de la comunidad estudiantil o del profesor para los tutores como son tareas, temas vistos en clase, trabajos, etc.
Precondiciones	Elegir los tutores receptores en caso de que el profesor envíe la notificación. En caso de ser la escuela generar y enviar notificación.
Postcondiciones	El profesor podrá notificar a los tutores elegidos.

	La escuela notificara a todos los tutores de la comunidad estudiantil.
Flujo normal	Profesor <ul style="list-style-type: none"> ○ El profesor ingresa al sistema. ○ Selecciona a los tutores a los que dese notificar. ○ Envía notificación. Escuela <ul style="list-style-type: none"> ○ Ingreso al sistema. ○ Genera notificación. ○ Envía a toda la comunidad.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 55 Caso de uso Enviar notificación

Nombre	Elegir receptor
Actores	Profesor
Descripción	Toda notificación debe tener por lo menos dos receptores.
Precondiciones	El profesor deberá acceder al sistema mediante su usuario y contraseña y elegir los tutores a los que desea notificar.
Postcondiciones	El profesor podrá notificar a más de dos tutores que haya elegido.
Flujo normal	El profesor ingresa al sistema. Elegirá el grupo en el que se encuentran los tutores a los que desea notificar.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

Tabla 56 Caso de uso Elegir receptor

Nombre	Generar notificación
Actores	Usuario escolar/Profesor
Descripción	El profesor/escuela generara la notificación la cual puede ser la información de alguna tarea, trabajo, examen o información que dese comunicarle a varios tutores.
Precondiciones	Elegir receptor solo en del profesor.
Postcondiciones	Notificación enviada.
Flujo normal	Profesor <ul style="list-style-type: none"> ○ El profesor selecciona los receptores. ○ Genera notificación.

	Escuela ○ Generar notificación.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

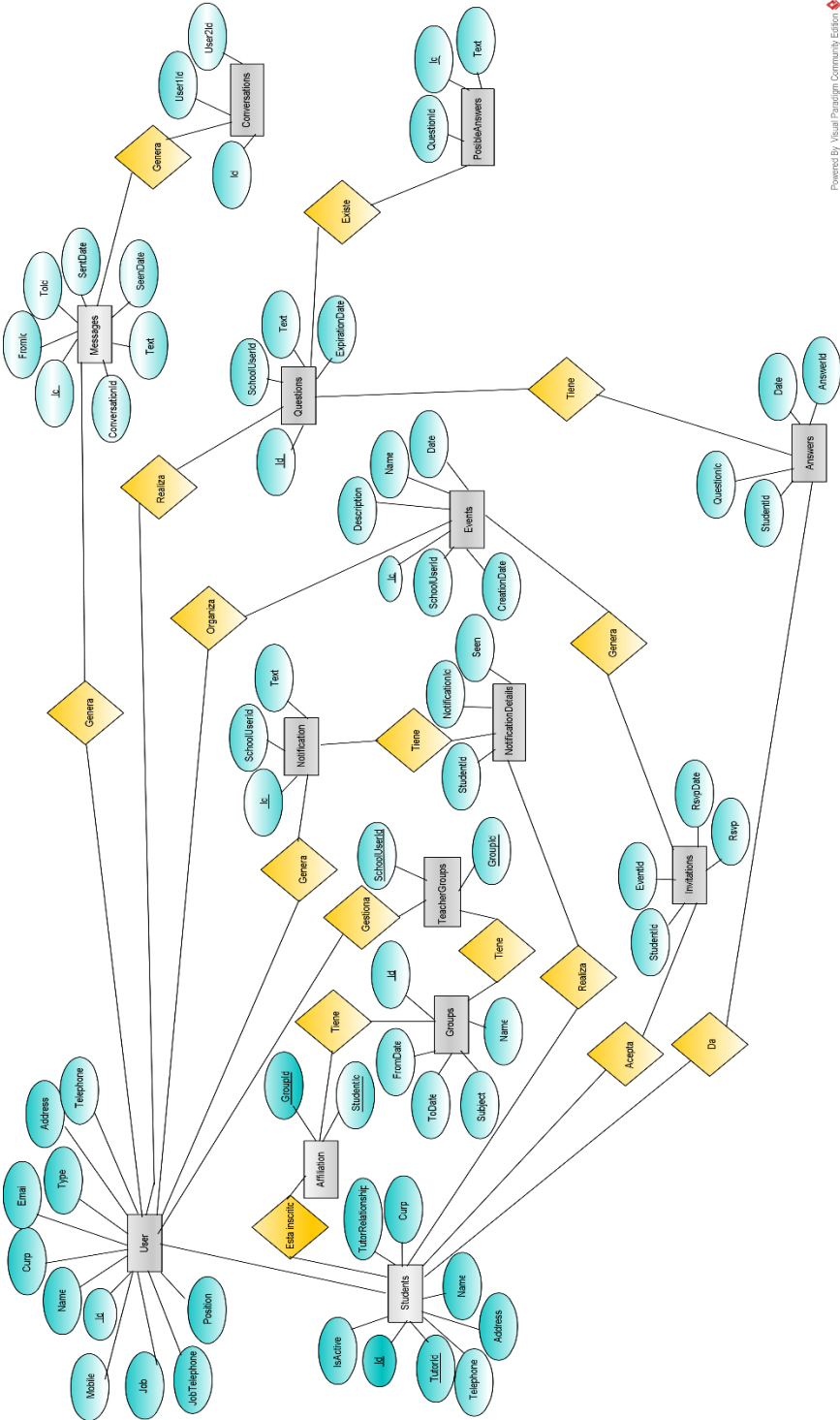
Tabla 57 Caso de uso Generar notificación

Nombre	Notificación recibida
Actores	Tutor
Descripción	Una vez generada y enviada la notificación por parte del profesor al tutor le llegara una notificación nueva.
Precondiciones	Haber recibido una notificación.
Postcondiciones	La notificación recibida se marca como vista
Flujo normal	Se genera notificación. Se envía notificación. Nueva notificación recibida.
Flujos alternativos	[No aplica]
Excepciones	[No aplica]
Prioridad	Alta
Frecuencia de uso	Alta

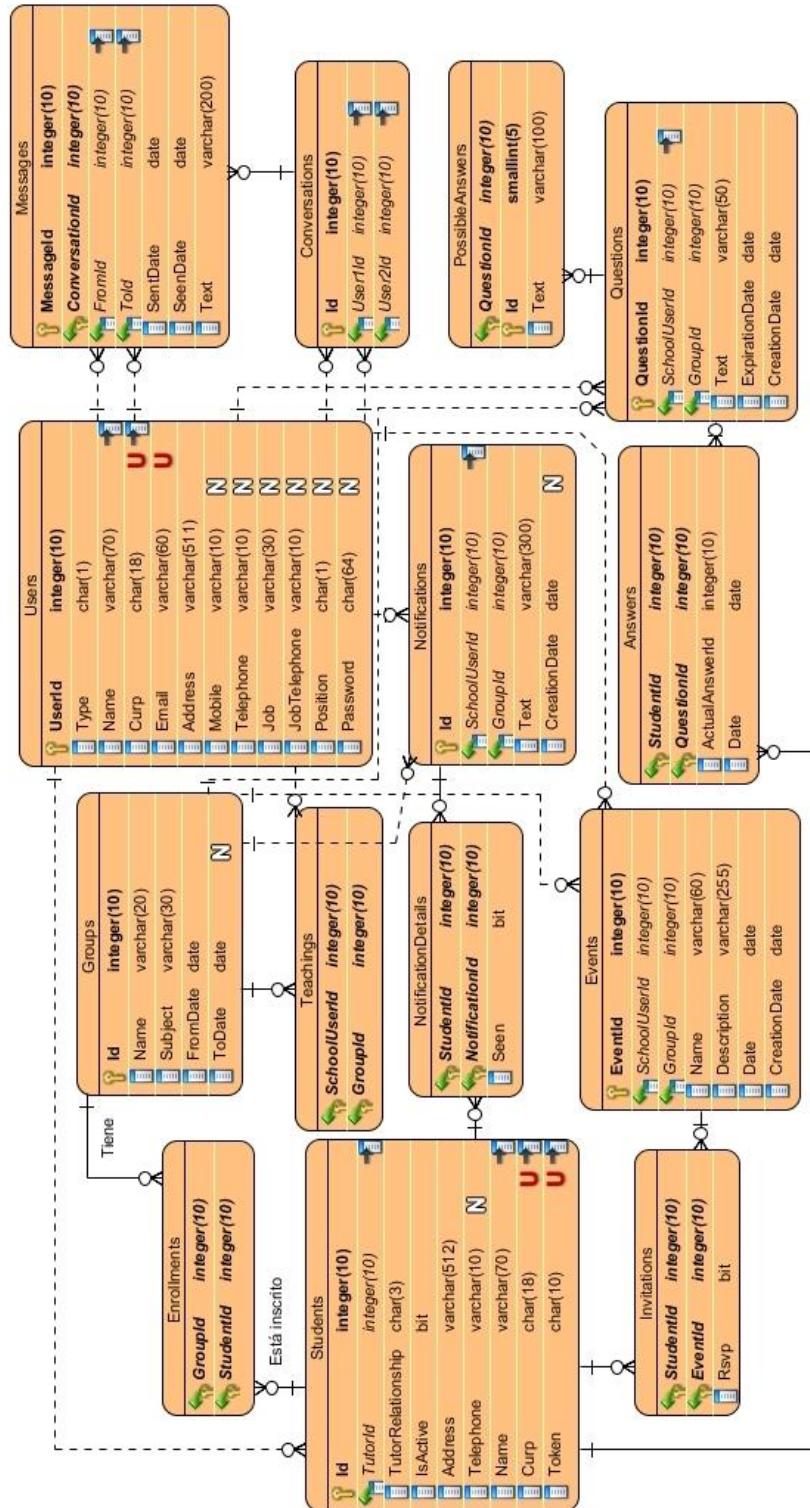
Tabla 58 Caso de uso Notificación recibida

Diseño de la base de datos

Diseño lógico de la base de datos



Diseño físico de la base de datos



Diccionario de datos

Se muestra el detalle de cada una de las tablas requeridas para el sistema, empleando la siguiente nomenclatura: **PK**: llave primaria, **FK**: llave foránea, **NN**: no nulo, **U**: índice único

Users				
Campo	Tipo dato	Longitud	Restricción	Notas
UserId	Int32	[No aplica]	PK	El identificador único de cada usuario
Type	Char	1	NN	Es un campo que identifica de qué tipo de usuario se trata Tutor (T) o Escolar (E)
Name	Varchar	60	NN	El nombre completo del usuario
Curp	Char	18	U, NN	La CURP del usuario, debe ser única dentro del sistema
Email	Varchar	50	U, NN	El email activo del usuario
Address	Varchar	511	NN	La dirección completa del usuario
Mobile	Varchar	10	[No aplica]	Número del teléfono móvil del usuario
Telephone	Varchar	10	[No aplica]	Número del teléfono fijo del usuario
Job	Varchar	20	[No aplica]	Ocupación del tutor
JobTelephone	Char	10	[No aplica]	Número telefónico de la ocupación del tutor
Position	Char	1	[No aplica]	Un único carácter que nos indica si el usuario es Administrador (A) o Profesor (P)
Password	Char	64	[No aplica]	Es la contraseña asignada a cada usuario

Tabla 59 Diccionario de la tabla Users

Students				
Campo	Tipo dato	Longitud	Restricción	Notas
StudentId	Int32	[No aplica]	PK	Es el identificador único de cada estudiante
TutorId	Int32	[No aplica]	FK, NN	El id del tutor que corresponde a este estudiante, se corresponde con

				<i>TutorId</i> en la tabla <i>Tutors</i>
TutorRelationship	Char	3	DF	La relación que el tutor tiene con ese alumno
Name	Varchar	60	NN	El nombre completo del estudiante
IsActive	Bit		NN	Indica si el estudiante está activo para este periodo escolar
Address	Varchar	511	NN	La dirección completa del estudiante
Curp	Char	18	NN, U	La CURP del estudiante, debe ser única dentro del sistema
Token	Char	10	NN, U	El token único generado para el estudiante
Phone	Varchar	10	[No aplica]	El número de teléfono fijo del estudiante

Tabla 60 Diccionario de la tabla *Students*

Groups				
Campo	Tipo dato	Longitud	Restricción	Notas
GroupId	Int32	[No aplica]	PK	El identificador único del grupo dentro del sistema
Name	Varchar	20	NN	El nombre del grupo
Subject	Varchar	35	[No aplica]	La materia del grupo
FromDate	Datetime	[No aplica]	NN	La fecha de inicio del grupo
ToDate	Datetime	[No aplica]	[No aplica]	La fecha en la que expira el grupo

Tabla 61 Diccionario de la tabla *Groups*

Campo	Tipo dato	Longitud	Restricción	Notas
GroupId	Int32	[No aplica]	PK, FK	El identificador único del grupo dentro del sistema, se corresponde con <i>GroupId</i> en la tabla <i>Groups</i>

StudentId	Int32	[No aplica]	PK, FK	El identificador único del estudiante dentro del sistema, se corresponde con <i>StudentId</i> en la tabla <i>Students</i>
------------------	-------	-------------	--------	---

Tabla 62 Diccionario de la tabla *Enrollments*

Teachings				
Campo	Tipo dato	Longitud	Restricción	Notas
SchoolUserId	Int32	[No aplica]	PK, FK	El identificador único del grupo dentro del sistema, se corresponde con <i>UserId</i> en la tabla <i>SchoolUsers</i>
GroupId	Int32	[No aplica]	PK, FK	El identificador único del grupo dentro del sistema, se corresponde con <i>GroupId</i> en la tabla <i>Groups</i>

Tabla 63 Diccionario de la tabla *TeacherGroup*

Notifications				
Campo	Tipo dato	Longitud	Restricción	Notas
NotificationId	Int32	[No aplica]	PK	El identificador único de la notificación dentro del sistema
SchoolUserId	Int32	[No aplica]	FK	El identificador único del usuario que generó la notificación, se corresponde con <i>UserId</i> en la tabla <i>SchoolUsers</i>
GroupId	Int32	[No aplica]	FK, NN	El identificador único del grupo con el que la notificación está asociada
Text	Varchar	300	NN	El texto de la notificación

Tabla 64 Diccionario de la tabla *Notifications*

NotificationDetails				
Campo	Tipo dato	Longitud	Restricción	Notas
StudentId	Int32	[No aplica]	PK, FK	El identificador único de un estudiante dentro del sistema, se corresponde con <i>StudentId</i> en la tabla <i>Students</i>

NotificationId	Int32	[No aplica]	PK, FK	El identificador único de la notificación dentro del sistema, se corresponde con <i>NotificationId</i> en la tabla <i>Notifications</i>
Seen	Bit	[No aplica]	NN	Nos indica si el usuario ha visto o no la notificación

Tabla 65 Diccionario de la tabla *NotificationDetails*

Events				
Campo	Tipo dato	Longitud	Restricción	Notas
EventId	Int32	[No aplica]	PK	El identificador único del evento dentro del sistema
SchoolUserId	Int32	[No aplica]	PK, FK	El identificador único del usuario que generó el evento, se corresponde con <i>UserId</i> en la tabla <i>SchoolUsers</i>
GroupId	Int32	[No aplica]	FK, NN	El identificador único del grupo con el que la notificación está asociada
Name	Varchar	30	NN	El nombre del evento
Date	Datetime	[No aplica]	NN	La fecha en la que el evento se llevará a cabo
Description	Varchar	255	NN	Una descripción breve del evento, como la temática o indicaciones extra
CreationDate	Datetime	[No aplica]	NN	La fecha en la que el evento fue creado por el usuario

Tabla 66 Diccionario de la tabla *Events*

Invitations				
Campo	Tipo dato	Longitud	Restricción	Notas
StudentId	Int32	[No aplica]	PK,FK	El identificador único de un estudiante dentro del sistema, se corresponde con <i>StudentId</i> en la tabla <i>Students</i>
EventId	Int32	[No aplica]	PK, FK	El identificador único de un evento del sistema, se corresponde con <i>EventId</i> en la tabla <i>Events</i>

Rsvp	Bit	[No aplica]	NN	Indica si el usuario asistirá o no al evento, por default es 0
-------------	-----	-------------	----	--

Tabla 67 Diccionario de la tabla *Invitations*

Questions				
Campo	Tipo dato	Longitud	Restricción	Notas
QuestionId	Int32	[No aplica]	PK	El identificador único de una pregunta dentro del sistema
SchoolUserId	Int32	[No aplica]	FK	El identificador único del usuario que generó el evento, se corresponde con <i>UserId</i> en la tabla <i>SchoolUsers</i>
GroupId	Int32	[No aplica]	FK, NN	El identificador único del grupo con el que la notificación está asociada
Text	Varchar	50	NN	El texto de la pregunta
ExpirationDate	Datetime	[No aplica]	NN	La fecha en la que la pregunta expirará y no podrá ser contestada

Tabla 68 Diccionario de la tabla *Questions*

PossibleAnswers				
Campo	Tipo dato	Longitud	Restricción	Notas
QuestionId	Int32	[No aplica]	PK,FK	El identificador único de una pregunta dentro del sistema, se corresponde con <i>QuestionId</i> en la tabla <i>Questions</i>
PossibleAnswerId	Int32	[No aplica]	PK	El identificador de la respuesta posible en relación a la pregunta
Text	Varchar	100	NN	El texto de la respuesta

Tabla 69 Diccionario de la tabla *PossibleAnswers*

Answers				
Campo	Tipo dato	Longitud	Restricción	Notas
QuestionId	Int32	[No aplica]	PK,FK	El identificador único de una pregunta dentro del

				sistema, se corresponde con <i>QuestionId</i> en la tabla <i>Questions</i>
StudentId	Int32	[No aplica]	PK, FK	El identificador único de un estudiante dentro del sistema, se corresponde con <i>StudentId</i> en la tabla <i>Students</i>
ActualAnswerId	Int32	[No aplica]	[No aplica]	El identificador de la respuesta debe corresponderse con uno de los posibles valores de <i>PossibleAnswerId</i> en la tabla <i>PossibleAnswers</i> cuando los valores de <i>QuestionId</i> sean iguales
AnswerDate	Datetime	[No aplica]	[No aplica]	La fecha en la que fue contestada la pregunta

Tabla 68 Diccionario de la tabla *Answers*

Conversations				
Campo	Tipo dato	Longitud	Restricción	Notas
ConversationId	Int32	[No aplica]	PK	El identificador único de una conversación dentro del sistema
User1Id	Int32	[No aplica]	FK	El identificador único de un usuario participante, se corresponde con <i>UserId</i> en la tabla <i>Users</i>
User2Id	Int32	[No aplica]	FK	El identificador único de un usuario participante, se corresponde con <i>UserId</i> en la tabla <i>Users</i>

Tabla 69 Diccionario de la tabla *Conversations*

Messages				
Campo	Tipo dato	Longitud	Restricción	Notas
MessageId	Int32	[No aplica]	PK	El identificador único de un mensaje dentro del sistema
ConversationId	Int32	[No aplica]	PK, FK	El identificador único de una conversación dentro del sistema, se corresponde con

				<i>ConversationId</i> en la tabla <i>Conversations</i>
FromId	Int32	[No aplica]	FK	El identificador único del usuario que envía el mensaje, se corresponde con <i>UserId</i> en la tabla <i>Users</i>
ToId	Int32	[No aplica]	FK	El identificador único del usuario que recibe el mensaje, se corresponde con <i>UserId</i> en la tabla <i>Users</i>
SentDate	Datetime	[No aplica]	NN	La fecha en que el mensaje fue enviado por el usuario emisor
SeenDate	Datetime	[No aplica]	[No aplica]	La fecha en que el mensaje fue recibido por el usuario receptor
Text	Varchar	200	NN	El contenido del mensaje

Tabla 70 Diccionario de la tabla Messages

Diagramas de clases

Se presentan a continuación un esbozo de las clases principales que componen al sistema, divididas de manera lógica de acuerdo a la relación que guardan entre ellas.

Edutor.Web.Api.Models

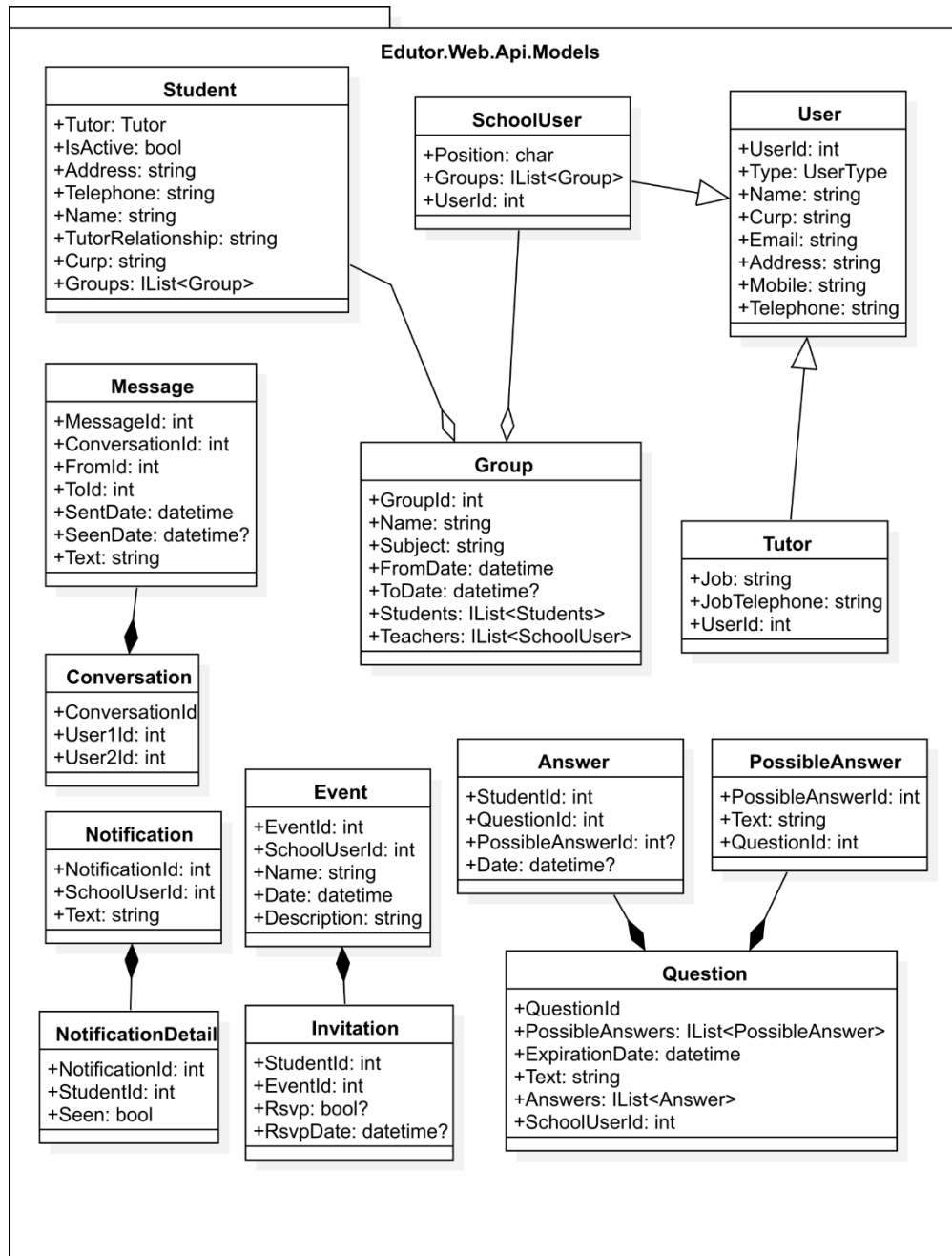


Ilustración 29 Diagrama de clases para *Edutor.Web.Api.Models*

Edutor.Web.Api

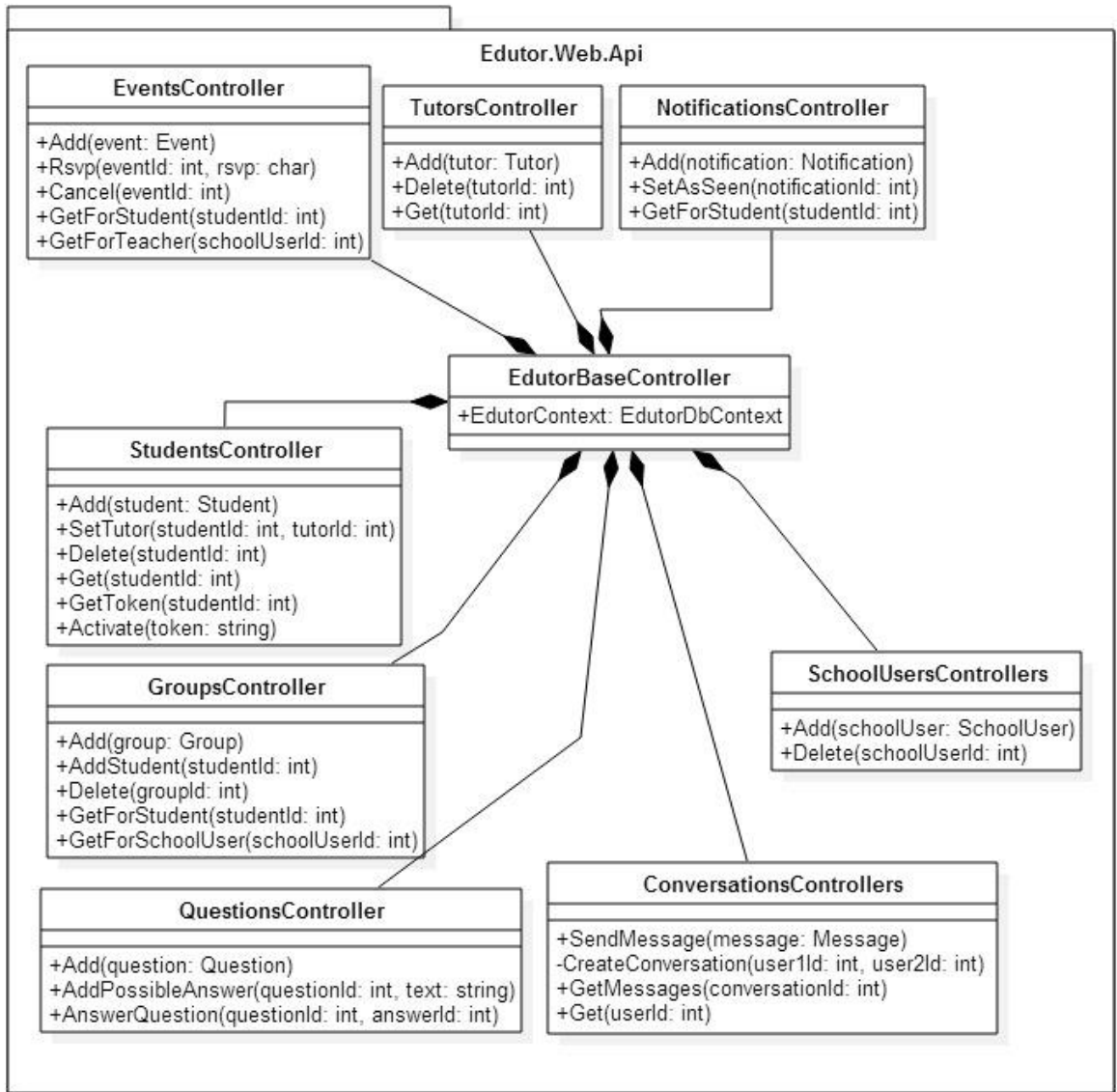


Ilustración 30 Diagrama de clases para *Edutor.Web.Api*

Diagrama de paquetes

Se muestra la estructura general del sistema, se omiten los paquetes que brindan los frameworks de desarrollo. Posteriormente se explica cada paquete a detalle.

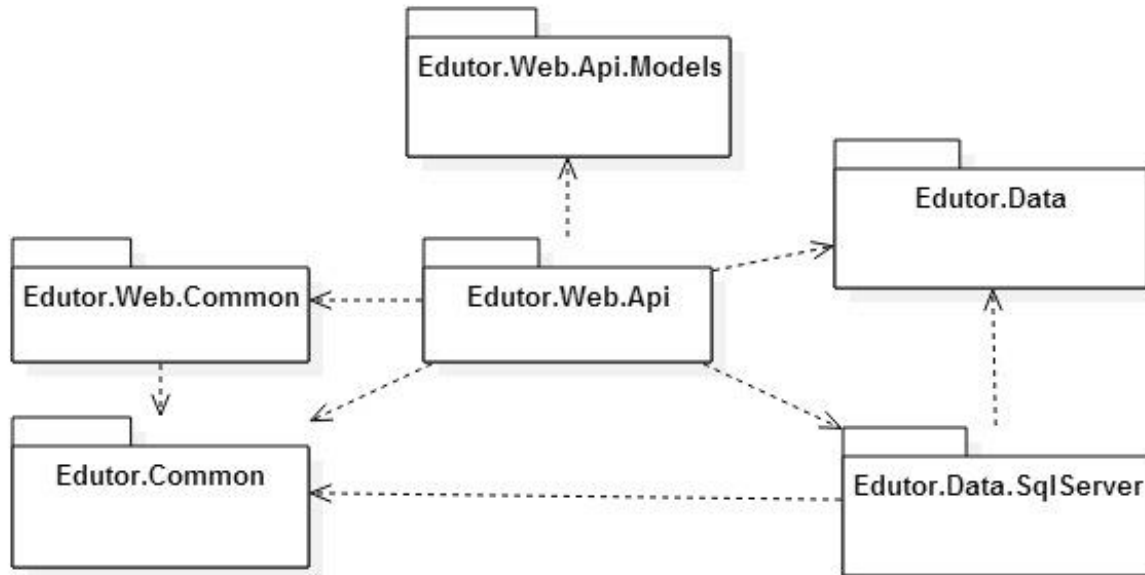


Ilustración 31 Diagrama de paquetes general

Edutor.Common

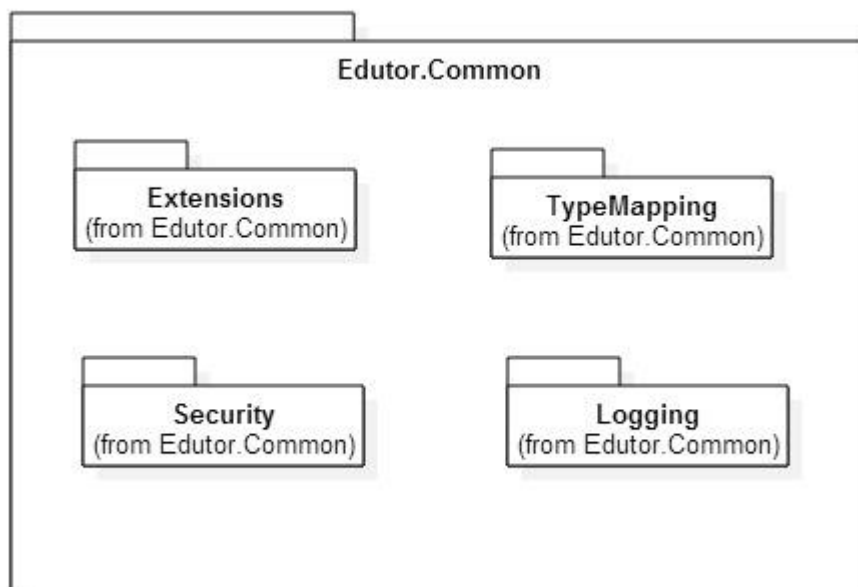


Ilustración 32 Diagrama de paquetes para Edutor.Common

Contiene elementos que no están directamente relacionados ni con el acceso a datos ni con la aplicación, ofrecen extensiones, creación de logs y algunos elementos de seguridad.

Edutor.Web.Common

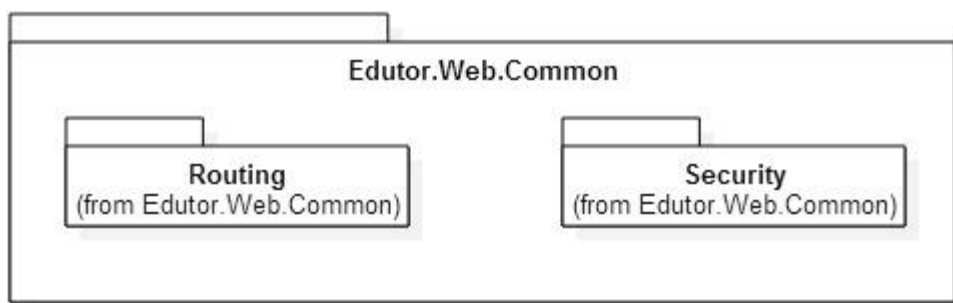


Ilustración 33 Diagrama de paquetes para Edutor.Web.Common

Contiene elementos comunes a la capa expuesta a la web. Se divide a su vez en dos paquetes, *Routing* y *Security* los cuales se encargan de:

- **Routing:** Ofrecer un servicio de ruteo a partir de la URL para dirigir al cliente al recurso adecuado
- **Security:** Ofrece los componentes de seguridad necesarios a la capa web.

Edutor.Data

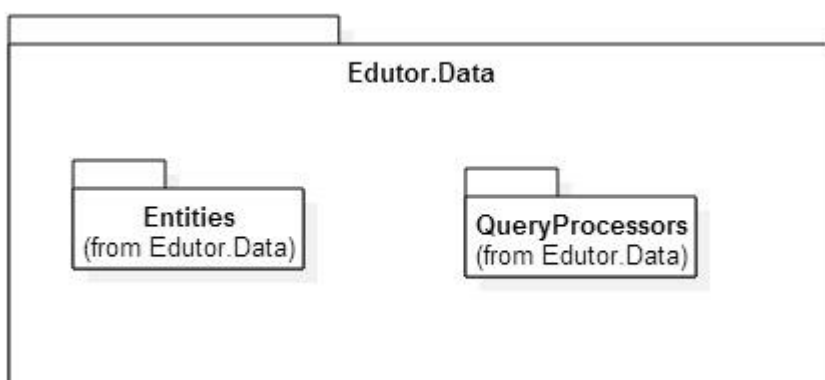


Ilustración 34 Diagrama de paquetes para Edutor.Data

Contiene los objetos del dominio que son usados por *NHibernate* para obtener y almacenar a la base de datos, estos están contenidos en el paquete **Entities**. También contiene las interfaces que definen los comportamientos y algunas clases auxiliares en el manejo de los datos dentro del

paquete **QueryProcessors**. Cabe destacar que nada de este paquete tiene alguna relación con el sistema gestor de base de datos escogido.

Edutor.Data.SqlServer

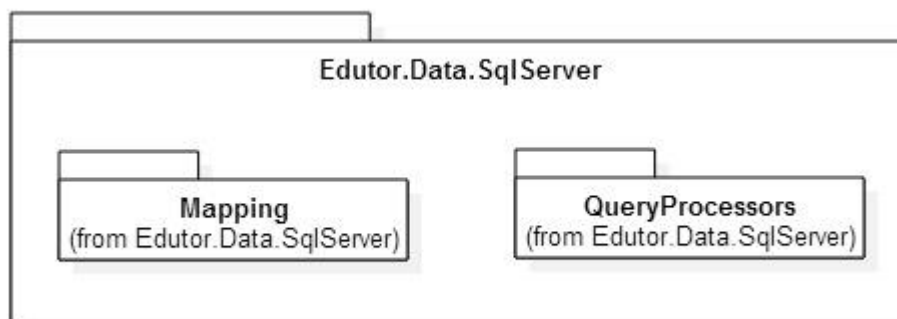


Ilustración 35 Diagrama de paquetes para Edutor.Data.SqlServer

Contiene las implementaciones del paquete **Edutor.Data.QueryProcessors**, aquí se implementan los comportamientos dentro del paquete **QueryProcessors**. También provee las clases de mapeo de *NHibernate* contenidas dentro del paquete **Mapping**. Los recursos contenidos en este proyecto están completamente ligado con la instancia de base de datos elegida, en este caso SQL Server.

Edutor.Web.Api.Models



Ilustración 36 Diagrama de paquetes para Edutor.Web.Api.Models

Los modelos que son expuestos a través del servicio web, se crearon en un paquete separado para facilitar la creación de test unitarios y para una mejor comprensión del código.

Edutor.Web.Api

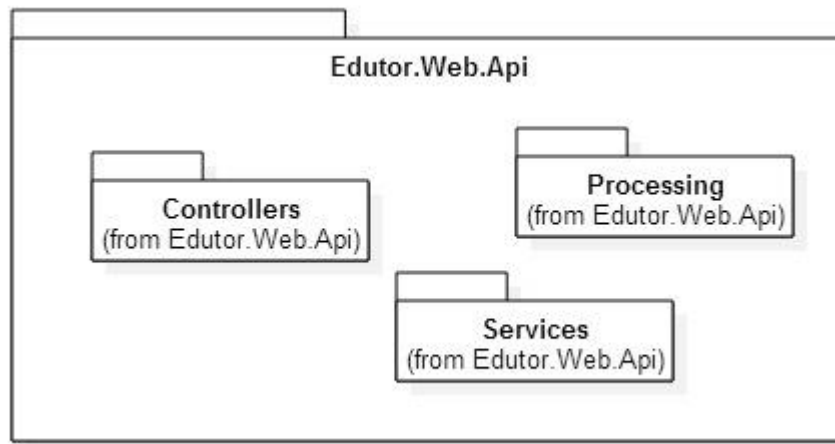


Ilustración 37 Diagrama de paquetes para Edutor.Web.Api

Este paquete es el corazón de la aplicación, contiene la aplicación REST y es el paquete que está expuesto al público para ser consumido. Este paquete depende de la existencia de los anteriores para funcionar.

Diagrama de secuencia

A continuación se muestra los diagramas de secuencias en el cual se muestra la interacción de un conjunto de objetos. Debido que se utiliza el patrón de diseño “Modelo – Vista – Controlador”, se utiliza la siguiente notación UML para dicho patrón.

Notación	Nombre	Descripción
	Controlador	Representa al controlador dentro de la arquitectura son las funciones que interactúan con el modelo y la vista.
	Vista	Representa la vista dentro de la arquitectura y es la interfaz gráfica de usuario con la que se permite la interacción entre el sistema y el usuario.
	Base de Datos	Representa el modelo dentro de la arquitectura, y está enfocado a las reglas de negocio que se ven respetadas dentro de la base de datos.

Tabla 71 Nomenclatura de los diagramas

Diagrama de Secuencia para Acceso

El flujo que sigue este diagrama de secuencia es ingresar los datos que le fueron otorgados, el sistema lo valida y en caso de ser correctos se otorgará acceso al sistema y a las peticiones que desee realizar. En caso contrario se envía un mensaje de error.

Es importante mencionar que este diagrama de secuencia se ejecutara cada vez que se realice una petición al sistema.

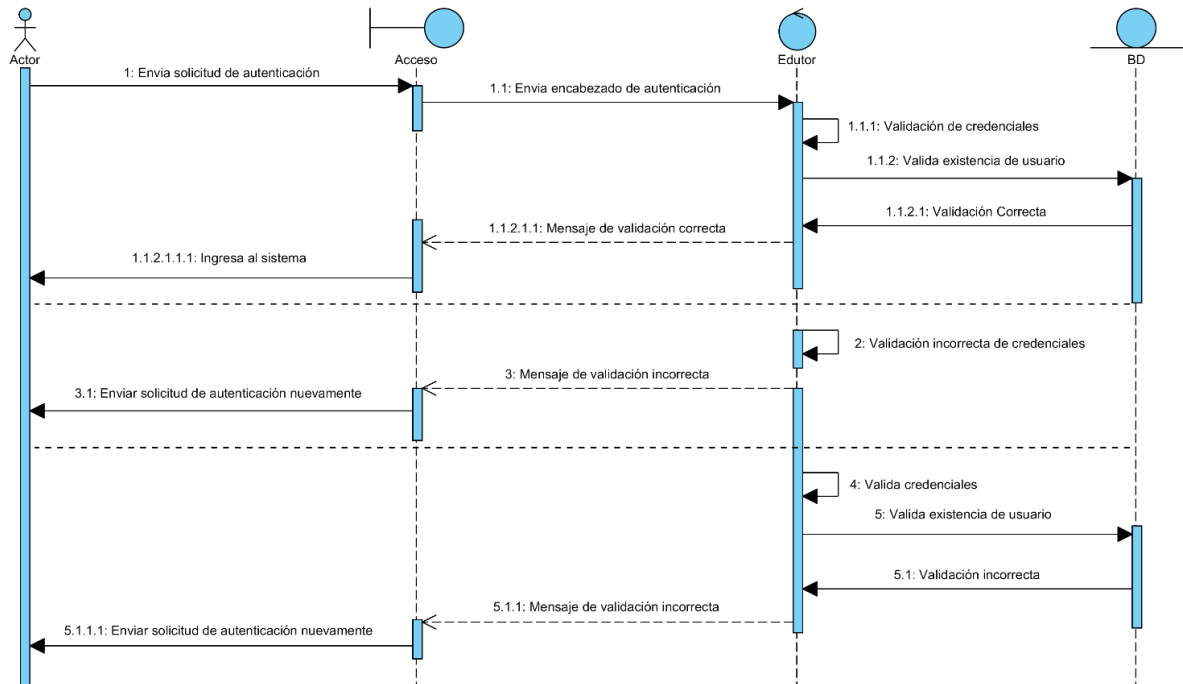


Ilustración 38 Diagrama de secuencia para Acceso

Diagrama de Secuencia para Alta de alumno

El alta de un alumno será responsabilidad del administrador, quien ingresara los datos del alumno y los enviara al sistema, este los validara y verificara que no existan en la base de datos para proceder a insertarlo y finalmente enviar un mensaje al usuario confirmando el alta del alumno. Si la validación de los datos no es correcta se envía un mensaje de error.

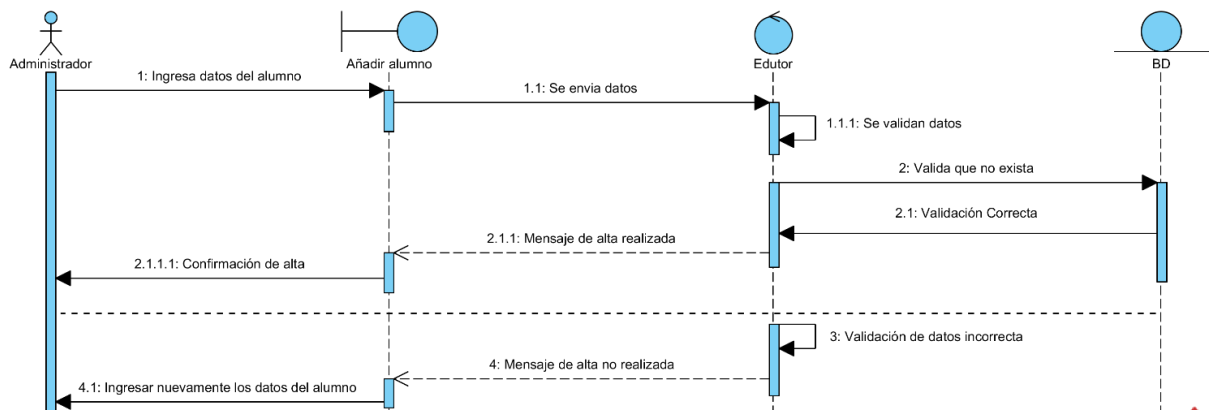


Ilustración 39 Diagrama de secuencia para ingresar alumno

Diagrama de Secuencia para Modificar Alumno

El administrador indicara al sistema el alumno que desea modificar, el sistema regresa los datos modificables y el usuario los modifica, se envían al sistema y se validan, si la validación es correcta se actualizan en la base de datos y se envía un mensaje al usuario confirmando la actualización. En caso de que la validación no sea favorable se envía un mensaje de error.

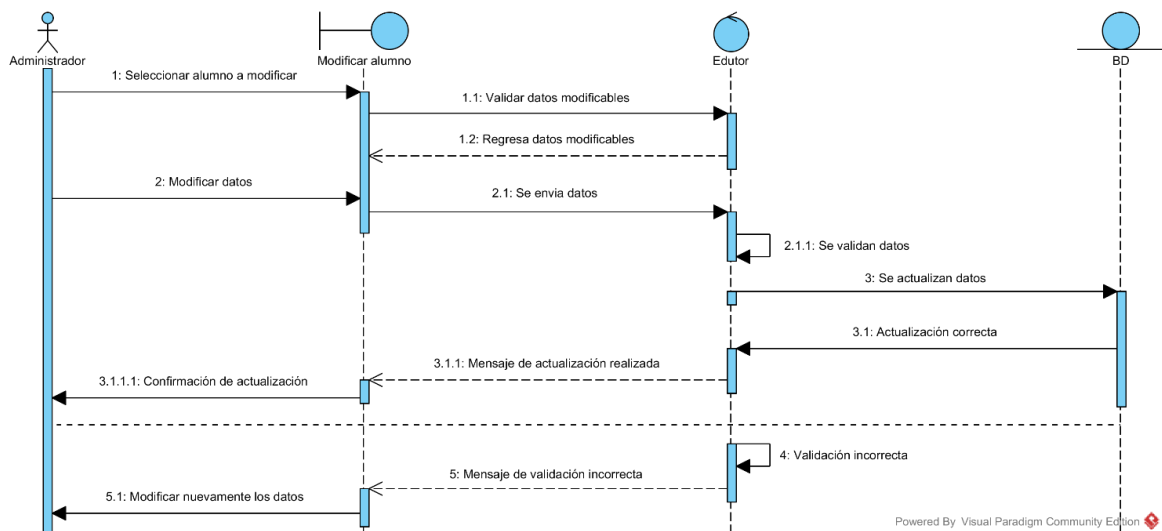


Ilustración 40 Diagrama de secuencia para Modificar Alumno

Diagrama de Secuencia para Eliminar un Alumno

Se indica al sistema el alumno que se desea eliminar, valida la posibilidad de eliminar al alumno, si es válido eliminar un alumno, se elimina y se envía mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

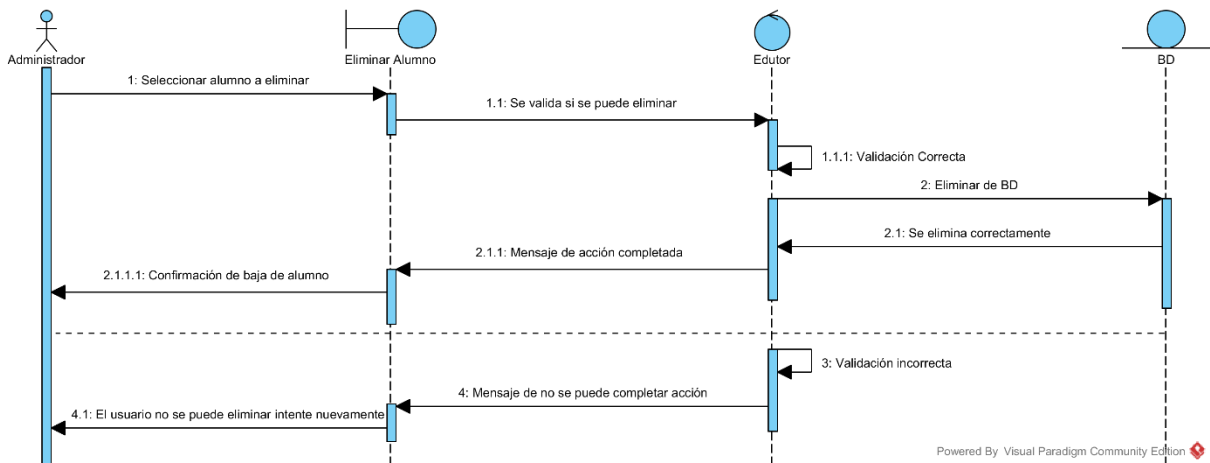


Ilustración 41 Diagrama de secuencia para Eliminar un Alumno

Diagrama de Secuencia para Añadir un Profesor

Se ingresan los datos del profesor, se envían al sistema este lo valida y si son válidos y no existe en el sistema, se da de alta. Finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

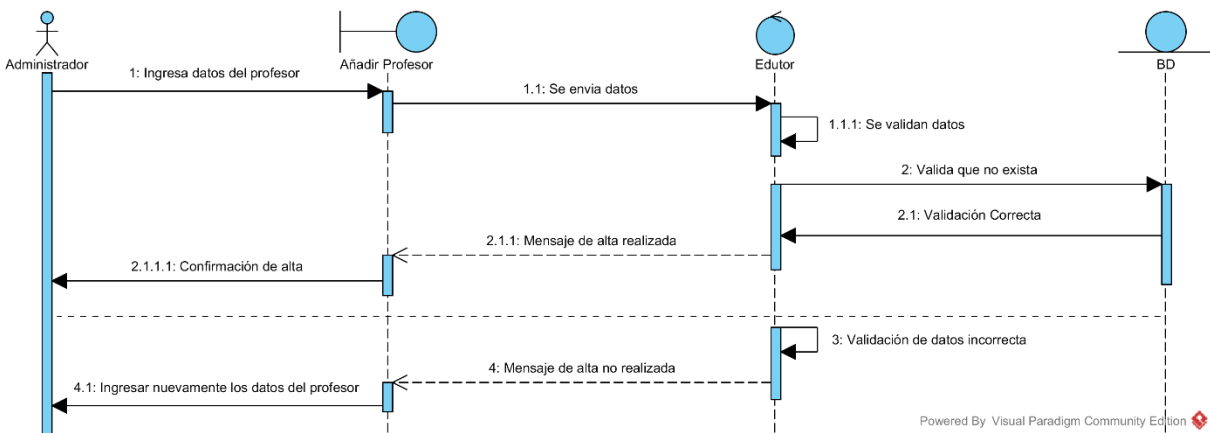


Ilustración 42 Diagrama de secuencia para Añadir un Profesor

Diagrama de Secuencia para Modificar un Profesor

Se indica al sistema el profesor que se desea modificar, se regresan los datos modificables, el usuario modifica los datos. Se envían los datos al sistema y se validan, si los datos son válidos, las modificaciones se llevan a cabo al final se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

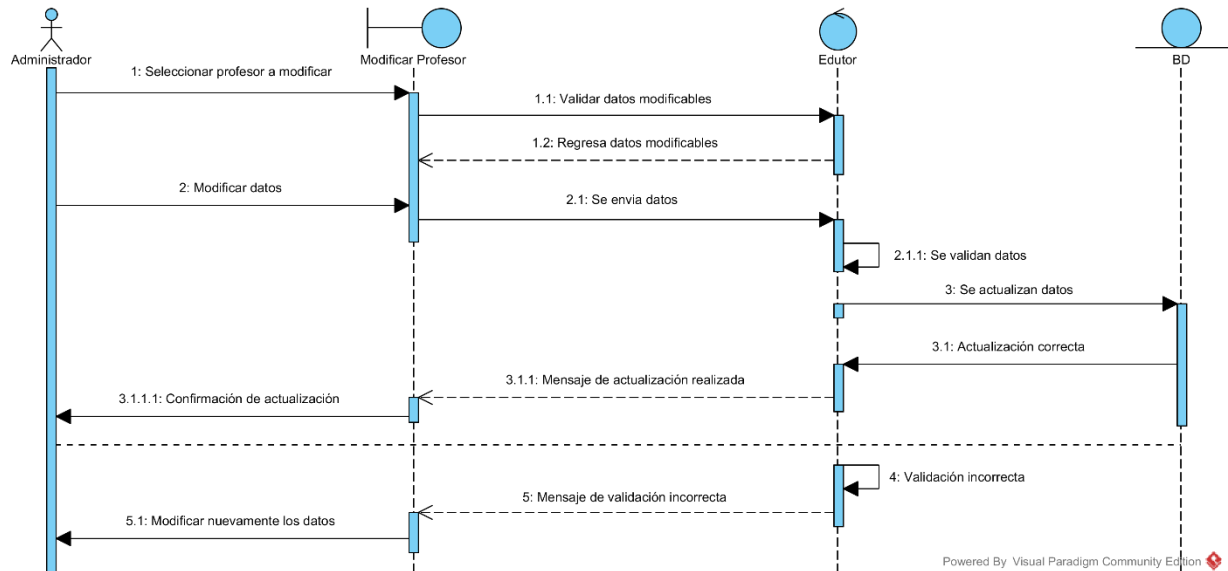


Ilustración 43 Diagrama de secuencia para Modificar un Profesor

Diagrama de Secuencia para Eliminar un Profesor

Se indica al sistema el profesor que se desea eliminar, este valida la posibilidad de eliminar al profesor y si es válido eliminar al profesor, se elimina finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

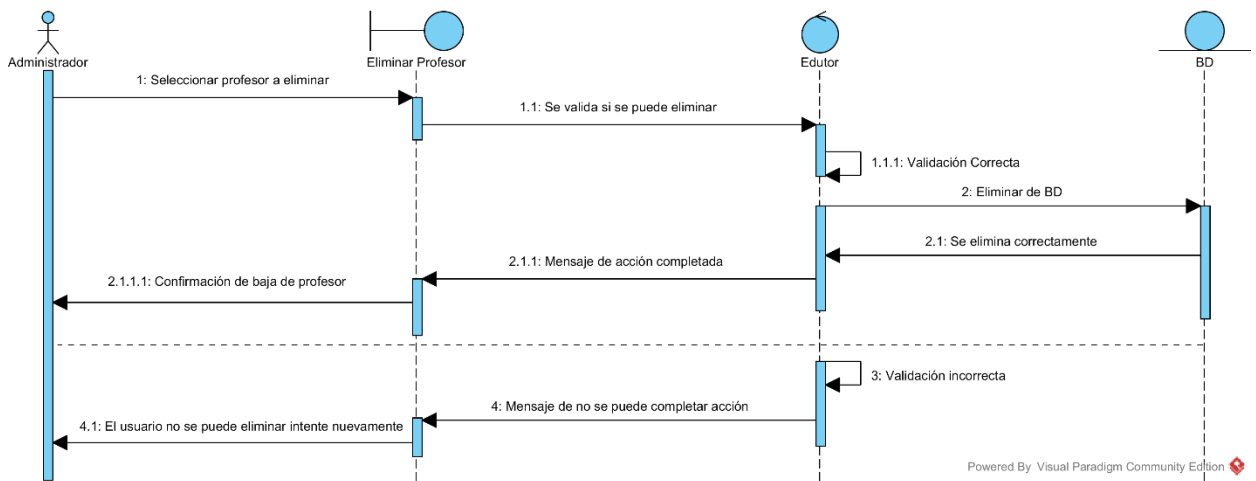


Ilustración 44 Diagrama de secuencia para Eliminar un Profesor

Diagrama de Secuencia para Añadir un Grupo

Se ingresan los datos del grupo y se envían al sistema este lo valida y si los datos son válidos y no existe en el sistema, se da de alta finalmente envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

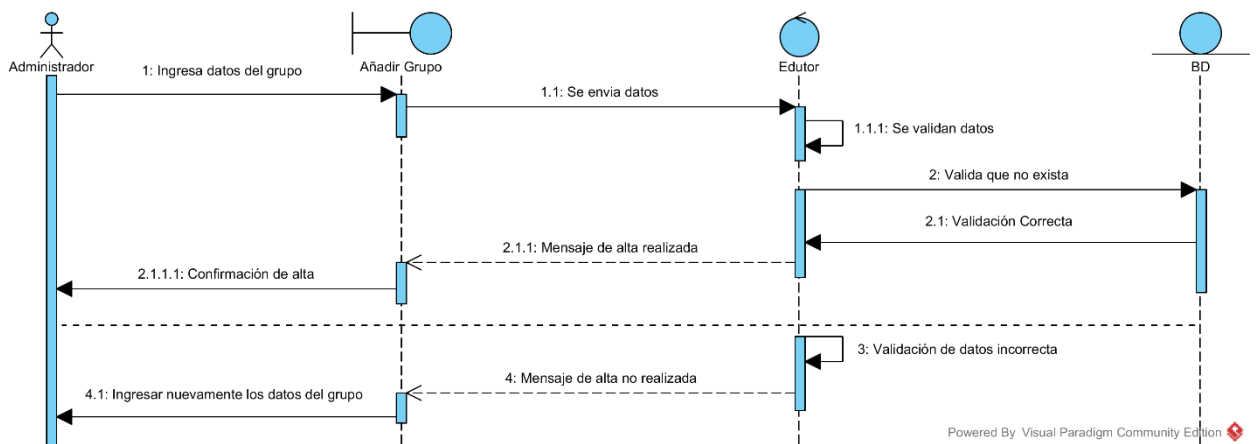


Ilustración 45 Diagrama de secuencia para Añadir un Grupo

Diagrama de Secuencia para Modificar un Grupo

Se indica al sistema el grupo que se desea modificar se muestran los datos modificables, el usuario modifica los datos y se envían los datos al sistema este los valida y si los datos son válidos, las

modificaciones se llevan a cabo finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

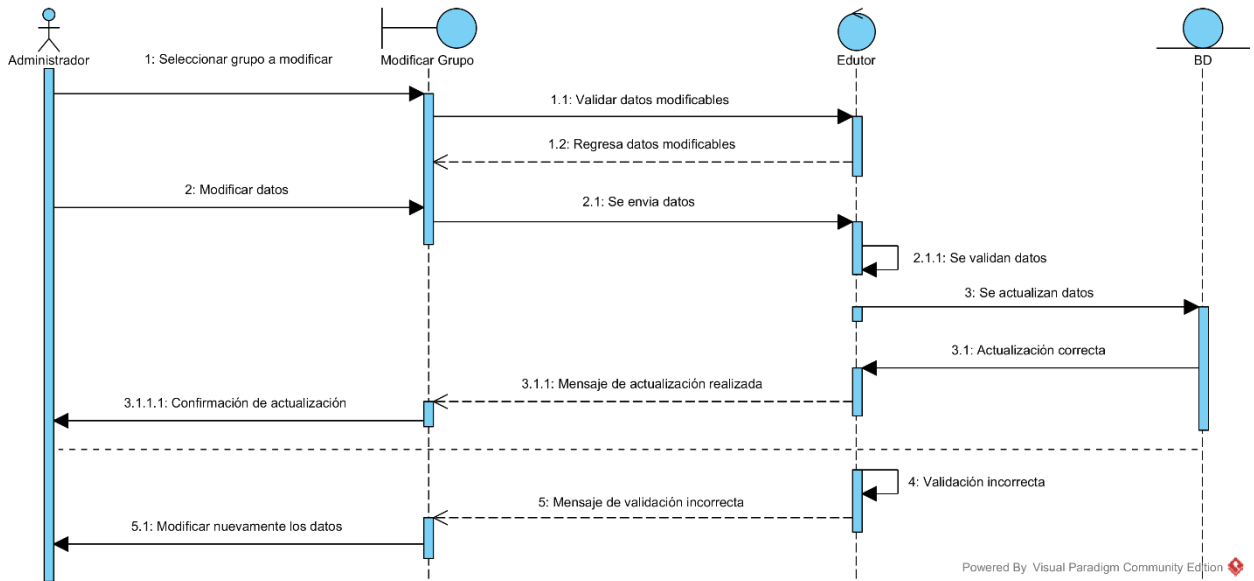


Ilustración 46 Diagrama de secuencia para Modificar un Grupo

Diagrama de Secuencia para Eliminar un Grupo

Se indica al sistema el grupo que se desea eliminar, valida la posibilidad de eliminar el grupo, si es válido eliminar el grupo, se elimina y finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

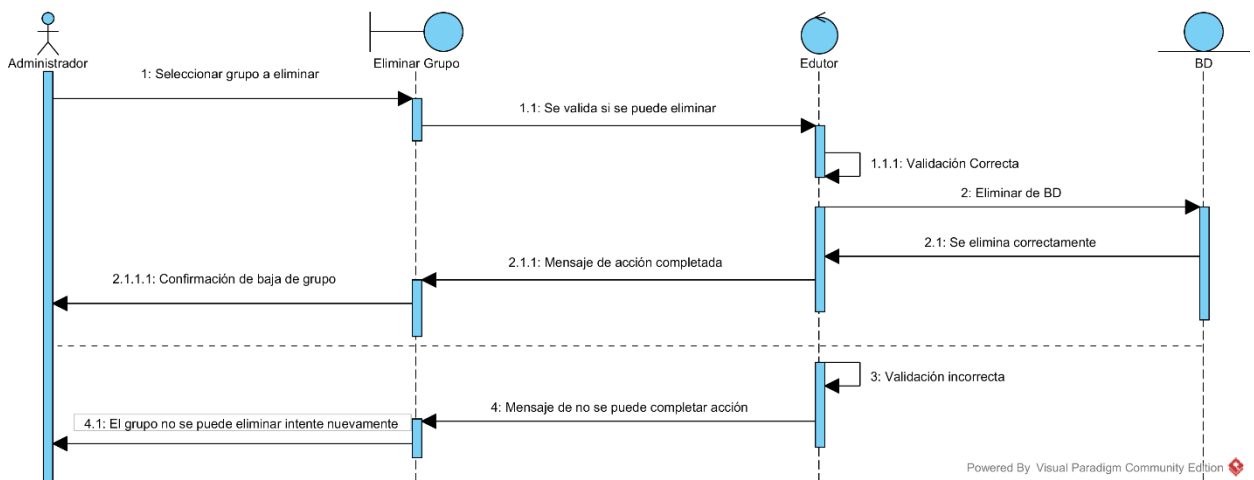


Ilustración 47 Diagrama de secuencia para Eliminar un Grupo

Diagrama de Secuencia para Vincular un Profesor con un Grupo

Se elige el ID del grupo, el ID del profesor, se envían los datos al sistema y se validan si son válidos, las modificaciones se llevan a cabo y se envía un mensaje de respuesta al usuario dependiendo de lo sucedido.

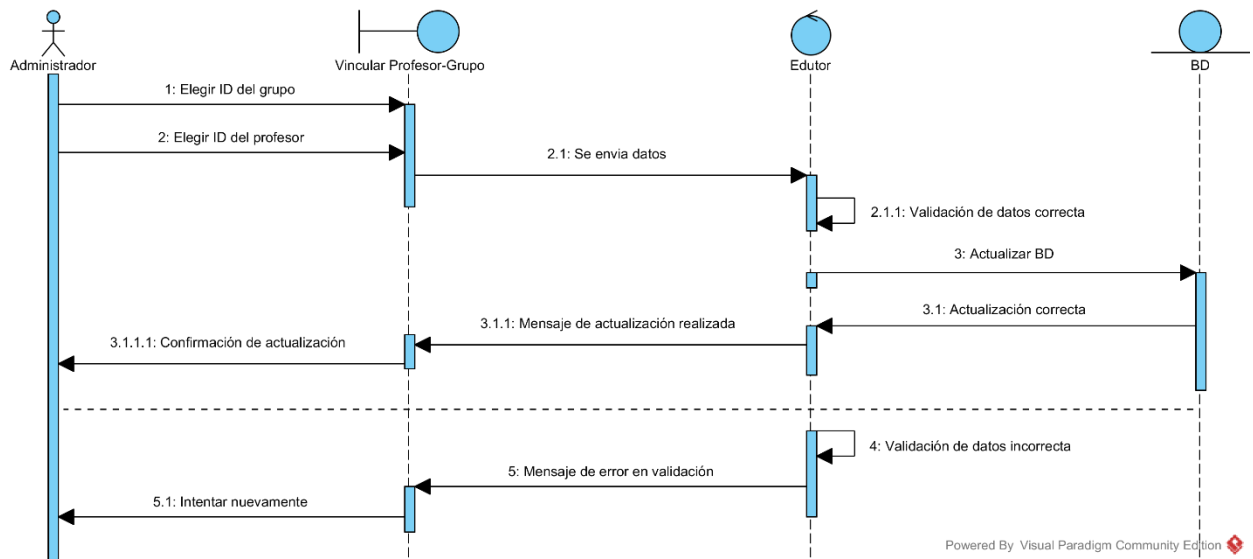


Ilustración 48 Diagrama de secuencia para Vincular un Profesor con un Grupo

Diagrama de Secuencia para Desvincular un Profesor de un Grupo

Se elige el ID del grupo, el ID del profesor y se envían los datos al sistema, se validan y si los datos son válidos, las modificaciones se llevan a cabo después se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

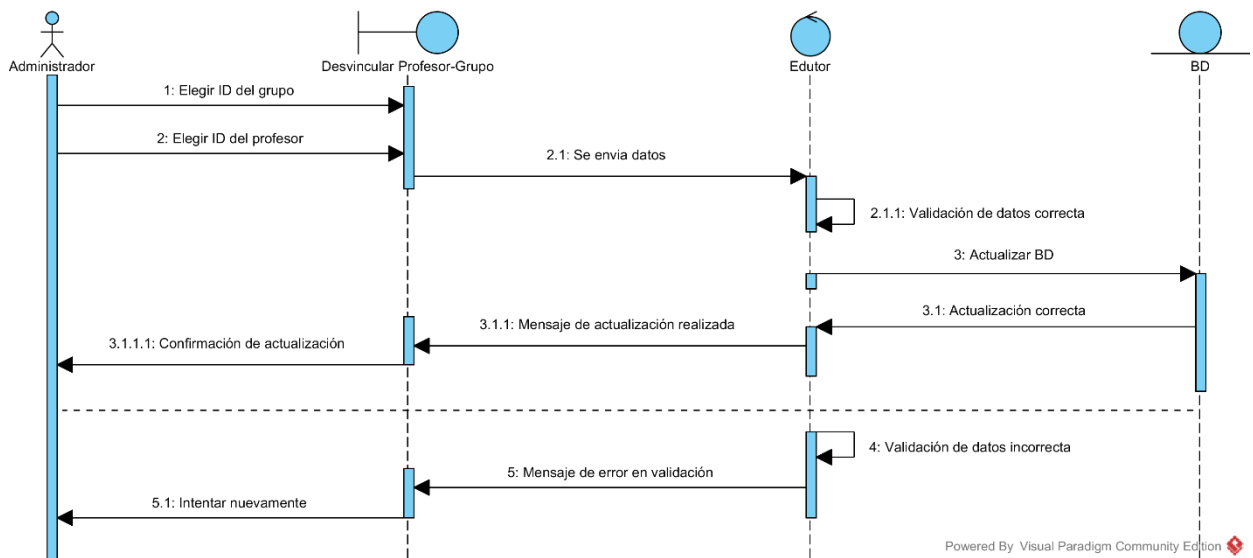


Diagrama de Secuencia para Vincular un Alumno con un Grupo

Se elige el ID del grupo, el ID del alumno, se envían los datos al sistema y se validan si son válidos, las modificaciones se llevan a cabo y se envía un mensaje de respuesta al usuario dependiendo de lo sucedido.

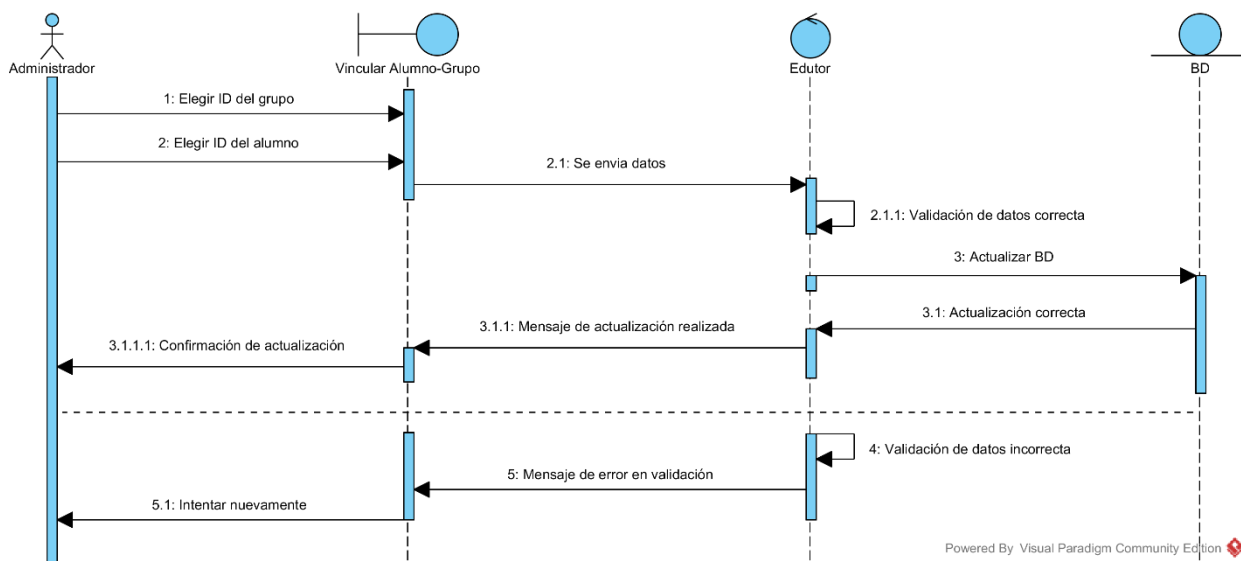


Ilustración 50 Diagrama de secuencia para Vincular un Alumno con un Grupo

Diagrama de Secuencia para Desvincular un Alumno de un Grupo

Se elige el ID del grupo, el ID del alumno y se envían los datos al sistema, se validan y si los datos son válidos, las modificaciones se llevan a cabo después se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

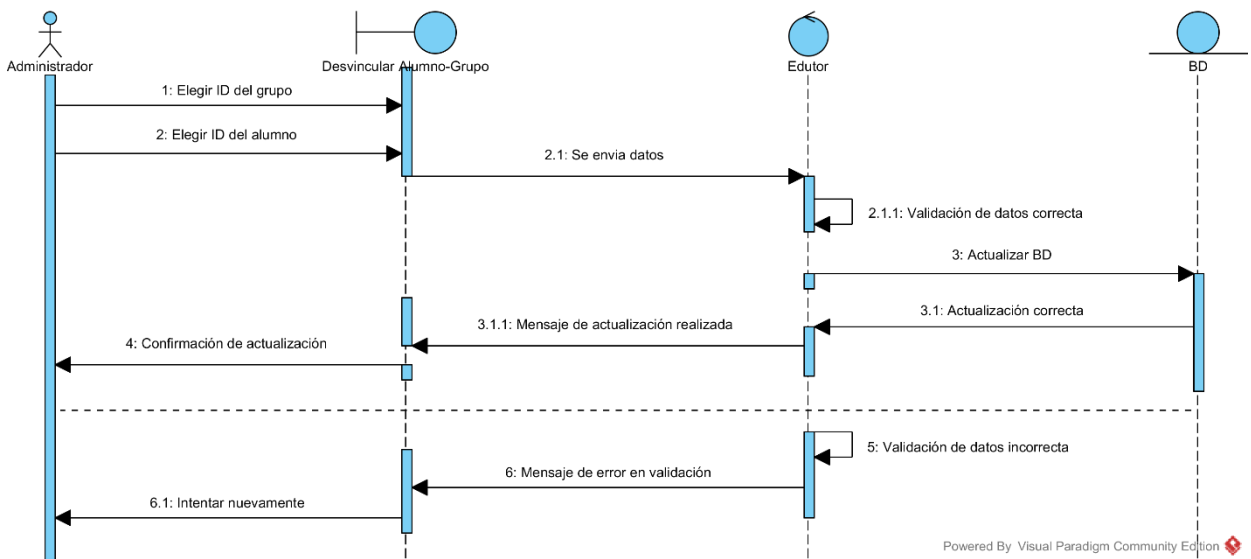


Ilustración 51 Diagrama de secuencia para Desvincular un Alumno de un Grupo

Diagrama de Secuencia para Generar Encuesta

El usuario elige crear una nueva encuesta, ingresa la pregunta a realizar, se envía al sistema. Se validan los datos de la pregunta y en caso de ser válidos, se da de alta. Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

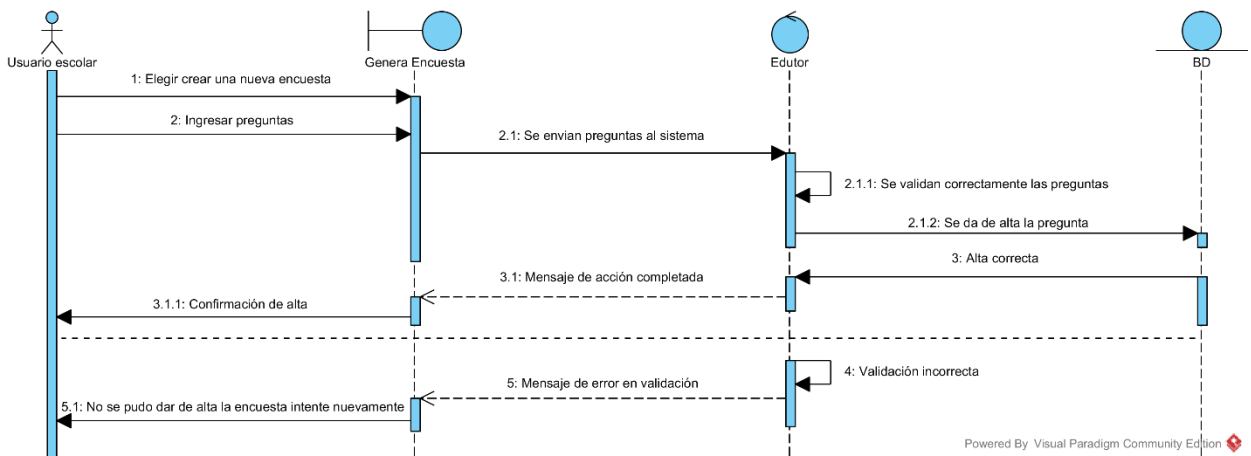


Ilustración 52 Diagrama de secuencia para Generar Encuesta

Diagrama de Secuencia para Asignar Duración a una Encuesta

El usuario ha añadido una pregunta. Se elige si desea asignar una duración o dejar la predefinida por el sistema. Se introducen los datos y se envían al sistema. El sistema valida los datos. Si son

válidos modifica la duración de la pregunta. Se envía un mensaje de respuesta de acuerdo a lo sucedido.

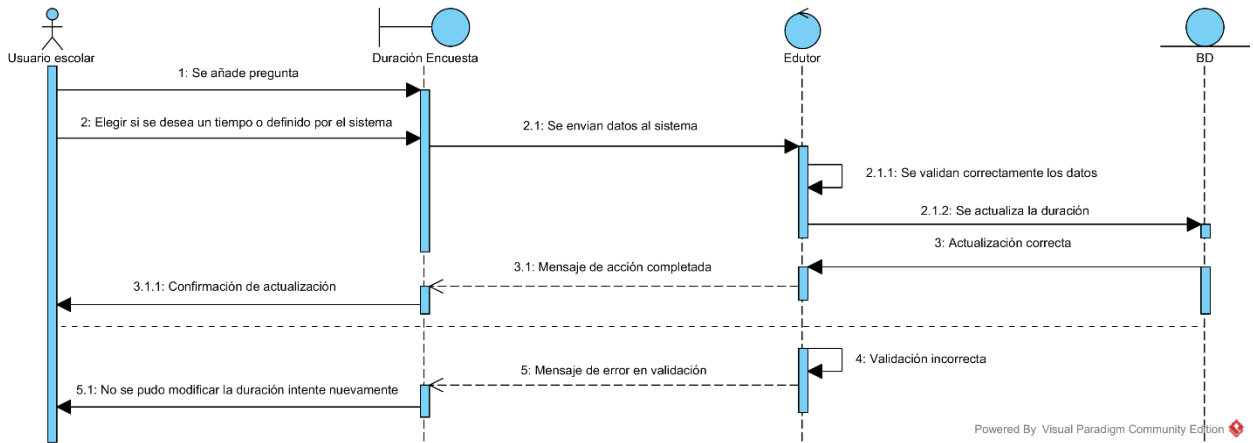


Ilustración 53 Diagrama de secuencia para Asignar Duración a una Encuesta

Diagrama de Secuencia para Añadir Respuesta

El usuario ha añadido una pregunta. Se introducen las posibles opciones de respuesta, una por una. El sistema valida cada respuesta posible. Si son válidas, se asigna la posible respuesta a la pregunta. Se envía un mensaje de acuerdo a lo sucedido.

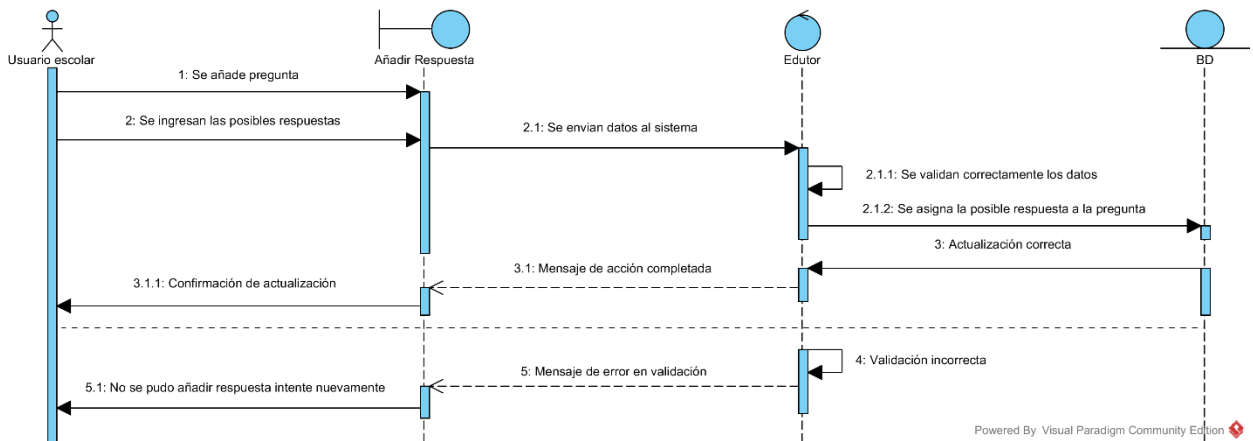


Ilustración 54 Diagrama de secuencia para Añadir Respuesta

Diagrama de Secuencia para Elegir un Grupo

El usuario ha añadido una pregunta. Se elige uno de los grupos que el profesor tiene asignados. Se envía un mensaje de acuerdo a lo sucedido.

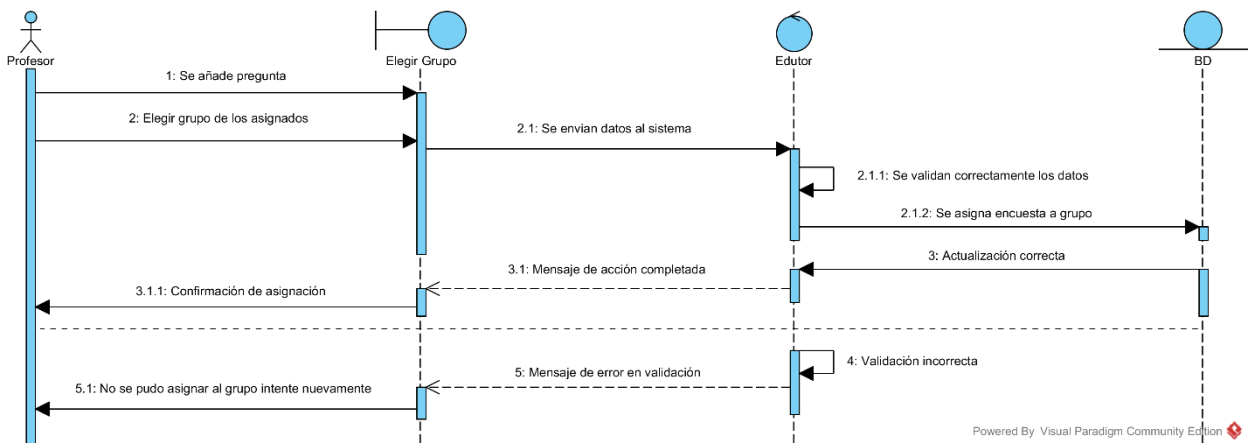


Ilustración 55 Diagrama de secuencia para Elegir un Grupo

Diagrama de Secuencia para Consultar Resultados

El usuario consulta sus preguntas válidas. Elige la que desea revisar. El sistema envía los resultados para que el tutor lo revise.

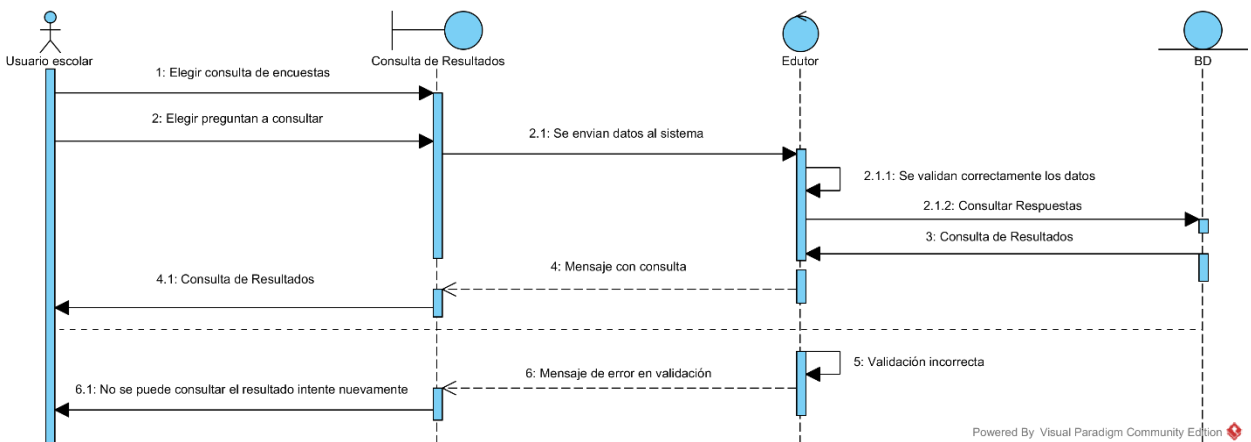


Ilustración 56 Diagrama de secuencia para Consultar Resultados

Diagrama de Secuencia para Contestar Encuesta

El tutor elige de entre todas las preguntas válidas que uno o varios profesores han registrado para él. El sistema entrega las posibles respuestas. El tutor elige una y el sistema registra su respuesta.

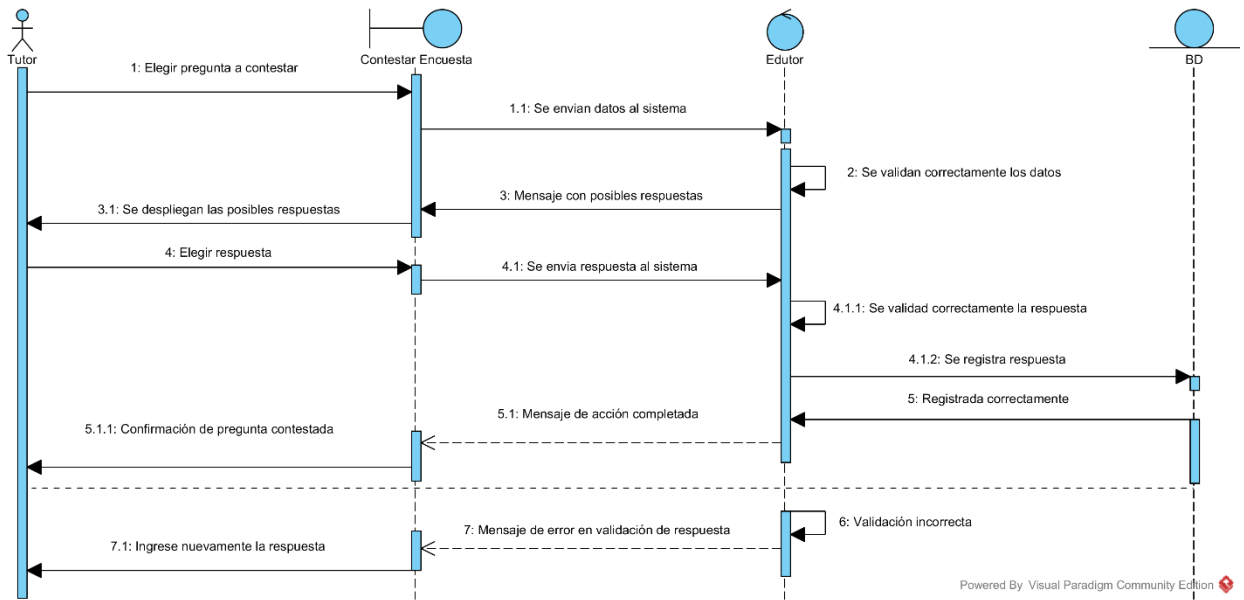


Ilustración 57 Diagrama de secuencia para Responder una Pregunta

Diagrama de Secuencia para Crear un Evento

El usuario elige crear un nuevo evento. Ingresar en título del evento y una descripción. Se envía al sistema. Se validan los datos del evento y en caso de ser válidos, se da de alta. Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

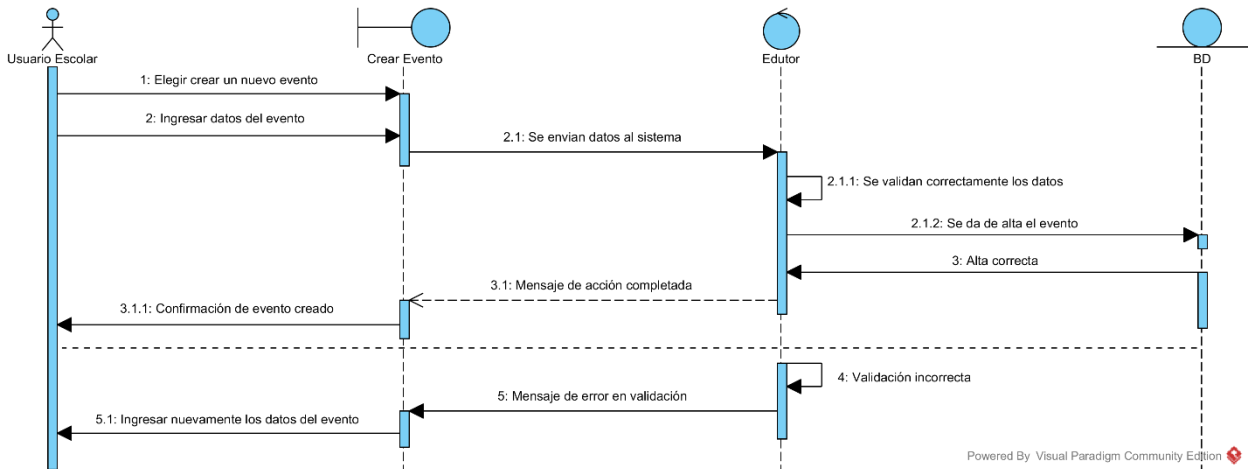


Ilustración 58 Diagrama de secuencia para Crear un Evento

Diagrama de Secuencia para Asignar Fecha a Evento

El usuario ha añadido un evento. Se introducen los datos y se envían al sistema. El sistema valida los datos. Si son válidos asigna la fecha de realización del evento. Se envía un mensaje de respuesta de acuerdo a lo sucedido.

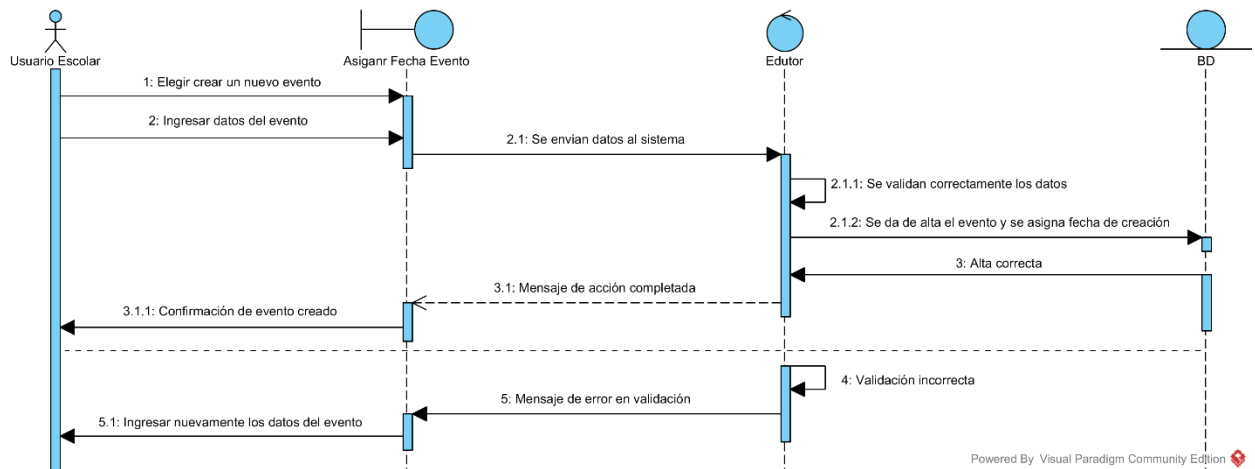


Ilustración 59 Diagrama de secuencia para Asignar Fecha a Evento

Diagrama de Secuencia para Elegir Tipo de Evento

El usuario ha añadido un evento. Se elige entre los dos tipos de evento. Se envía un mensaje de respuesta de acuerdo a lo sucedido.

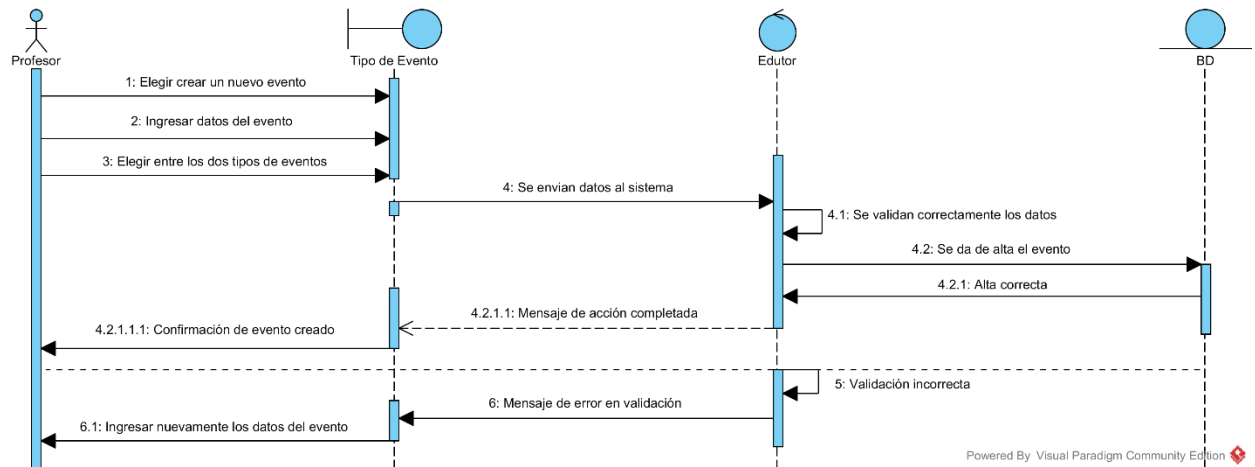


Ilustración 60 Diagrama de secuencia para Elegir Tipo de Evento

Diagrama de Secuencia para Elegir Invitados al Evento

El usuario ha añadido un evento. Según el tipo de evento el sistema entrega una lista de posibles invitados a él. Se envían los invitados. El sistema los registra y envía las notificaciones.

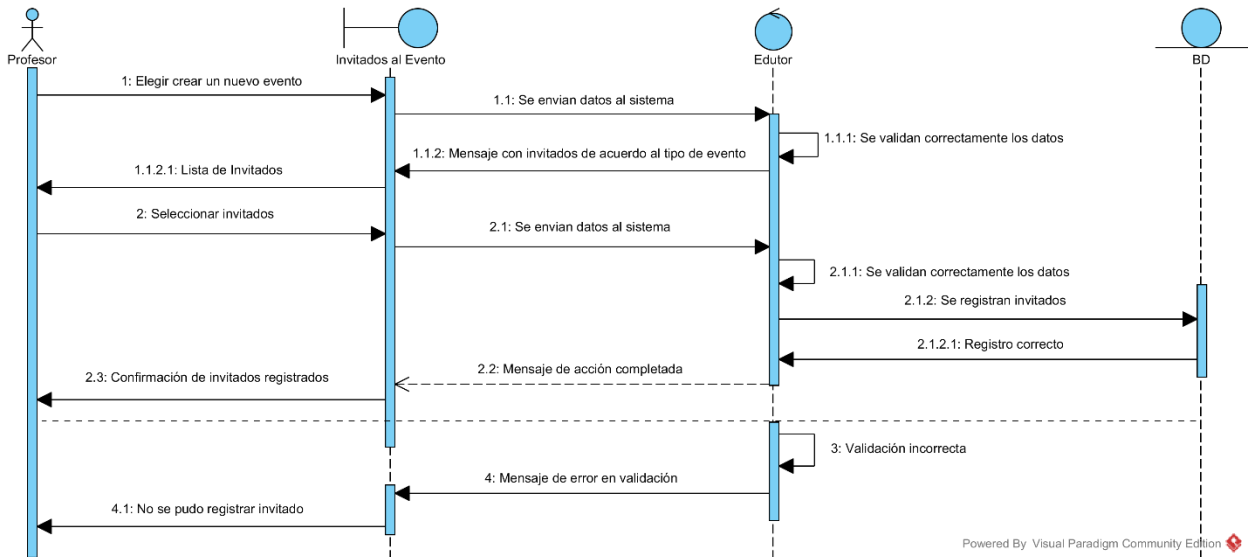


Ilustración 61 Diagrama de secuencia para Elegir Invitados al Evento

Diagrama de Secuencia para Responder Invitación a Evento

El tutor elige uno de entre todos los eventos a los que ha sido invitado. El tutor envía una confirmación sobre su participación en el evento. El sistema registra su respuesta.

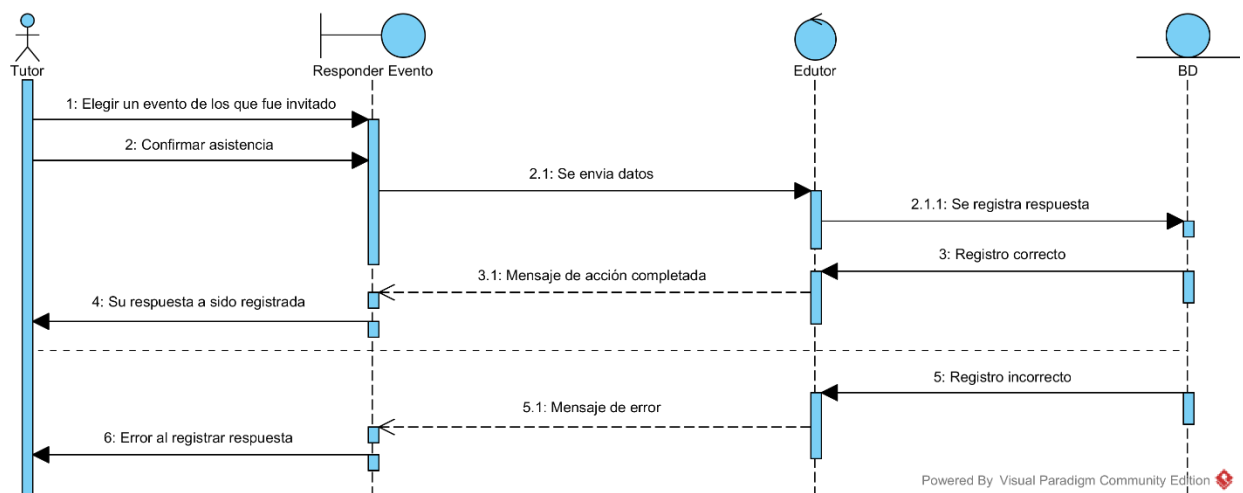


Ilustración 62 Diagrama de secuencia para Responder Invitación a Evento

Diagrama de Secuencia para Enviar Mensaje a un Tutor

El usuario ingresa al sistema. Selecciona al profesor/tutor con el que dese establecer comunicación. Envía mensaje.

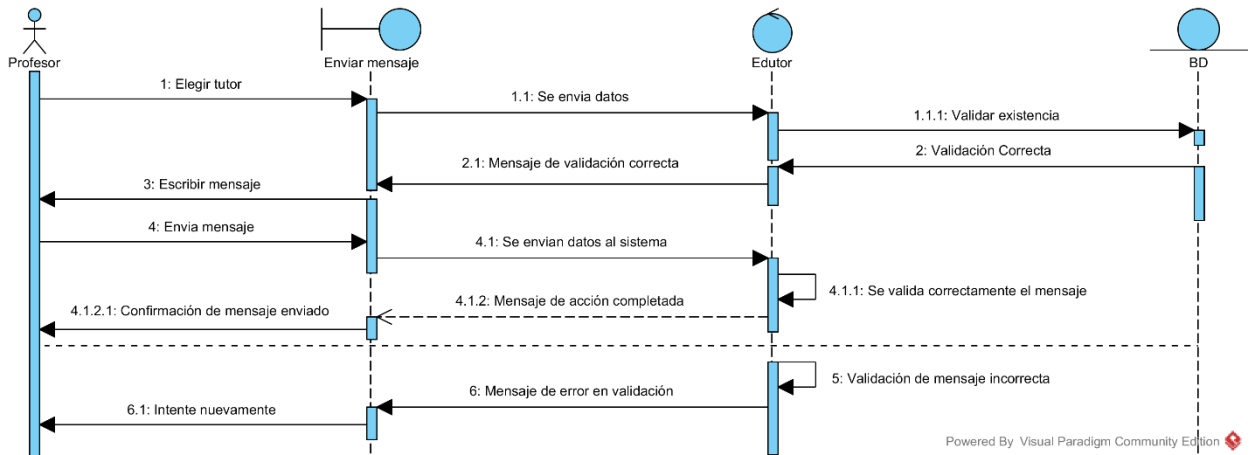


Ilustración 63 Diagrama de secuencia para Enviar mensaje a Tutor

Diagrama de Secuencia para Enviar Mensaje a un Profesor

El usuario ingresa al sistema. Selecciona al profesor/tutor con el que dese establecer comunicación. Envía mensaje.

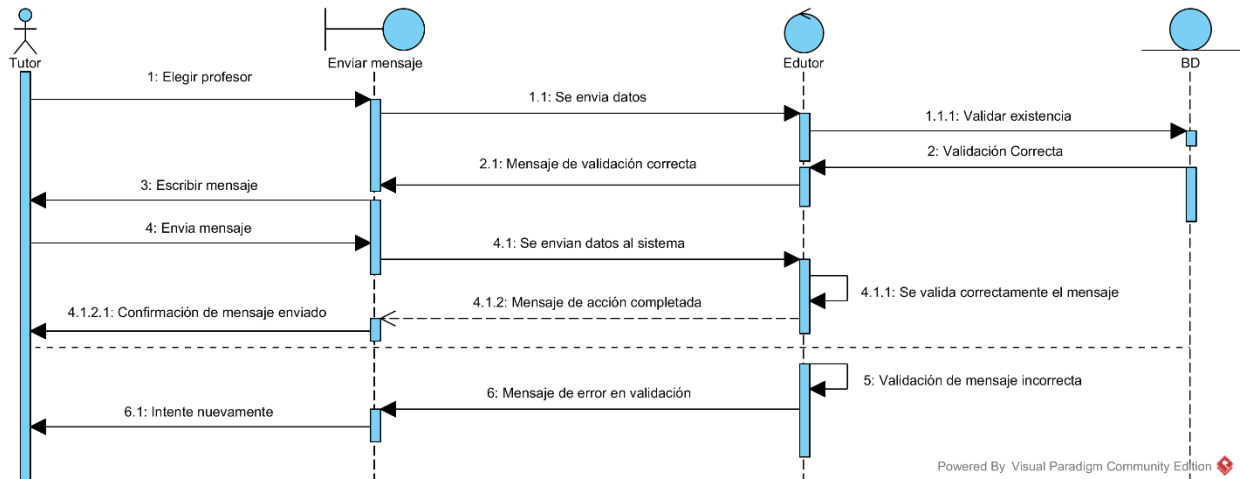


Ilustración 64 Diagrama de secuencia para Enviar mensaje a un Profesor

Diagrama de Secuencia para Elegir un Tutor Receptor

El usuario ingresa al sistema. Si es el caso del profesor selecciona el grupo, el nombre del alumno y establecerá comunicación con el tutor. Si es el caso del tutor se enlistaran únicamente los profesores que den clase a su hijo, selecciona al profesor con el que dese establecer comunicación.

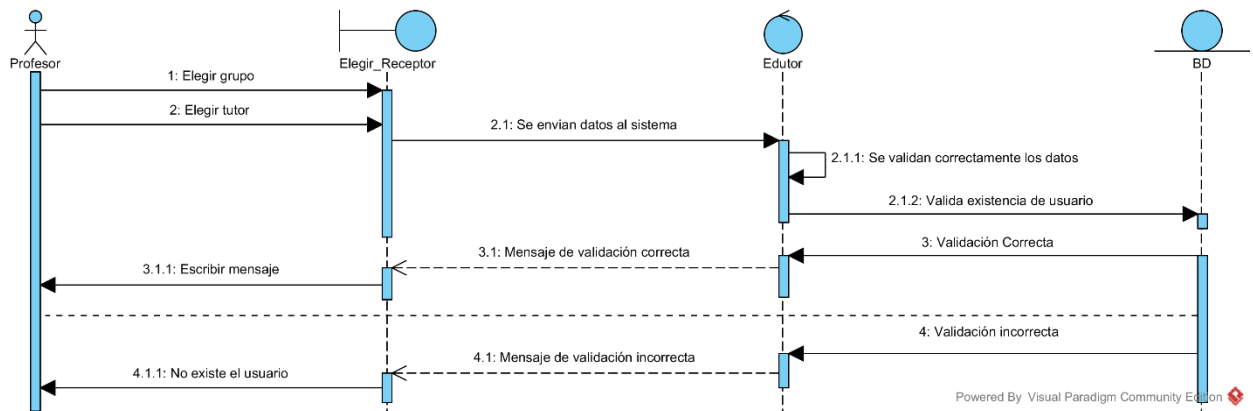


Ilustración 65 Diagrama de secuencia para elegir a un tutor receptor

Diagrama de Secuencia para Elegir un Profesor Receptor

El usuario ingresa al sistema. Si es el caso del profesor selecciona el grupo, el nombre del alumno y establecerá comunicación con el tutor. Si es el caso del tutor se enlistaran únicamente los profesores que den clase a su hijo, selecciona al profesor con el que dese establecer comunicación.

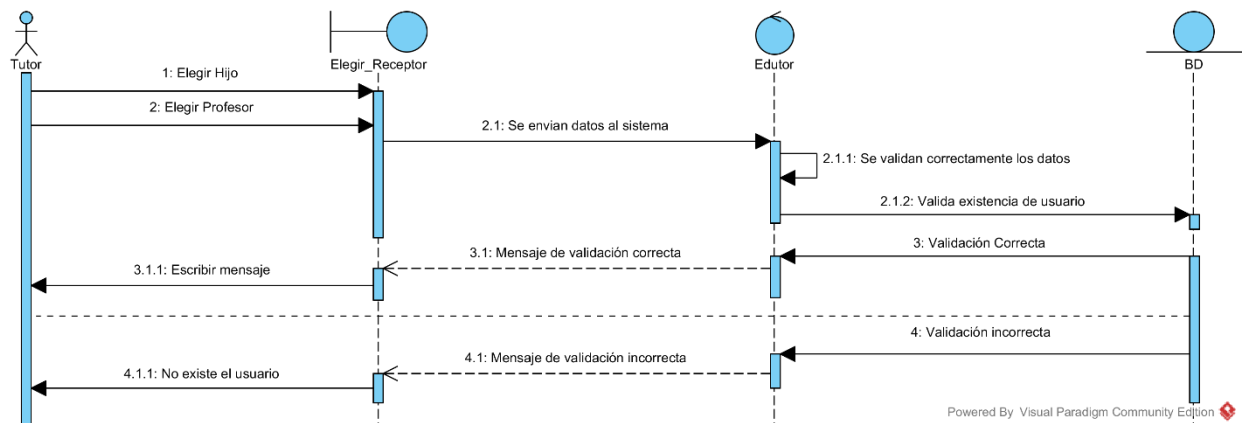


Ilustración 66 Diagrama de secuencia para elegir a un profesor receptor

Diagrama de Secuencia para Establecer comunicación

El usuario selecciona un receptor. Envía mensaje. Se establece la comunicación.

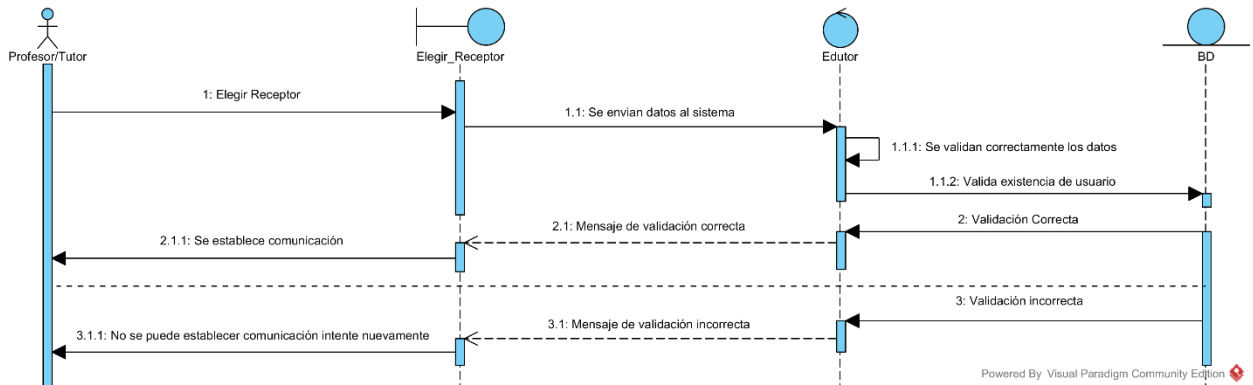


Ilustración 67 Diagrama de secuencia para establecer comunicación

Diagrama de Secuencia para Notificar Mensaje

Se envía mensaje. Se establece comunicación. Notificación de mensaje recibido.

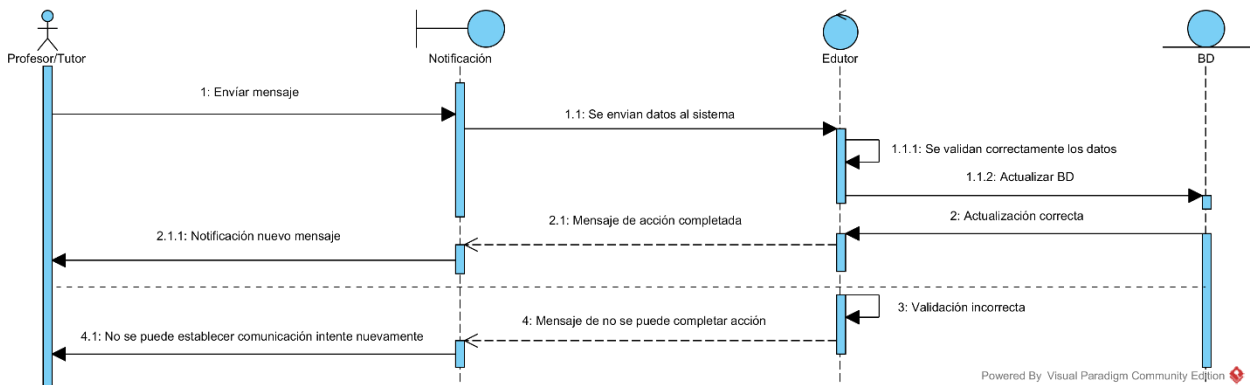


Ilustración 68 Diagrama de secuencia para notificar mensaje

Diagrama de Secuencia para Enviar Notificación como Escuela

Ingreso al sistema, se genera notificación. Envía a toda la comunidad.

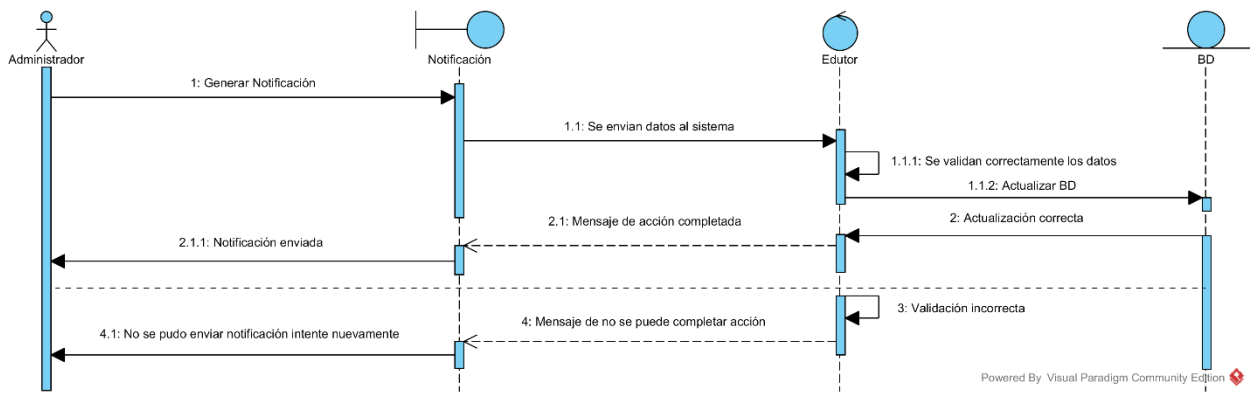


Ilustración 69 diagrama de secuencia para enviar notificación como escuela

Diagrama de Secuencia para Enviar Notificación como Profesor

El profesor ingresa al sistema, selecciona a los tutores a los que dese notificar. Envía notificación.

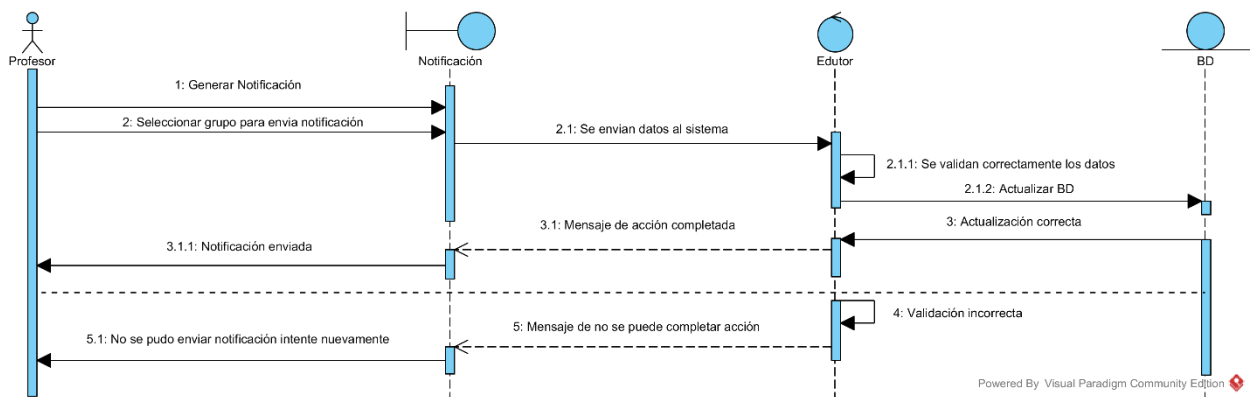


Ilustración 70 Diagrama de secuencia para enviar notificación como profesor

Diagrama de Secuencia para Elegir un Receptor

El profesor ingresa al sistema. Elegirá el grupo en el que se encuentran los tutores a los que desea notificar.

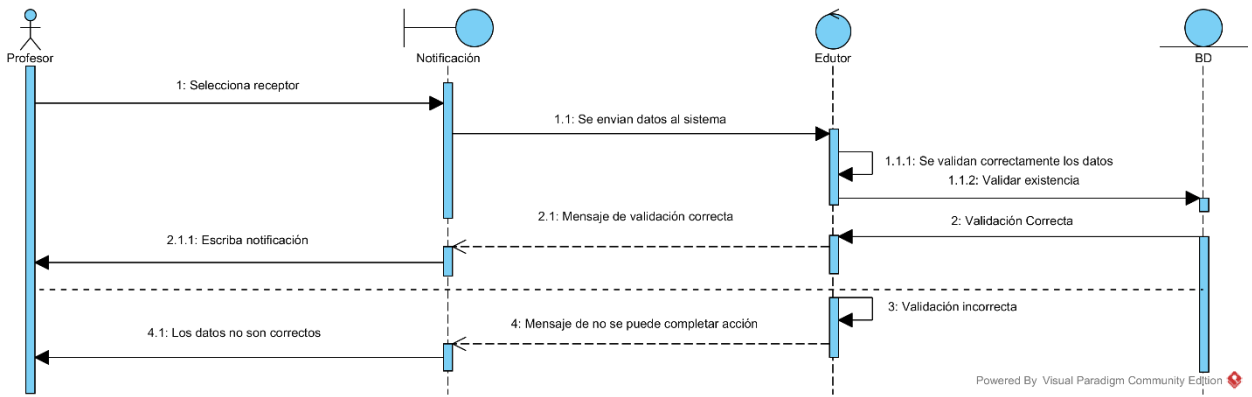


Ilustración 71 Diagrama de secuencia para elegir un receptor

Diagrama de Secuencia para Notificaciones Recibidas

Se genera notificación. Se envía notificación. Nueva notificación recibida.

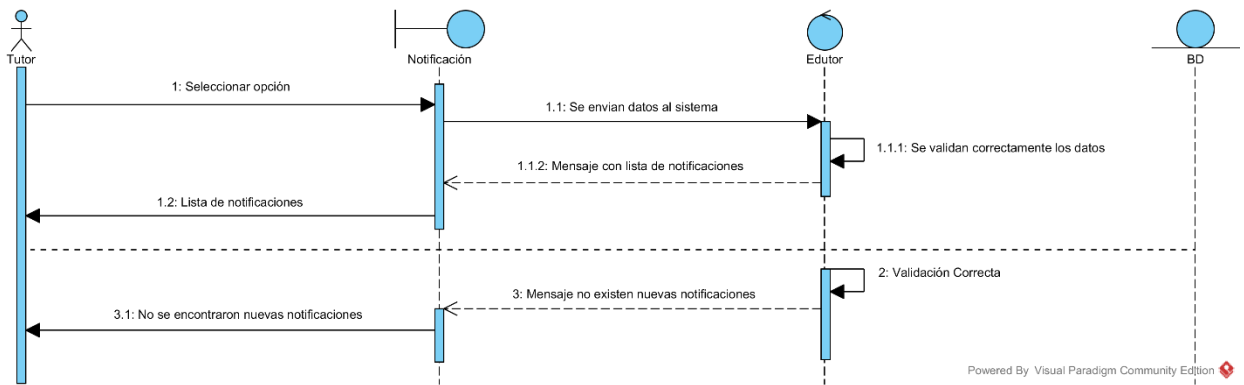


Ilustración 72 Diagrama de secuencia para notificaciones recibidas

Diagramas de Actividades

Diagrama de Actividades para Acceso

Se ingresan los datos que le fueron otorgados, el sistema lo valida y en caso de ser correctos se otorgará acceso al sistema y a las peticiones que desee realizar. En caso contrario se envía un mensaje de error.

Es importante mencionar que este diagrama de actividades se ejecutara cada vez que se realice una petición al sistema.

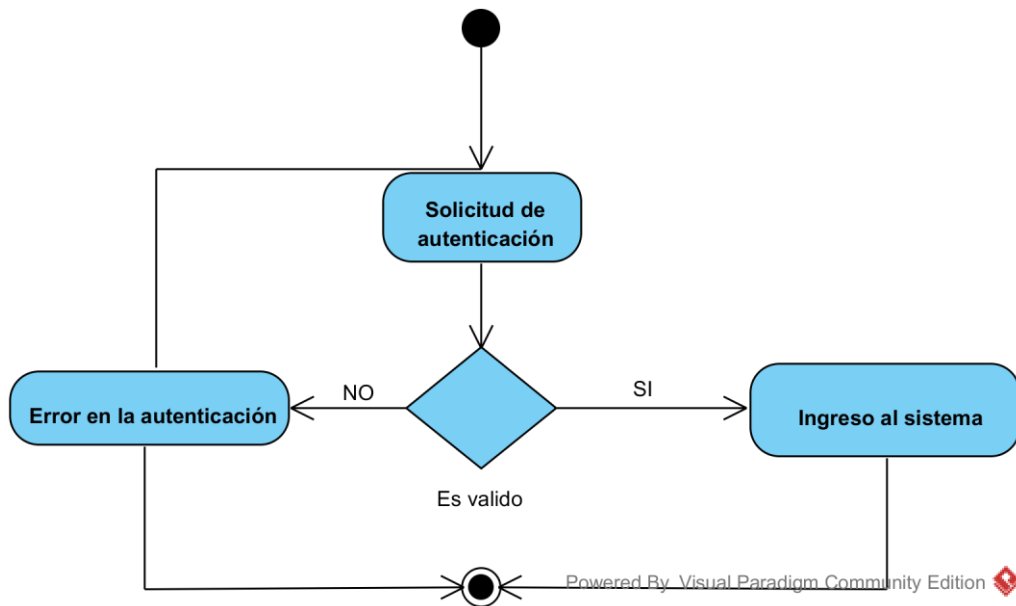


Ilustración 73 Diagrama de actividades para Acceso

Diagrama de Actividades para Alta de alumno

El alta de un alumno será responsabilidad del administrador, quien ingresara los datos del alumno y los enviara al sistema, este los validara y verificara que no existan en la base de datos para proceder a insertarlo y finalmente enviar un mensaje al usuario confirmando el alta del alumno. Si la validación de los datos no es correcta se envía un mensaje de error.

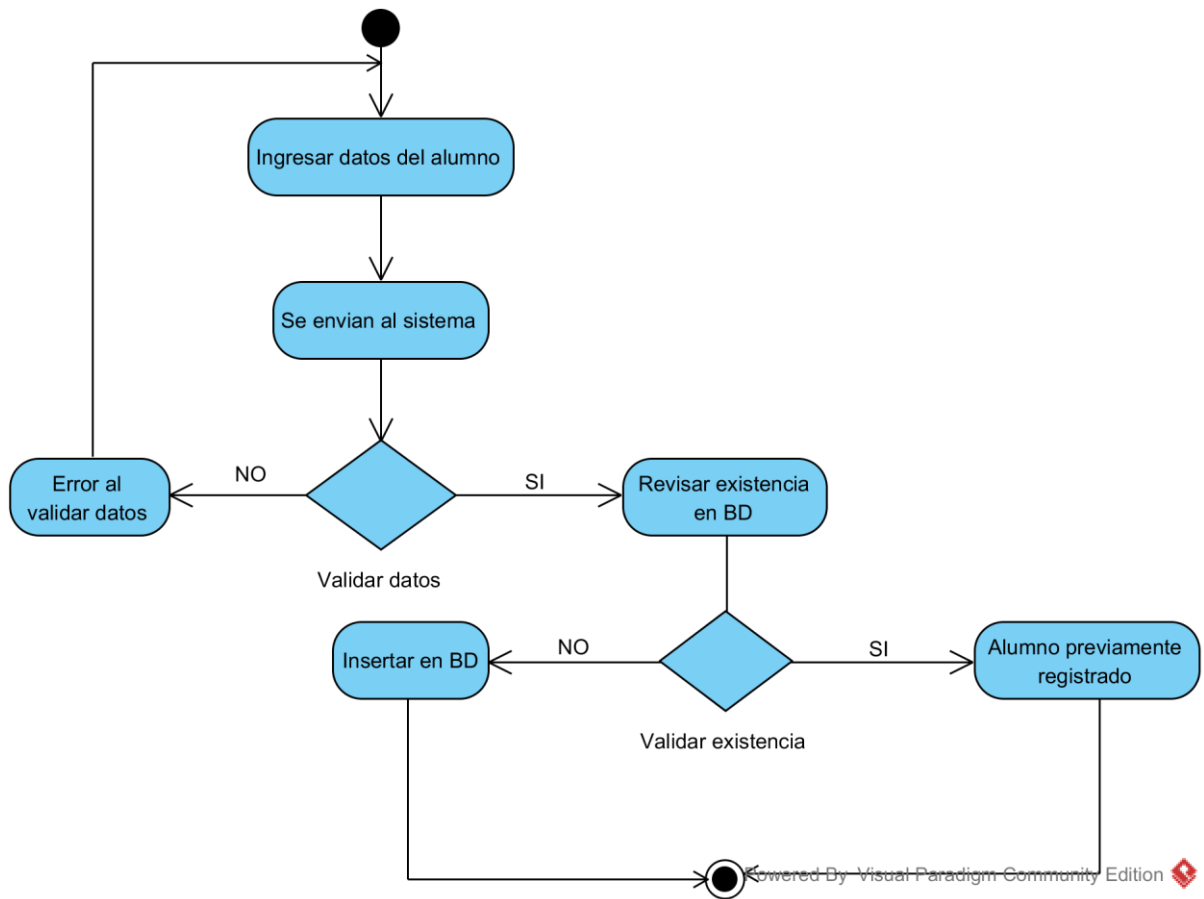


Ilustración 74 Diagrama de actividades para alta de alumno

Diagrama de Actividades para Modificar alumno

El administrador indicara al sistema el alumno que desea modificar, el sistema regresa los datos modificables y el usuario los modifica, se envían al sistema y se validan, si la validación es correcta se actualizan en la base de datos y se envía un mensaje al usuario confirmando la actualización. En caso de que la validación no sea favorable se envía un mensaje de error.

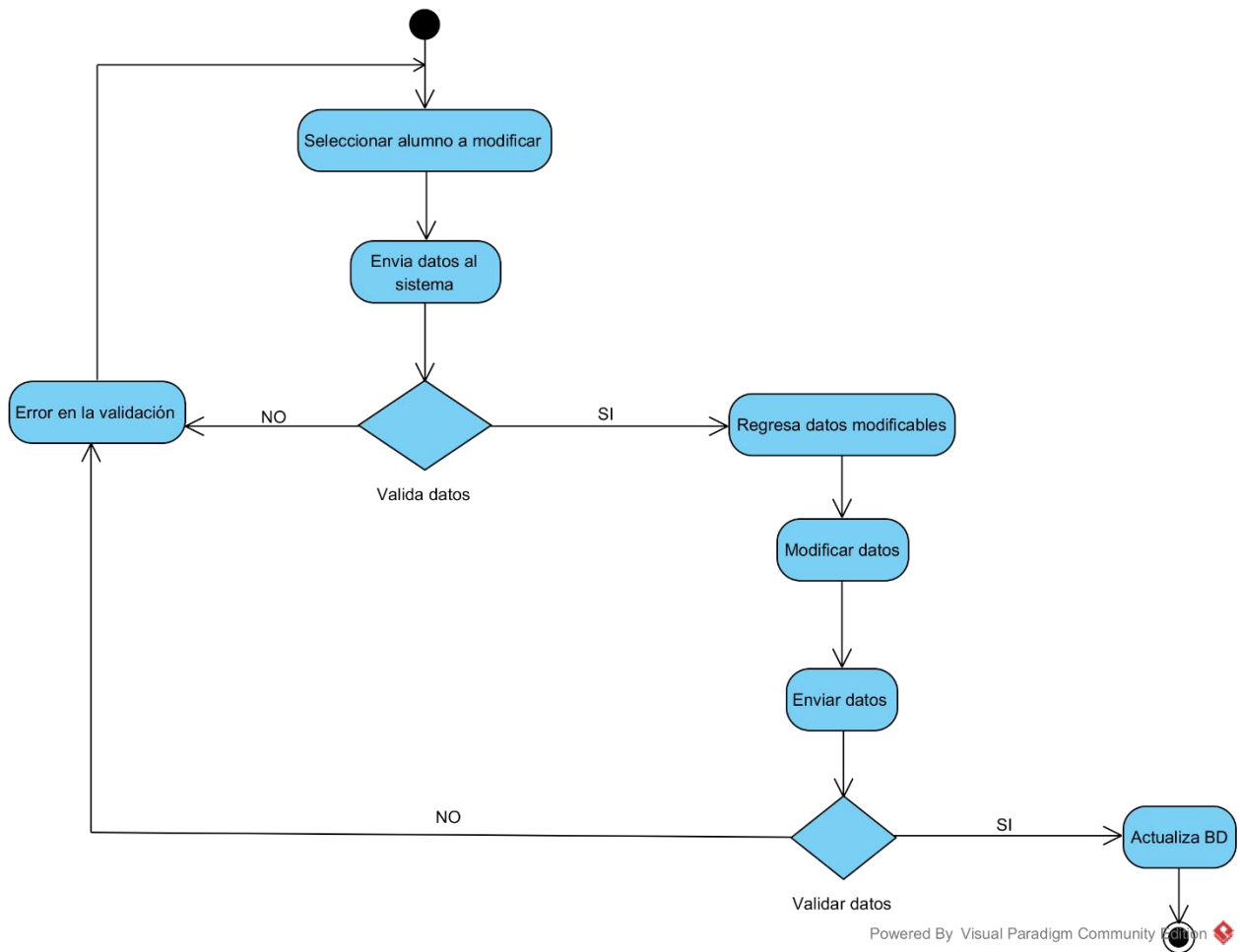
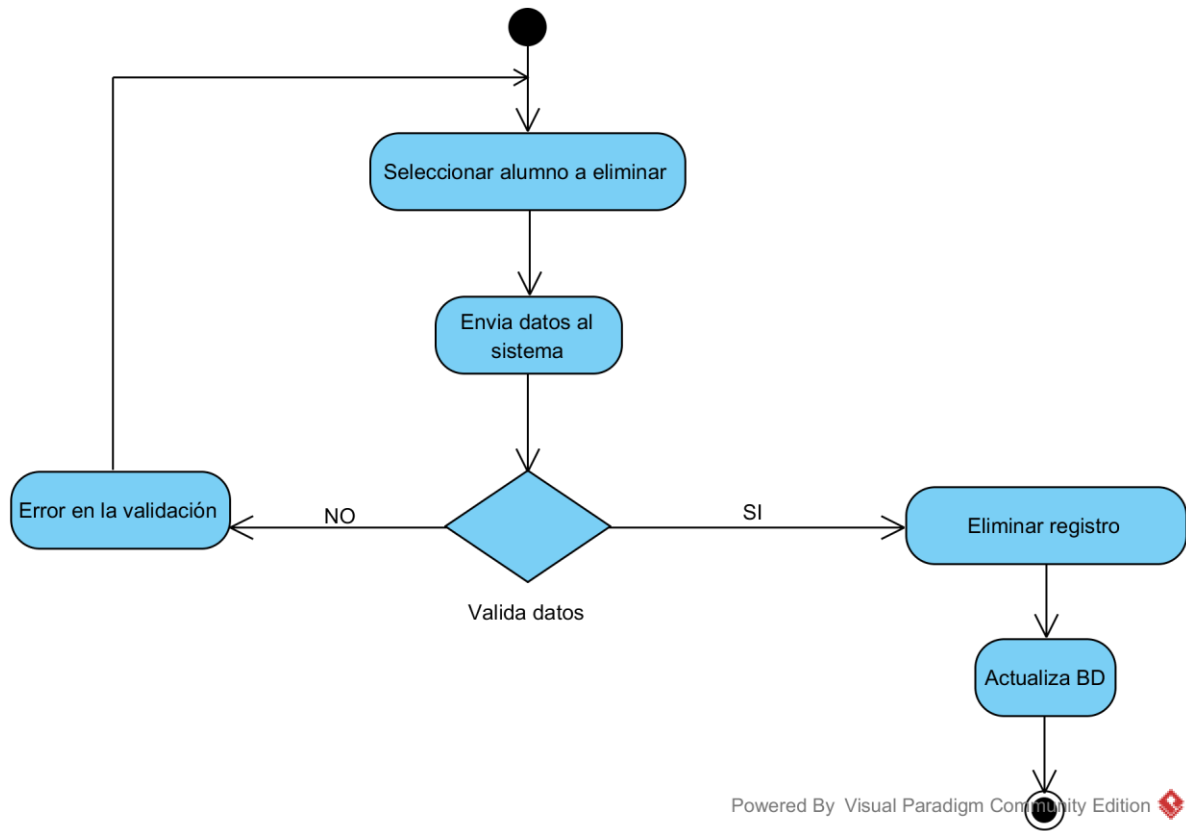


Ilustración 75 Diagrama de actividades para modificar alumno

Diagrama de Actividades para Eliminar alumno

Se indica al sistema el alumno que se desea eliminar, valida la posibilidad de eliminar al alumno, si es válido eliminar un alumno, se elimina y se envía mensaje de respuesta al usuario dependiendo de lo que haya sucedido.



Powered By Visual Paradigm Community Edition

Ilustración 76 Diagrama de actividades para eliminar alumno

Diagrama de Actividades para Alta de profesor

Se ingresan los datos del profesor, se envían al sistema este lo valida y si son válidos y no existe en el sistema, se da de alta. Finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

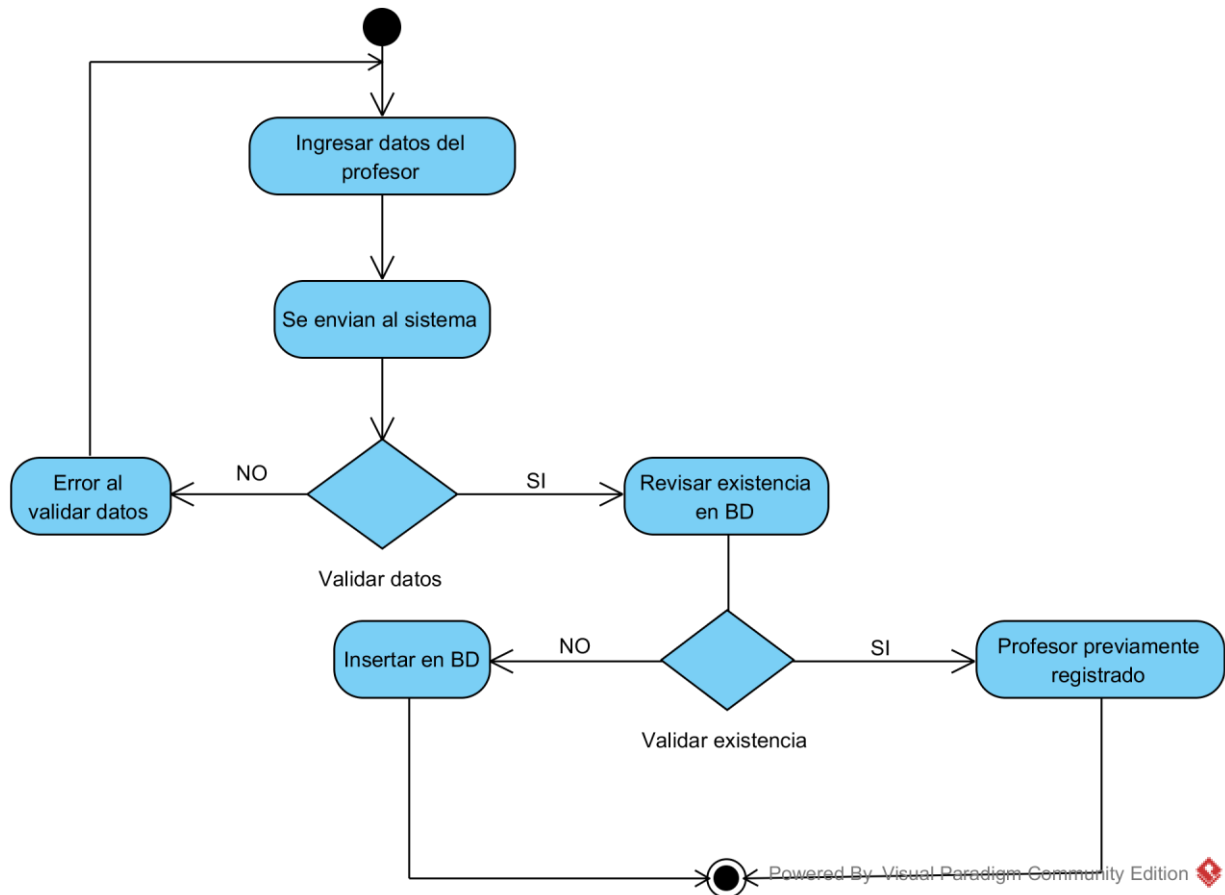


Ilustración 77 Diagrama de actividades para alta de profesor

Diagrama de Actividades para Modificar profesor

Se indica al sistema el profesor que se desea modificar, se regresan los datos modificables, el usuario modifica los datos. Se envían los datos al sistema y se validan, si los datos son válidos, las modificaciones se llevan a cabo al final se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

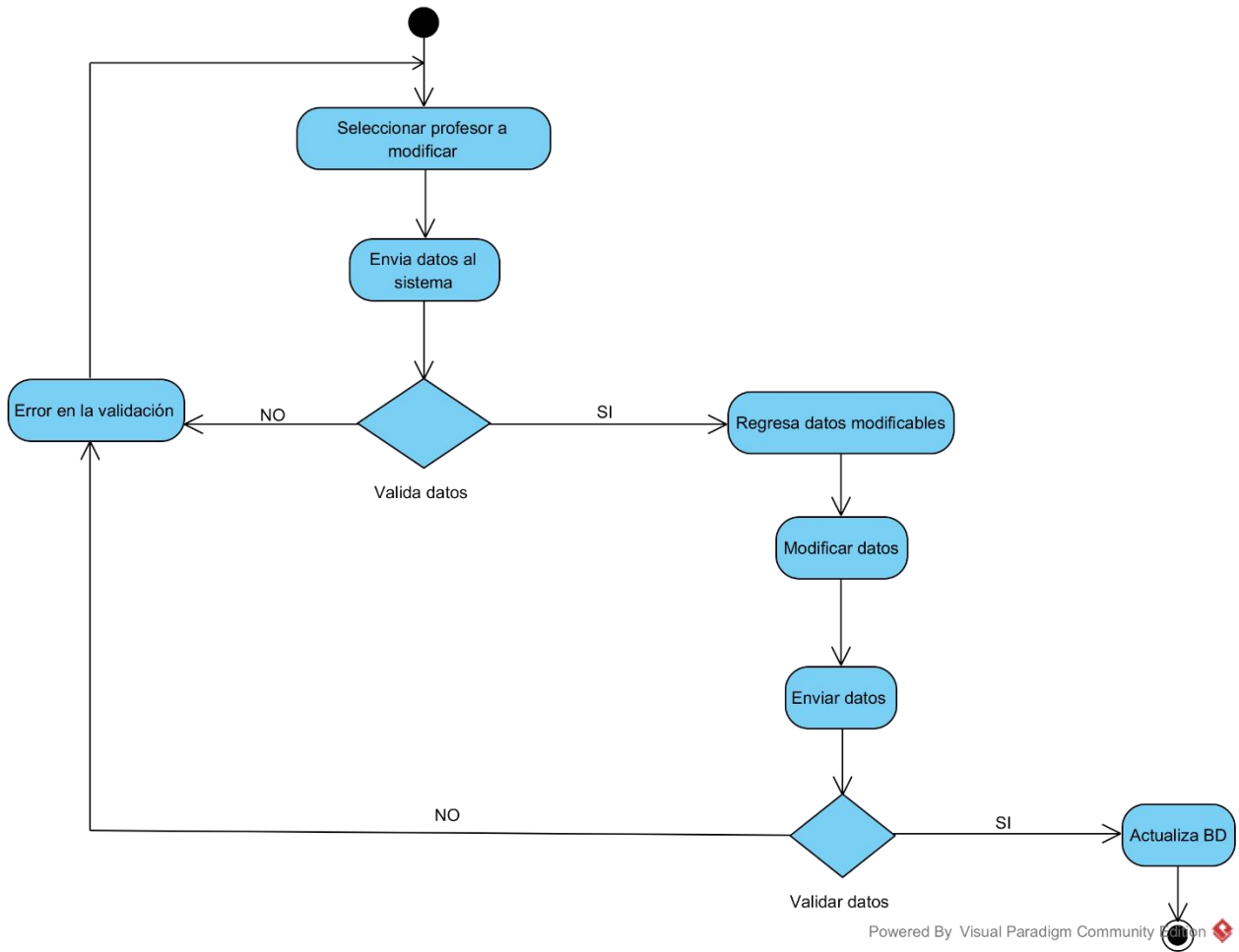
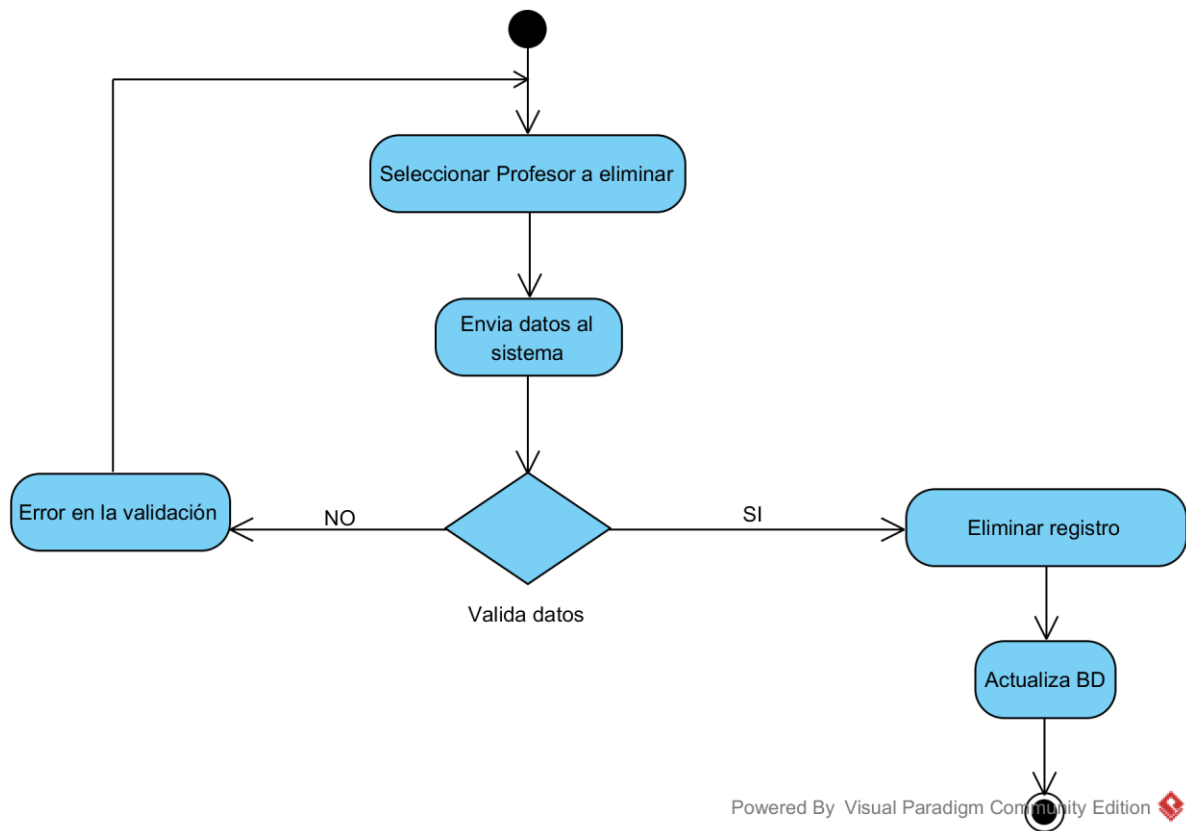


Ilustración 78 Diagrama de actividades para modificar profesor

Diagrama de Actividades para Eliminar profesor

Se indica al sistema el profesor que se desea eliminar, este valida la posibilidad de eliminar al profesor y si es válido eliminar al profesor, se elimina finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.



Powered By Visual Paradigm Community Edition

Ilustración 79 Diagrama de Actividades para Eliminar profesor

Diagrama de Actividades para Añadir un Grupo

Se ingresan los datos del grupo y se envían al sistema este lo valida y si los datos son válidos y no existe en el sistema, se da de alta finalmente envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

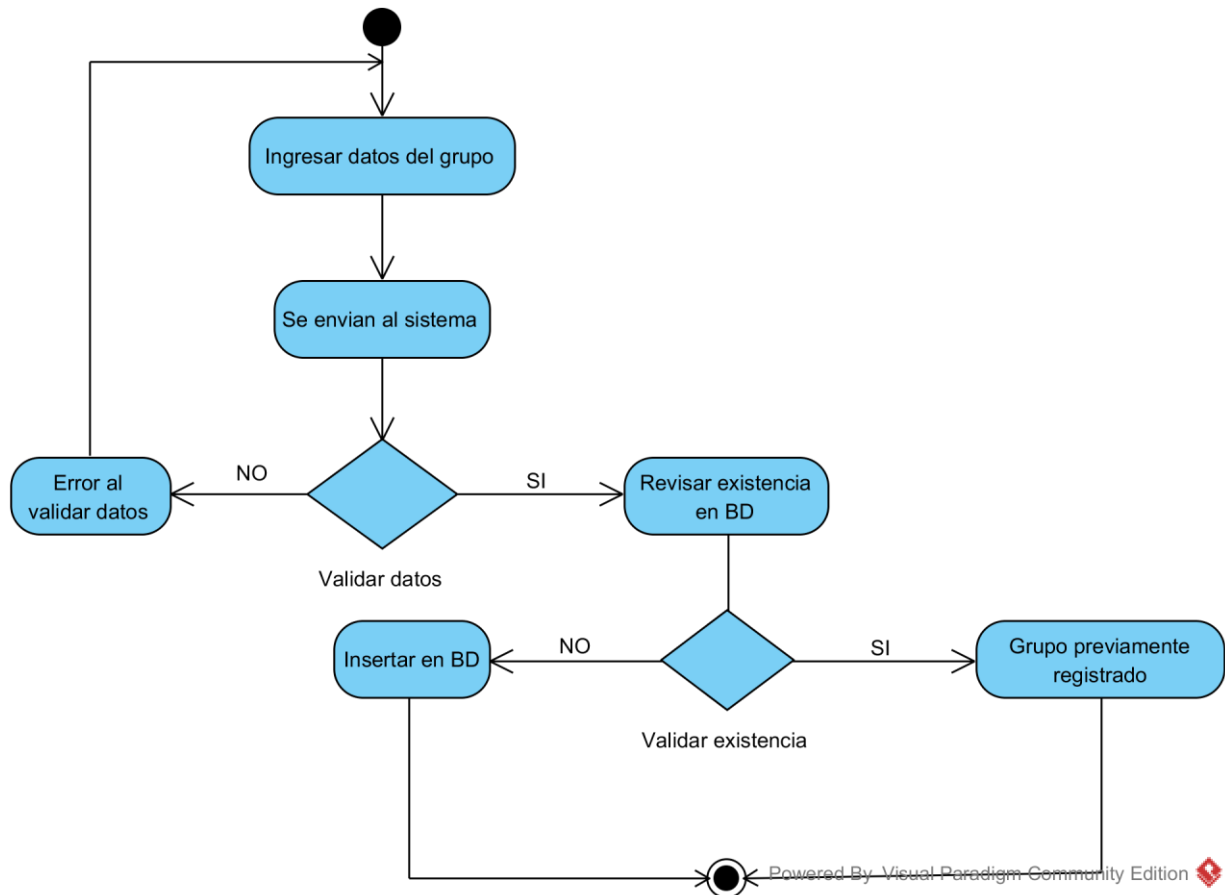


Ilustración 80 Diagrama de Actividades para Añadir un Grupo

Diagrama de Actividades para Modificar un Grupo

Se indica al sistema el grupo que se desea modificar se muestran los datos modificables, el usuario modifica los datos y se envían los datos al sistema este los valida y si los datos son válidos, las modificaciones se llevan a cabo finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

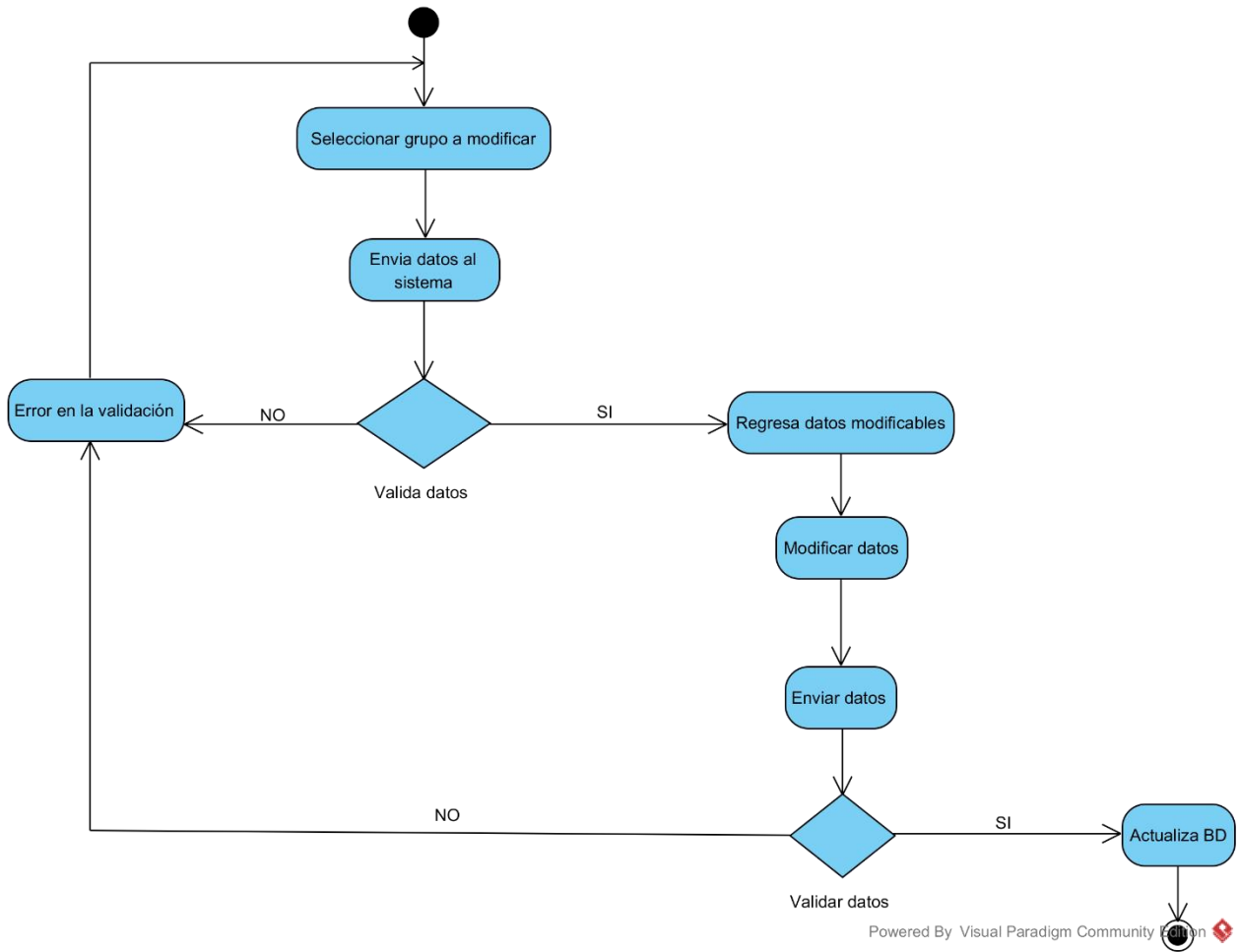


Ilustración 81 Diagrama de Actividades para Modificar un Grupo

Diagrama de Actividades para Eliminar Grupo

Se indica al sistema el grupo que se desea eliminar, valida la posibilidad de eliminar el grupo, si es válido eliminar el grupo, se elimina y finalmente se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

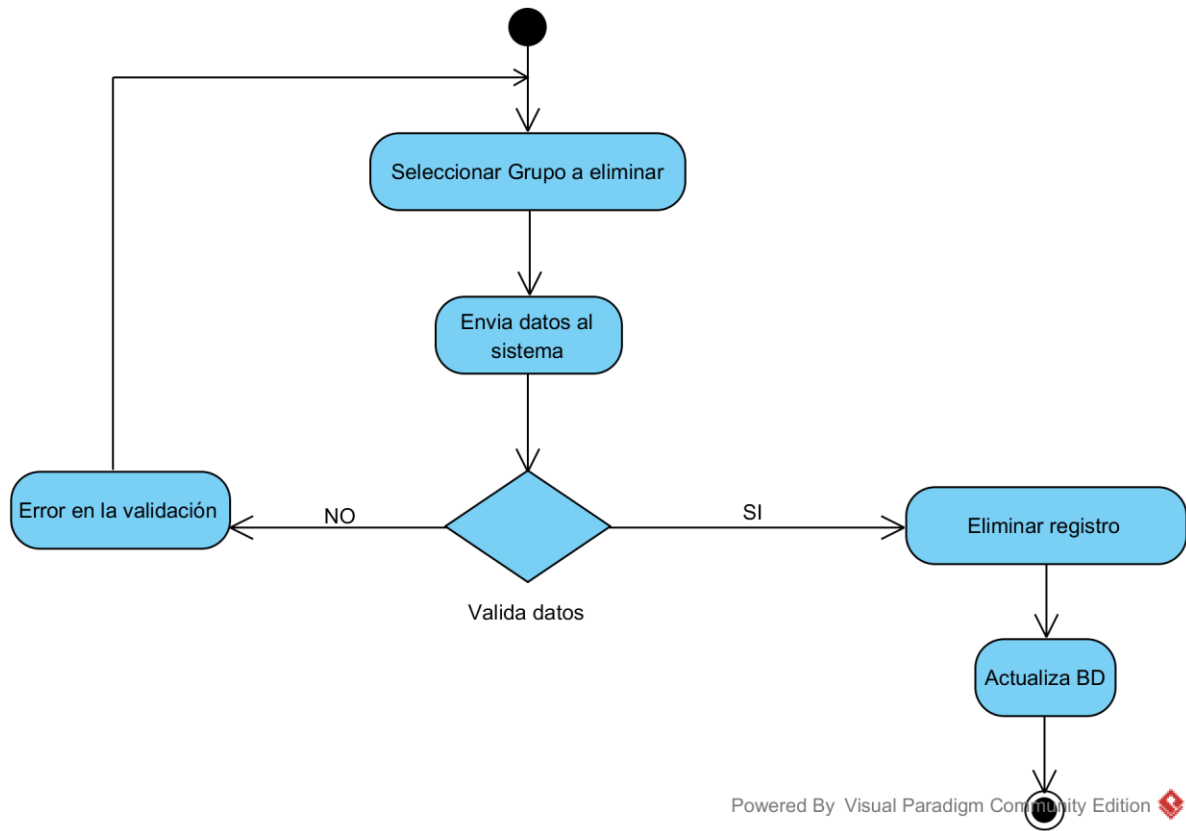
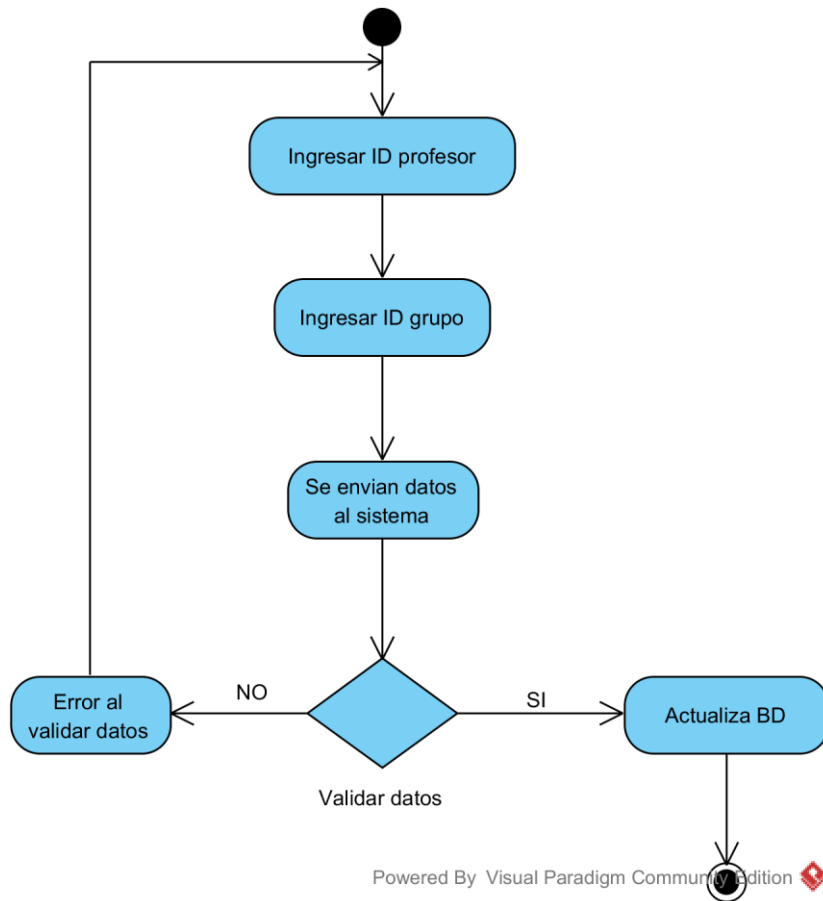


Ilustración 82 Diagrama de Actividades para Eliminar Grupo

Diagrama de Actividades para Vincular un Profesor con un Grupo

Se elige el ID del grupo, el ID del profesor, se envían los datos al sistema y se validan si son válidos, las modificaciones se llevan a cabo y se envía un mensaje de respuesta al usuario dependiendo de lo sucedido.



Powered By Visual Paradigm Community Edition

Ilustración 83 Diagrama de Actividades para Vincular un Profesor con un Grupo

Diagrama de Actividades para Desvincular un Profesor de un Grupo

Se elige el ID del grupo, el ID del profesor y se envían los datos al sistema, se validan y si los datos son válidos, las modificaciones se llevan a cabo después se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

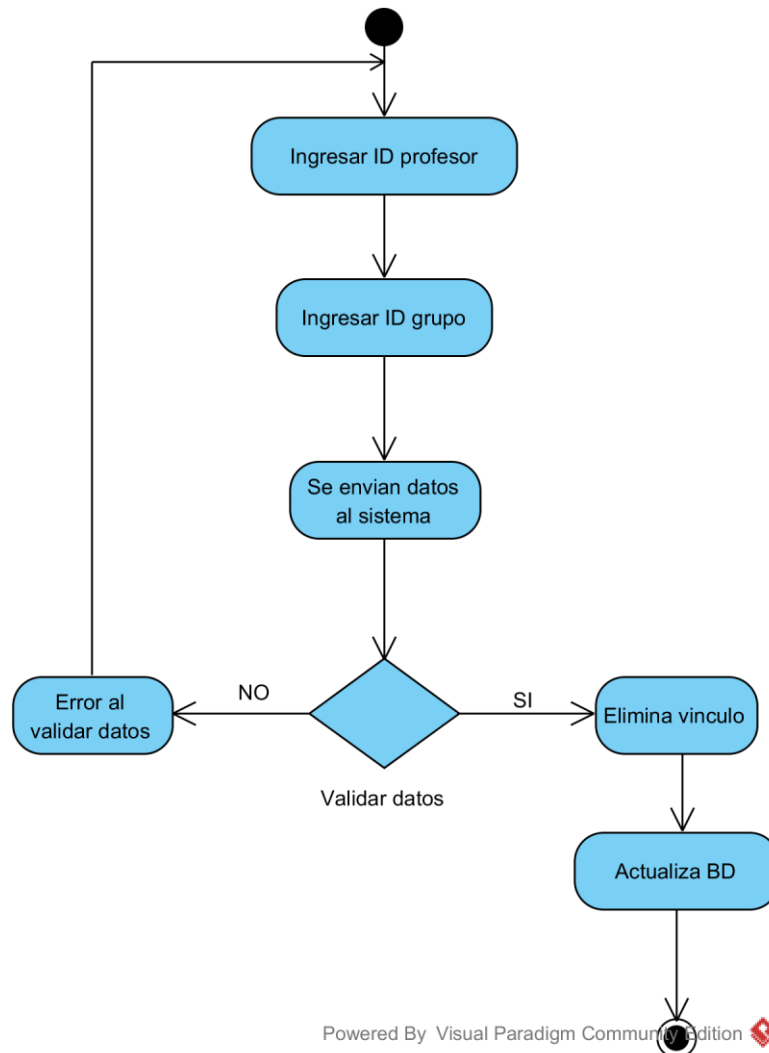
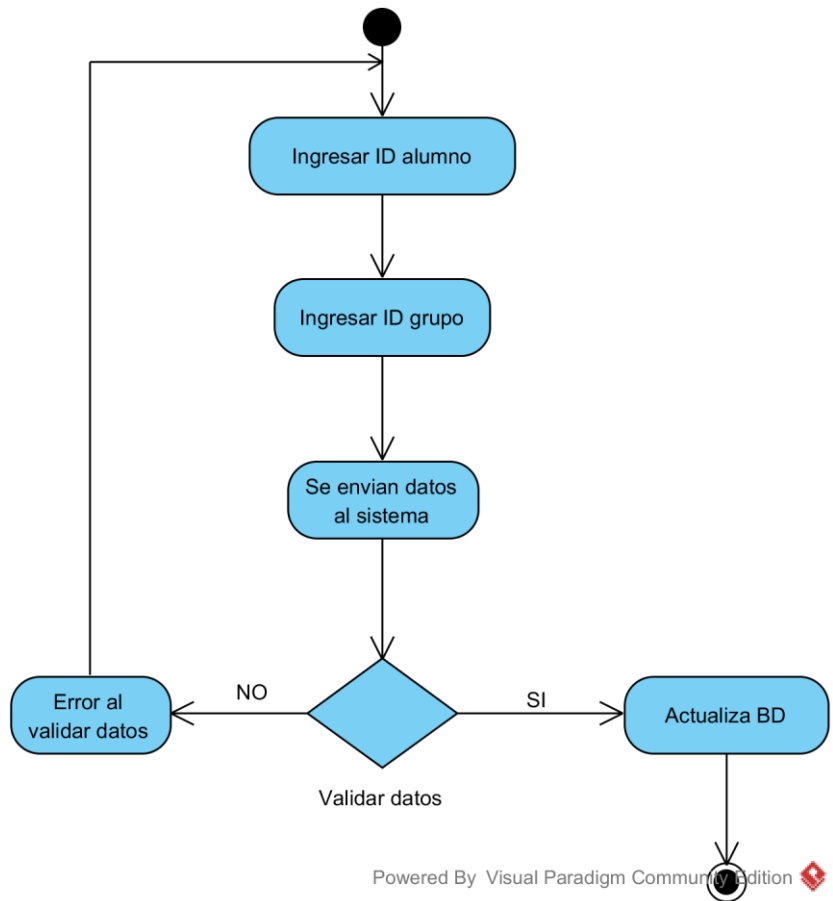


Ilustración 84 Diagrama de Actividades para Desvincular un Profesor de un Grupo

Diagrama de Actividades para Vincular un Alumno a un Grupo

Se elige el ID del grupo, el ID del alumno, se envían los datos al sistema y se validan si son válidos, las modificaciones se llevan a cabo y se envía un mensaje de respuesta al usuario dependiendo de lo sucedido.



Powered By Visual Paradigm Community Edition

Ilustración 85 Diagrama de Actividades para Vincular un Alumno a un Grupo

Diagrama de Actividades para Desvincular un Alumno de un Grupo

Se elige el ID del grupo, el ID del alumno y se envían los datos al sistema, se validan y si los datos son válidos, las modificaciones se llevan a cabo después se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

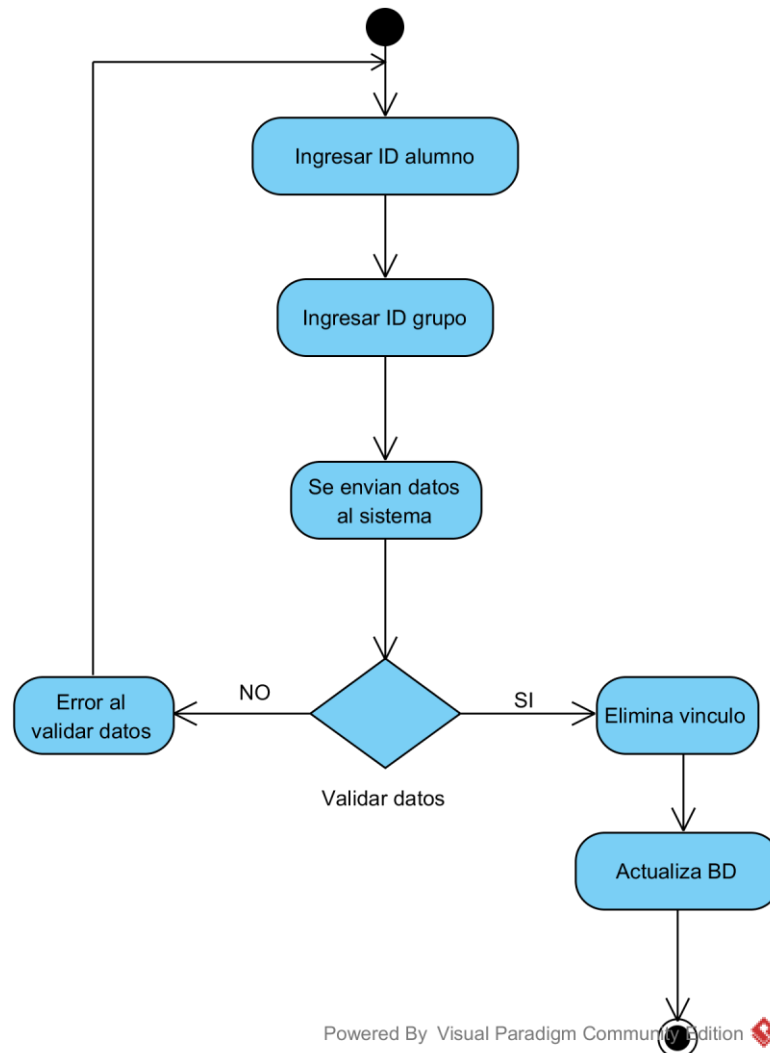


Ilustración 86 Diagrama de Actividades para Desvincular un Alumno de un Grupo

Diagrama de Actividades para Generar Encuestas

El usuario elige crear una nueva encuesta, ingresa la pregunta a realizar, se envía al sistema. Se validan los datos de la pregunta y en caso de ser válidos, se da de alta. Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

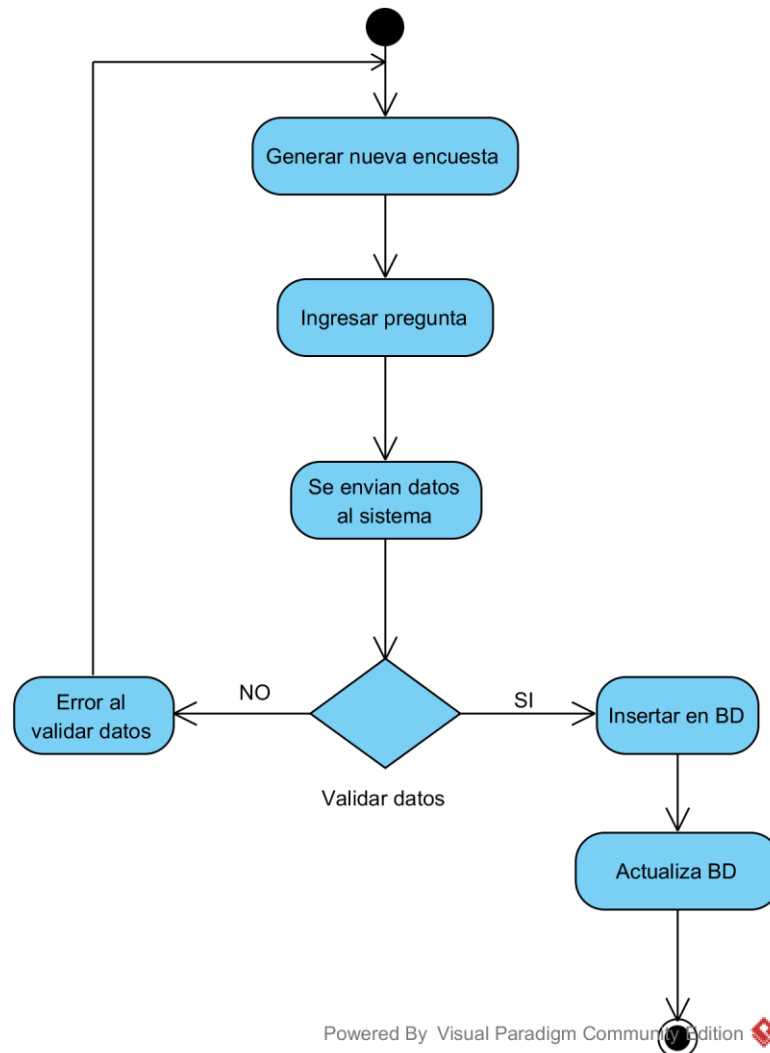


Ilustración 87 Diagrama de Actividades para Generar Encuestas

Diagrama de Actividades para Asignar Duración a Encuesta

El usuario ha añadido una pregunta. Se elige si desea asignar una duración o dejar la predefinida por el sistema. Se introducen los datos y se envían al sistema. El sistema valida los datos. Si son válidos modifica la duración de la pregunta. Se envía un mensaje de respuesta de acuerdo a lo sucedido.

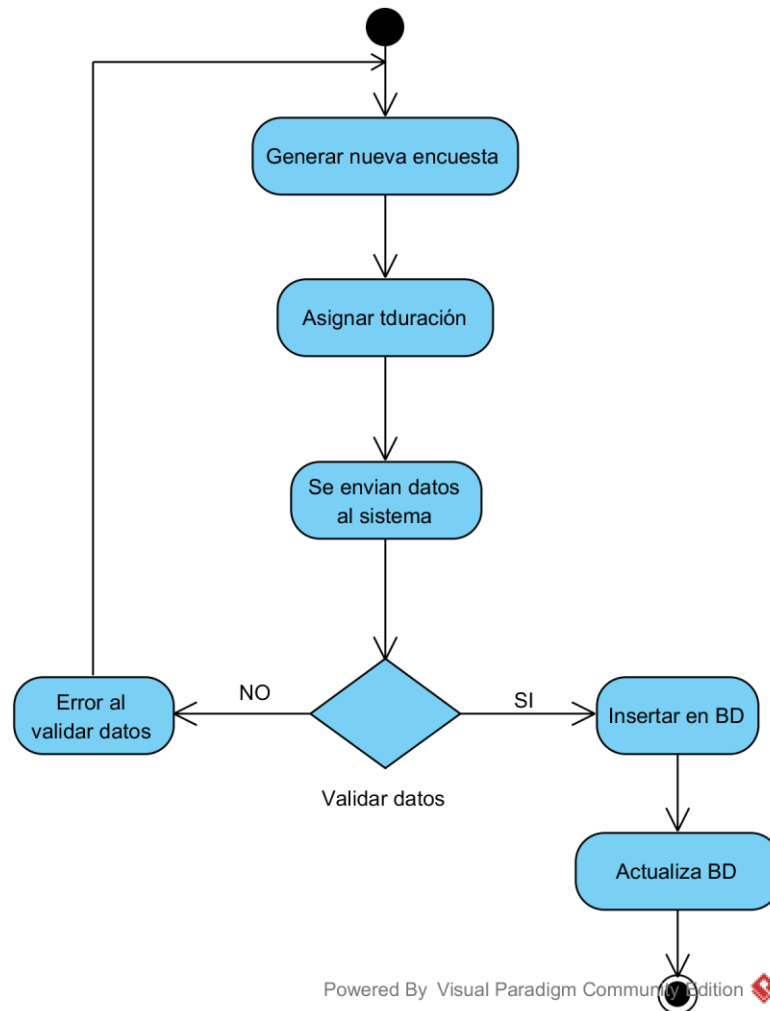


Ilustración 88 Diagrama de Actividades para Asignar Duración a Encuesta

Diagrama de Actividades para Añadir Respuesta

El usuario ha añadido una pregunta. Se introducen las posibles opciones de respuesta, una por una. El sistema valida cada respuesta posible. Si son válidas, se asigna la posible respuesta a la pregunta. Se envía un mensaje de acuerdo a lo sucedido.

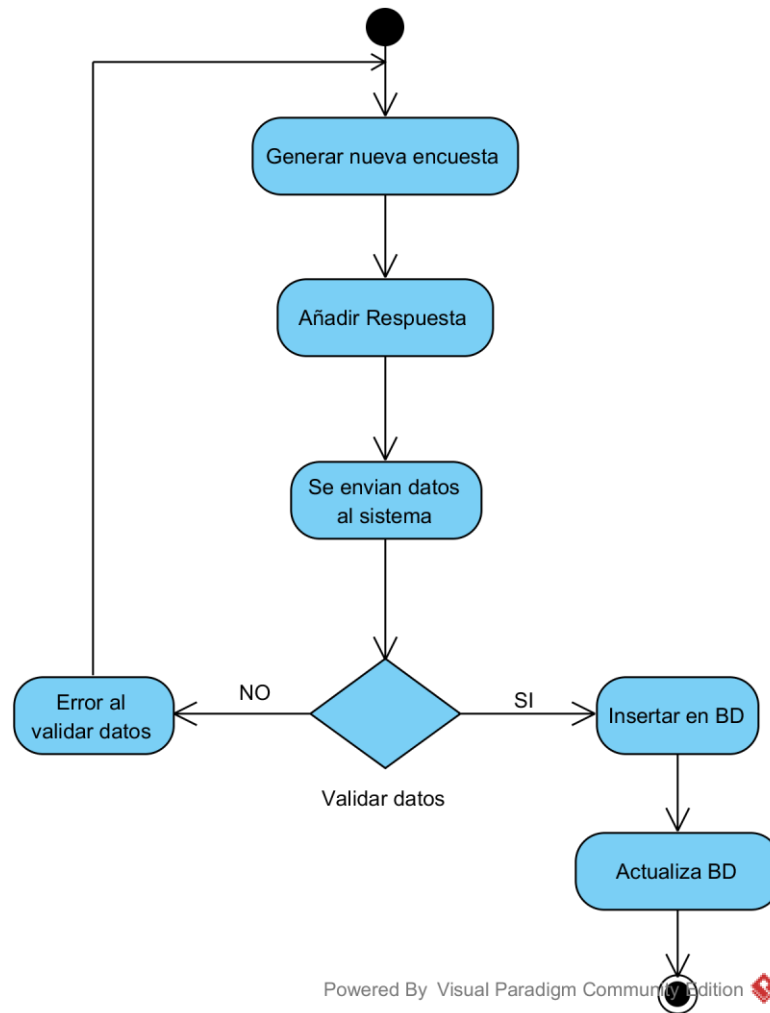


Ilustración 89 Diagrama de Actividades para Añadir Respuesta

Diagrama de Actividades para Elegir un Grupo

El usuario ha añadido una pregunta. Se elige uno de los grupos que el profesor tiene asignados. Se envía un mensaje de acuerdo a lo sucedido.

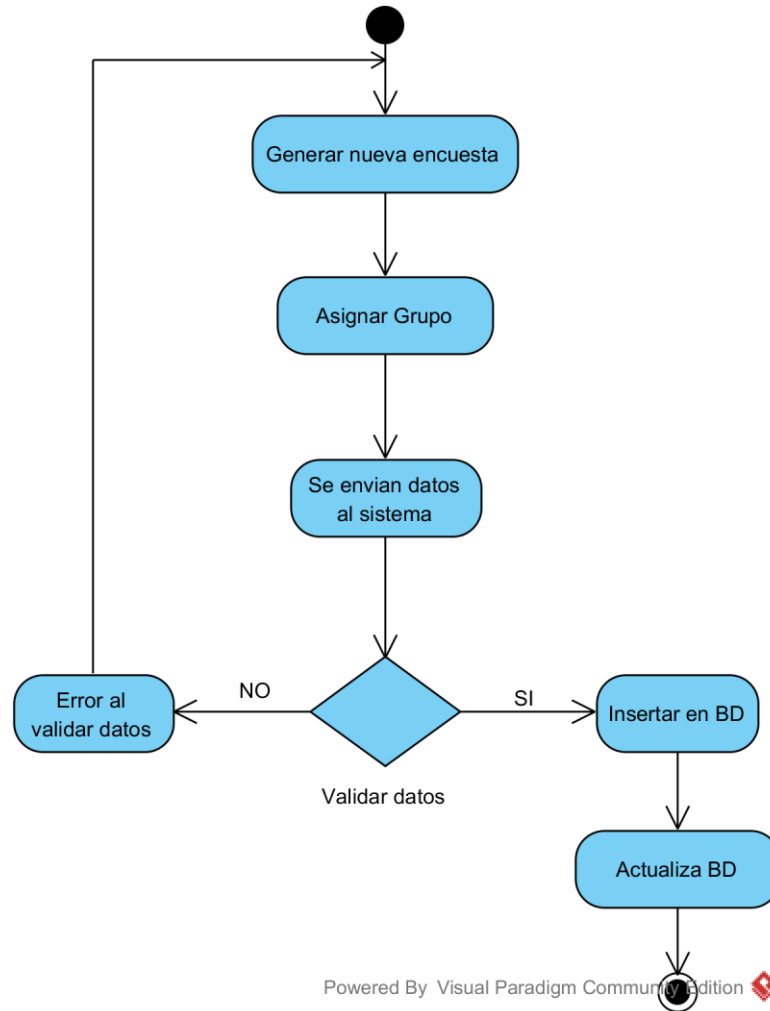


Ilustración 90 Diagrama de Actividades para Elegir un Grupo

Diagrama de Actividades para Consulta de Resultados

El usuario consulta sus preguntas válidas. Elige la que desea revisar. El sistema envía los resultados para que el tutor lo revise.

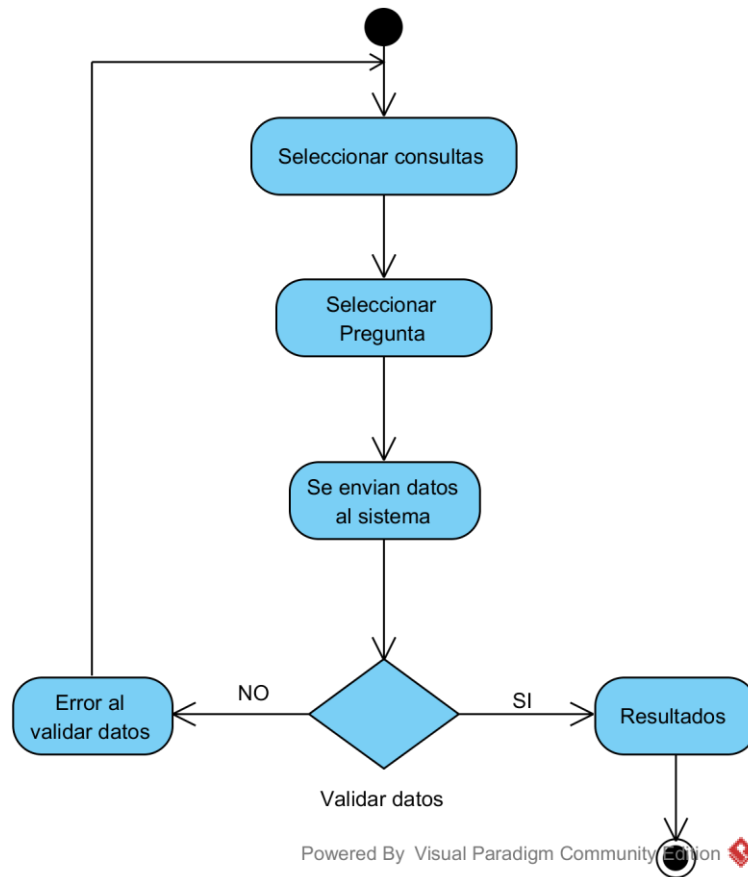


Ilustración 91 Diagrama de Actividades para Consulta de Resultados

Diagrama de Actividades para Contestar Encuesta

El tutor elige de entre todas las preguntas válidas que uno o varios profesores han registrado para él. El sistema entrega las posibles respuestas. El tutor elige una y el sistema registra su respuesta.

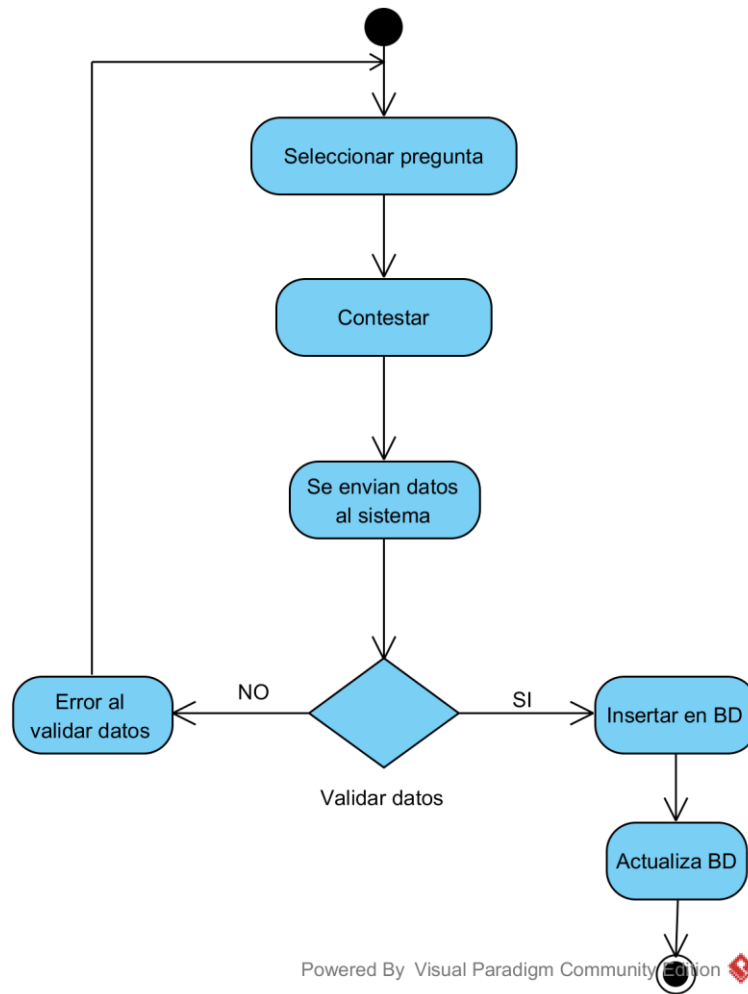


Ilustración 92 Diagrama de Actividades para Contestar Encuesta

Diagrama de Actividades para Crear Evento

El usuario elige crear un nuevo evento. Ingresa en título del evento y una descripción. Se envía al sistema. Se validan los datos del evento y en caso de ser válidos, se da de alta. Se envía un mensaje de respuesta al usuario dependiendo de lo que haya sucedido.

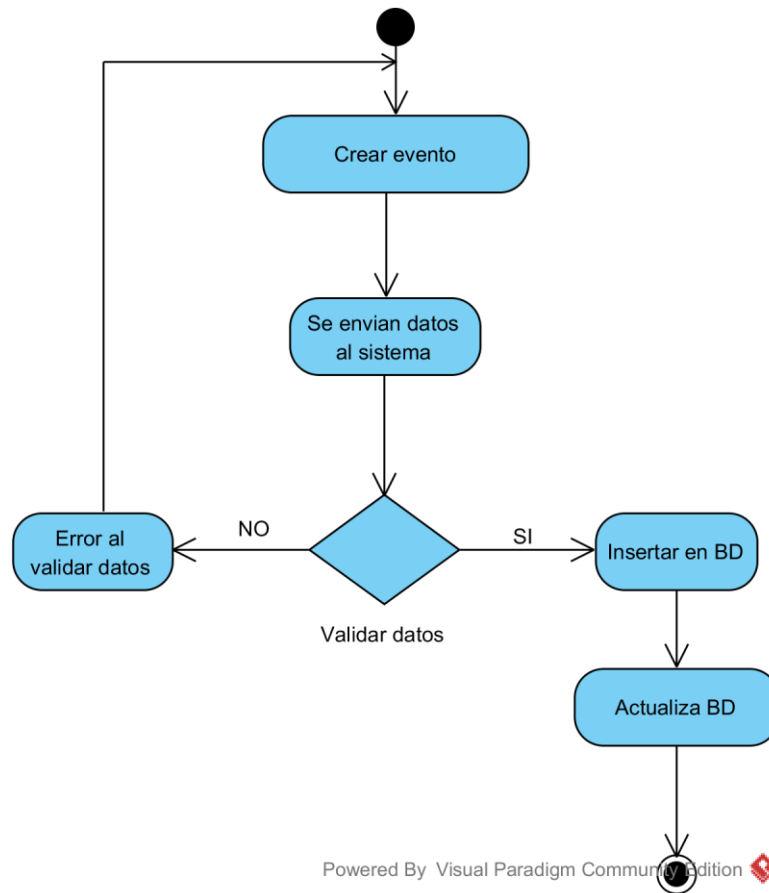


Ilustración 93 Diagrama de Actividades para Crear Evento

Diagrama de Actividades para Asignar Fecha al Evento

El usuario ha añadido un evento. Se introducen los datos y se envían al sistema. El sistema valida los datos. Si son válidos asigna la fecha de realización del evento. Se envía un mensaje de respuesta de acuerdo a lo sucedido.

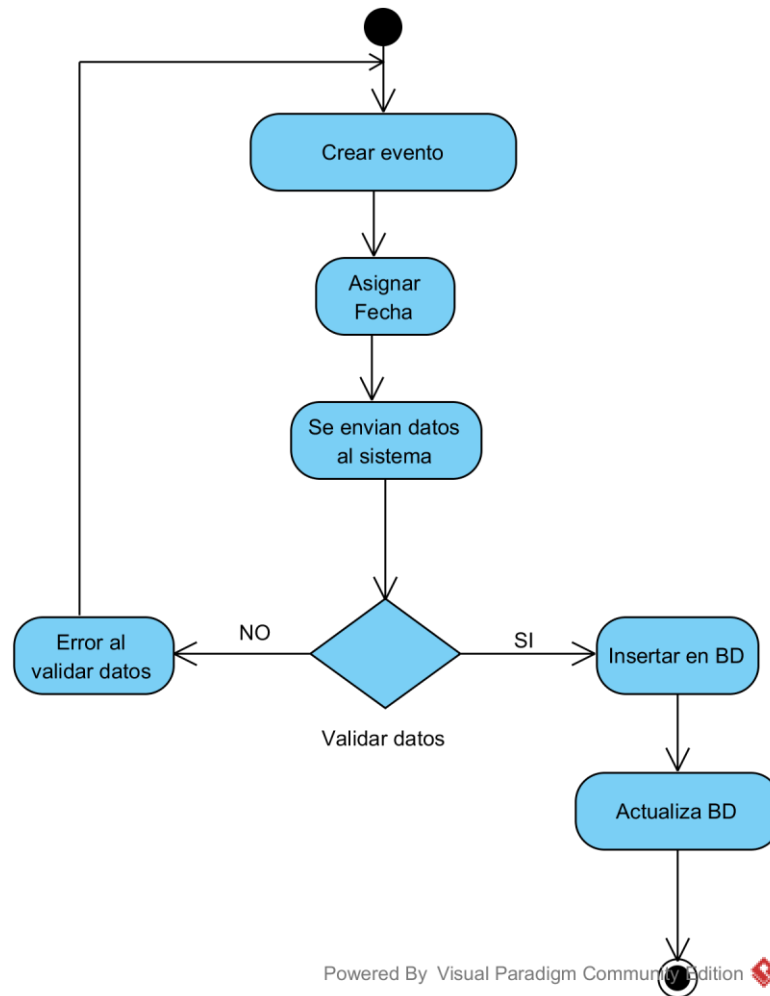


Ilustración 94 Diagrama de Actividades para Asignar Fecha al Evento

Diagrama de Actividades para Elegir Tipo de Evento

El usuario ha añadido un evento. Se elige entre los dos tipos de evento. Se envía un mensaje de respuesta de acuerdo a lo sucedido.

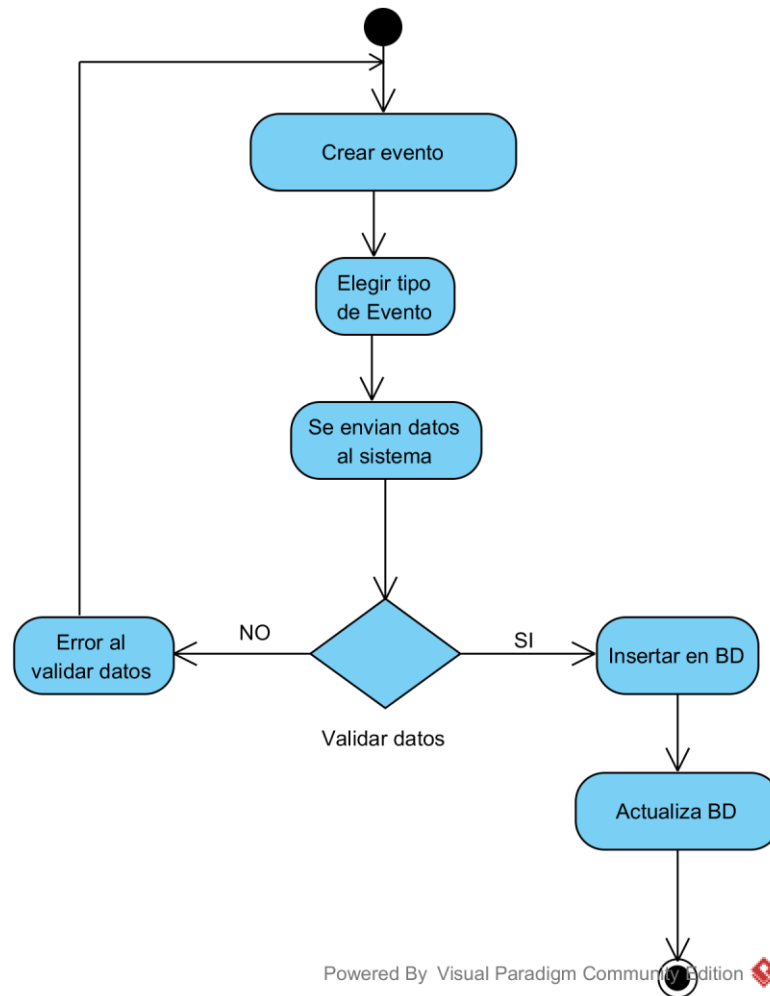


Ilustración 95 Diagrama de Actividades para Elegir Tipo de Evento

Diagrama de Actividades para Elegir Invitados al Evento

El usuario ha añadido un evento. Según el tipo de evento el sistema entrega una lista de posibles invitados a él. Se envían los invitados. El sistema los registra y envía las notificaciones.

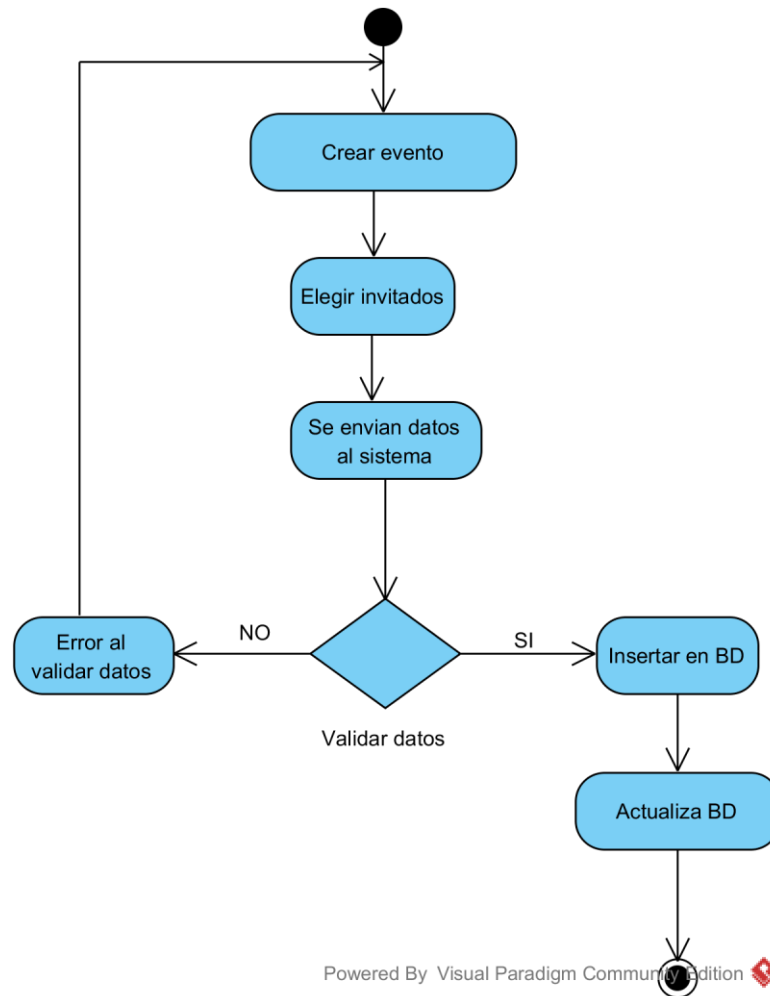


Ilustración 96 Diagrama de Actividades para Elegir Invitados al Evento

Diagrama de Actividades para Responder Invitación a Evento

El tutor elige uno de entre todos los eventos a los que ha sido invitado. El tutor envía una confirmación sobre su participación en el evento. El sistema registra su respuesta.

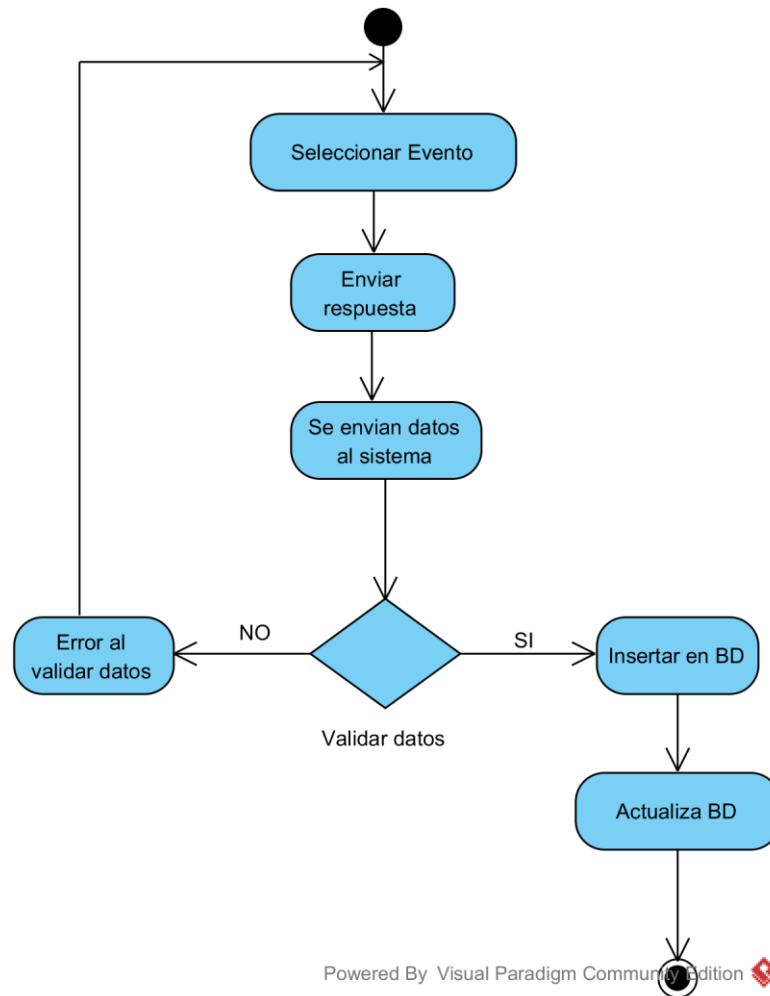


Ilustración 97 Diagrama de Actividades para Responder Invitación a Evento

Diagrama de Actividades para Enviar Mensaje a Tutor

El usuario ingresa al sistema. Selecciona al profesor/tutor con el que dese establecer comunicación. Envía mensaje.

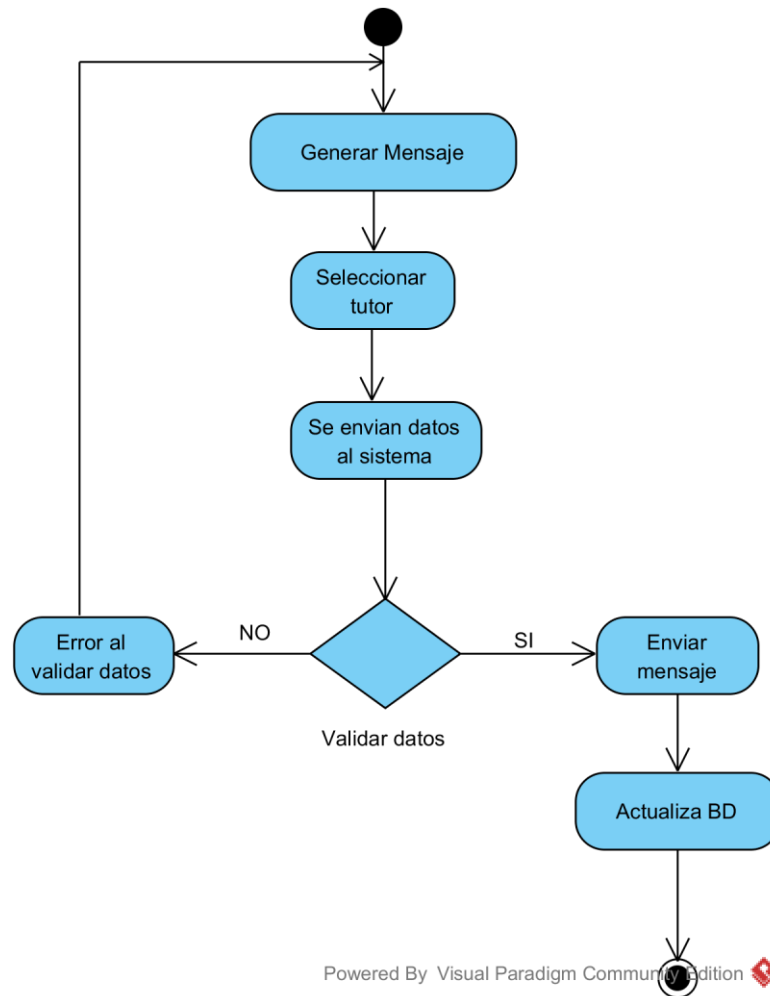


Ilustración 98 Diagrama de Actividades para Enviar Mensaje a Tutor

Diagrama de Actividades para Enviar Mensaje a Profesor

El usuario ingresa al sistema. Selecciona al profesor/tutor con el que dese establecer comunicación. Envía mensaje.

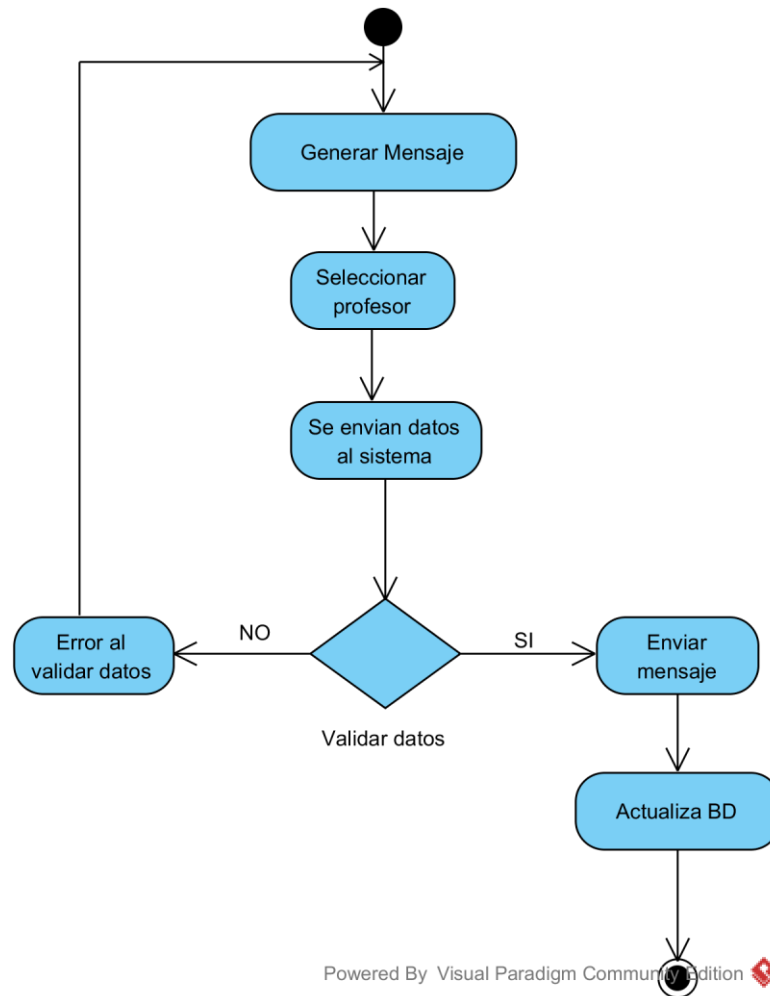


Ilustración 99 Diagrama de Actividades para Enviar Mensaje a Profesor

Diagrama de Actividades para Notificar Mensaje

Se envía mensaje. Se establece comunicación. Notificación de mensaje recibido.

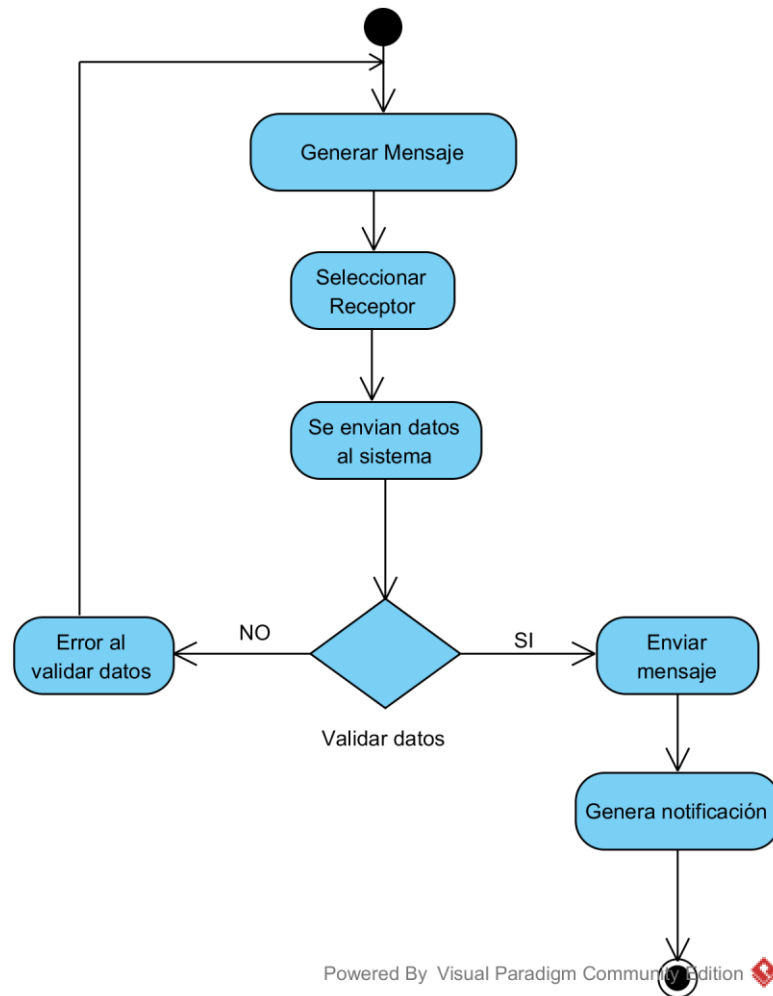


Ilustración 100 Diagrama de Actividades para Notificar Mensaje

Diagrama de Actividades para Enviar Notificación como Escuela

Ingresa al sistema, se genera notificación. Envía a toda la comunidad.

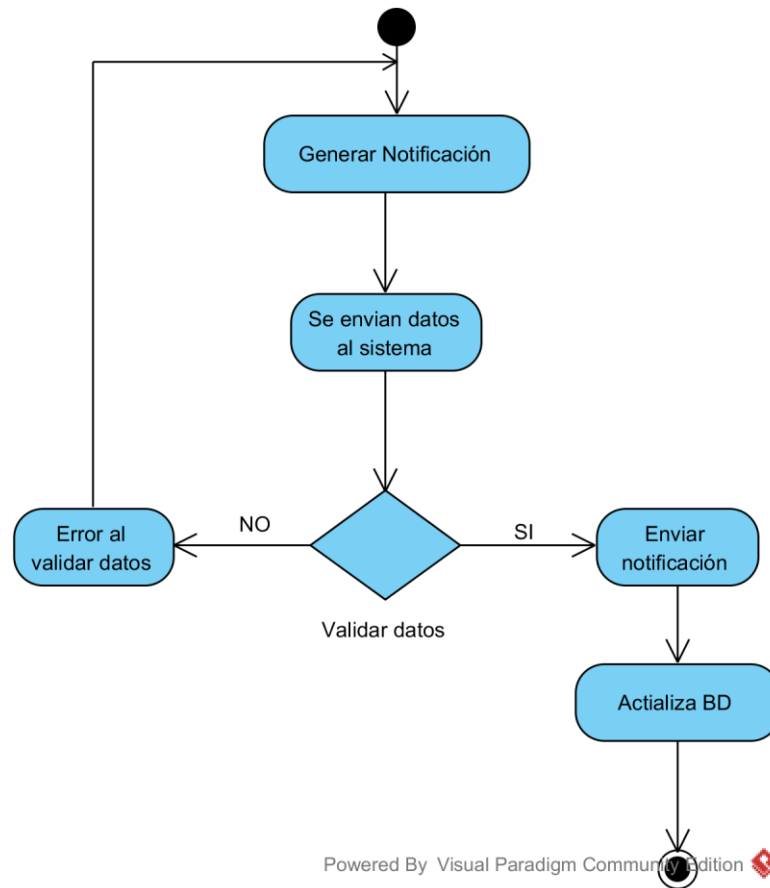


Ilustración 101 Diagrama de Actividades para Enviar Notificación como Escuela

Diagrama de Actividades para Enviar Notificación como Profesor

El profesor ingresa al sistema, selecciona a los tutores a los que dese notificar. Envía notificación.

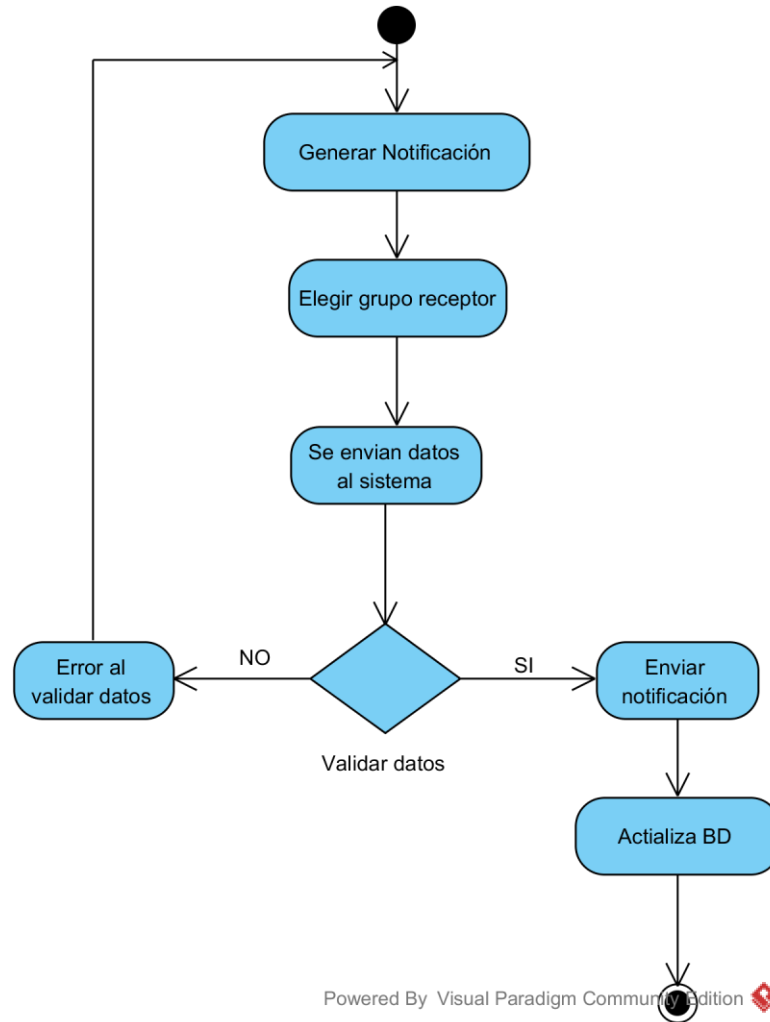


Ilustración 102 Diagrama de Actividades para Enviar Notificación como Profesor

Diagrama de Actividades para Notificaciones Recibidas

Se genera notificación. Se envía notificación. Nueva notificación recibida.

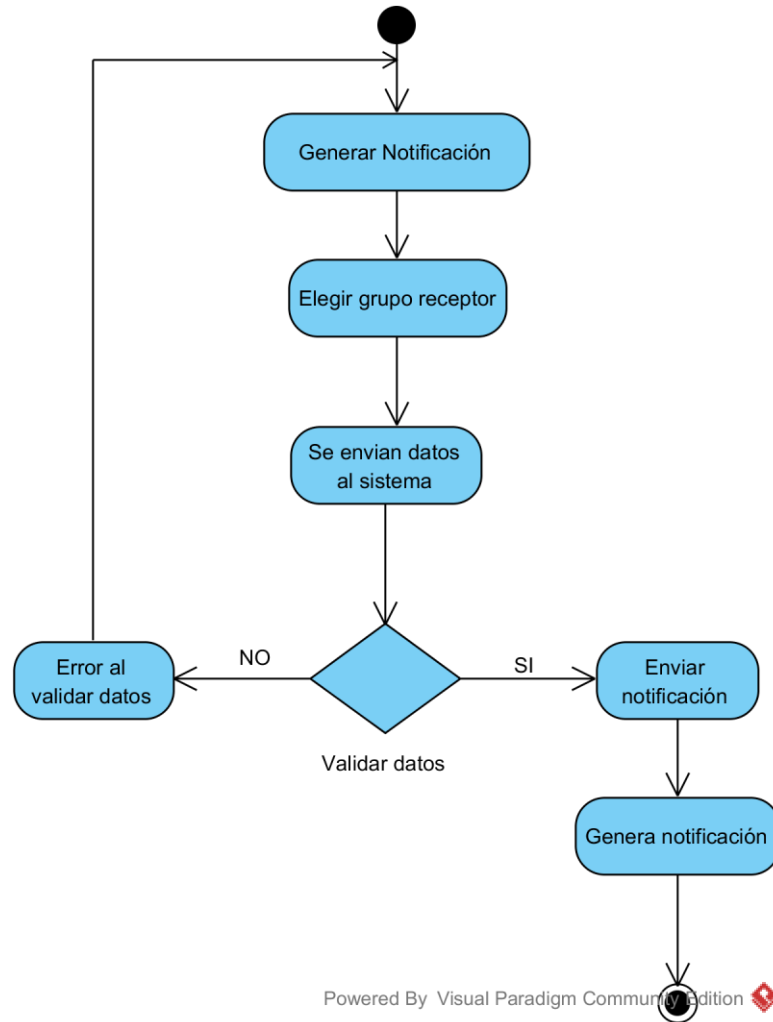


Ilustración 103 Diagrama de Actividades para Notificaciones Recibidas

Capítulo VI. Desarrollo de la API

En este capítulo se abordará el desarrollo de la API, se abordarán algunos detalles del entorno de desarrollo, la programación, arquitectura elegida, validaciones aplicadas.

Entorno de desarrollo

Para desarrollar la gran parte del sistema se empleó el entorno de desarrollo integrado *Visual Studio 2013 with Update 4*, ya que además de ser el entorno de desarrollo por excelencia para aplicaciones escritas en C# con .NET, nos otorga diversas funcionalidades que realizan algunas de las tareas de manera simple y sencilla. Una de las tareas que hace sencilla es la capacidad de publicar de manera remota la aplicación a un entorno de producción.

Se procederá a explicar de manera concisa las características empleadas para el desarrollo de la aplicación.

Proyectos de base de datos

La creación de proyectos de base de datos ofrece la posibilidad de fragmentar la escritura de scripts de tal modo que sea sencilla su edición, se pueden crear un archivo para cada tabla, procedimiento almacenado o vista por separado. El entorno posteriormente ejecuta una especie de proceso de compilación en la que comprueba la coherencia de nuestra estructura de datos y ordena nuestros scripts generando un solo archivo de sentencias SQL ejecutables en nuestro sistema gestor de base de datos. Es importante resaltar que el script generado es útil solo para versiones de SQL server 2008 en adelante, incluyendo las versiones de Azure, que son las usadas para el desarrollo de *Eduator*.

Otra funcionalidad bastante útil y que fue empleada en el desarrollo fue la capacidad de publicar la base de datos directamente desde el IDE, esta manera es la que se recomienda para cuando la aplicación tenga que ser montada en un entorno productivo, dada su sencillez y efectividad para hacerlo.

Proyectos de librerías de clases

Como su nombre lo indica una librería de clases es un conjunto de archivos (principalmente clases), que, en este caso están relacionadas entre sí y otorgan una funcionalidad bien definida, a nivel de análisis podemos ver cada paquete como una librería de clases. Los archivos de código fuente en este proyecto pasan por el mismo proceso de compilación que cualquier otro, con la única diferencia de que no son ejecutables por si mismos puesto que no tienen un punto de entrada al código, sino que resultado de la compilación se generan archivos *.dll* a los cuales se puede referenciar desde otros programas, incluyendo otras librerías de clases.

Proyecto de aplicación web

Al contar también con un nombre auto descriptivo podemos identificar cuál es la labor de este proyecto cumple dentro del desarrollo de la aplicación sin embargo es necesario decir que este proyecto agrupa archivos que estarán expuestos de manera “pública” a través de un servidor web, generalmente estos incluyen archivos *.html*, *.css* o *.js*, pero para el caso de la aplicación construida no es tal el caso. Para *Edutor*, por ser un servicio web, el proyecto contiene principalmente clases de C# que posteriormente el framework empleado se encargará de traducir en elementos accesibles a través del servidor.

Arquitectura del sistema

El framework Web API, que fue el elegido para el desarrollo del sistema reutiliza los conceptos del patrón Modelo-Vista-Controlador, en el cual la vista no existe como en el patrón, mientras que toda la atención la recibe el controlador al cual se accede directamente mediante peticiones HTTP.

Se procederá a explicar cada uno de los componentes esenciales del sistema, que se asemejan en demasía a los paquetes mencionados en la sección de Diagrama de paquetes de este mismo documento. Dichos paquetes también fueron clasificados de acuerdo a la función que cumplen cada uno de ellos dentro del modelo.

Modelo

El modelo está compuesto de diversos proyectos, en donde todos son Proyectos de Librerías de clases, las cuales nos generan componentes binarios a los que después se hace referencia.

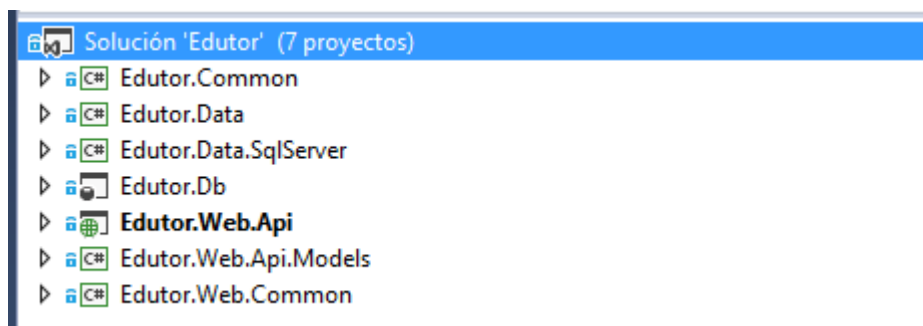


Ilustración 104 Proyectos que componen la aplicación

Los proyectos *Edutor.Data*, *Edutor.Data.SqlServer*, *Edutor.Common*, conforman el modelo. Cada uno cumple con funciones específicas que se describen a continuación.

Edutor.Common

El paquete contiene elementos comunes a la aplicación entera, como valores constantes, extensiones, componentes de *logging* y algunos elementos dedicados a la seguridad.

Por citar un ejemplo tenemos la clase *Constants.cs* que contiene valores constantes que son usados a través de todo el código de la aplicación, como los nombres de los roles, y hace sencillo poder cambiar sus valores al tener que cambiarlos una sola vez en este archivo.

```
namespace Edutor.Common
{
    81 referencias | Antonio Feregrino +1, Hace 18 horas | 12 cambios
    public static class Constants
    {
        20 referencias | Antonio Feregrino +1, Hace 4 días | 3 cambios
        public static class RoleNames
        {
            public const string Teacher = "Teacher";
            public const string Administrator = "Administrator";
            public const string Tutor = "Tutor";
            public const string All = Teacher + "," + Administrator + "," + Tutor;
            public const string TeacherAndTutor = Teacher + "," + Tutor;
            public const string SchoolUser = Teacher + "," + Administrator;
        }
    }
}
```

Ilustración 105 Clase Constants.cs

Edutor.Data

En este paquete residen dos componentes importantes: *Entities* y *QueryProcessors*, el primero contiene clases que representan cada una de las tablas dentro de la base de datos, con todos sus campos como tipos de dato de C#, mientras que las relaciones se especifican mediante referencias entre clases.

El segundo, *QueryProcessors* contiene un conjunto de interfaces que nos describen el comportamiento que se desea reflejar en la base de datos. Es importante señalar que este paquete es independiente de qué sistema gestor de base de datos estemos usando, y es por eso que se usan interfaces, para posteriormente realizar la implementación en otro paquete de acuerdo al SGBD que se elija.

Edutor.Data.SqlServer

Este paquete tiene relación muy cercana con el anterior, ya que en este se implementan las interfaces que definen el comportamiento de la aplicación con la base de datos y el mapeo de objeto a base de datos relacional, pero esta vez se realiza tomando en cuenta el sistema gestor de bases de datos escogido.

Para la tarea se usó la herramienta de mapeo objeto-relacional *NHibernate*, que es la conversión a C# del popular *Hibernate* para Java. Comúnmente el mapeo en *NHibernate* se realiza mediante archivos *XML*, sin embargo para esta aplicación se empleó *Fluent NHibernate* que permite realizar esta labor escribiendo código en C# en lugar de archivos de configuración extra.

Los mapeos están contenidos dentro de la carpeta *Mapping* y a continuación se muestra un ejemplo de mapeo


```

namespace Edutor.Data.SqlServer.Mapping
{
    2 referencias | Antonio Feregrino +1, Hace 13 días | 2 cambios
    public class StudentMap : VersionedClassMap<Student>
    {
        0 referencias | Antonio Feregrino +1, Hace 13 días | 2 cambios
        public StudentMap()
        {
            Table("Students");
            Id(x => x.StudentId);
            Map(x => x.Name).Not.Nullable();
            Map(x => x.Curp).Not.Nullable();
            Map(x => x.Address).Not.Nullable();
            Map(x => x.TutorRelationship).Not.Nullable();
            Map(x => x.IsActive).Nullable();
            Map(x => x.Phone).Nullable();
            Map(x => x.Token).Not.Nullable();

            References<User>(x => x.Tutor).Column("TutorId");

            HasManyToMany(x => x.Groups).Table("Enrollments")
                .ParentKeyColumn("GroupId")
                .ChildKeyColumn("StudentId");
        }
    }
}

```

Ilustración 106 Ejemplo de mapeo ORM

Controlador

El controlador está compuesto por tres proyectos, *Edutor.Web.Common*, *Edutor.Web.Api.Models*, y *Edutor.Web.Api*.

Edutor.Web.Common

Este paquete contiene elementos comunes a la capa web de *Edutor*, como el gestor de transacciones en la base de datos, el *tracer* que nos ayuda a llevar un registro de sucesos importantes que ocurren en la aplicación y un manejador de excepciones que se encarga de “traducir” las excepciones lanzadas a respuestas HTTP con sus respectivos códigos para que estos puedan ser entendibles para los clientes que consuman el servicio.

```

namespace Edutor.Web.Common.ErrorHandling
{
    1 referencia | 6 cambios
    public class GlobalExceptionHandler : ExceptionHandler
    {
        0 referencias | 6 cambios
        public override void Handle(ExceptionHandlerContext context)
        {
            var exception = context.Exception;

            var httpException = exception as HttpException;
            if (httpException != null)
            {
                context.Result = new SimpleErrorResult(context.Request,
                    (HttpStatusCode)httpException.GetHttpCode(), httpException.Message);
                return;
            }

            if (exception is ObjectNotFoundException)
            {
                // En caso de que la excepción sea lanzada al no encontrar un registro en la base de datos se debe
                // devolver un código 404
                context.Result = new SimpleErrorResult(context.Request, HttpStatusCode.NotFound, exception.Message);
                return;
            }
            if (exception is CustomAuthorizationException)
            {
                context.Result = new SimpleErrorResult(context.Request, HttpStatusCode.Unauthorized, exception.Message);
            }
        }
    }
}

```

Ilustración 107 Código para manejar los errores

En el caso de que ocurra una excepción del tipo *ObjectNotFoundException* si no existiera el manejador de excepciones, al cliente que consulta se devolvería un código de error 500 que es el estándar para indicar que ocurrió un error en el servidor, cuando esto no es del todo cierto, ya que si bien el error ocurrió, fue provocado porque el cliente solicitó un recurso que no existía, por lo tanto es mejor responder con el código 404, que significa No Encontrado.

Edutor.Web.Api.Models

Este paquete es uno de los más sencillos pero a su vez uno de los más importantes puesto que contiene los modelos con los que estarán interactuando los clientes que usen la API de *Edutor*, es importante explicar la razón de que estos modelos no sean los mismos que los que existen en el paquete *Edutor.Data*. Los modelos existentes en este paquete tienen limitaciones en cuanto a los datos que pueden recibir de interacciones externas. Dichas limitaciones las podemos usar a nuestro favor para evitar que se realicen interacciones con datos que no deseamos recibir o compartir a través de la capa pública de *Edutor*.

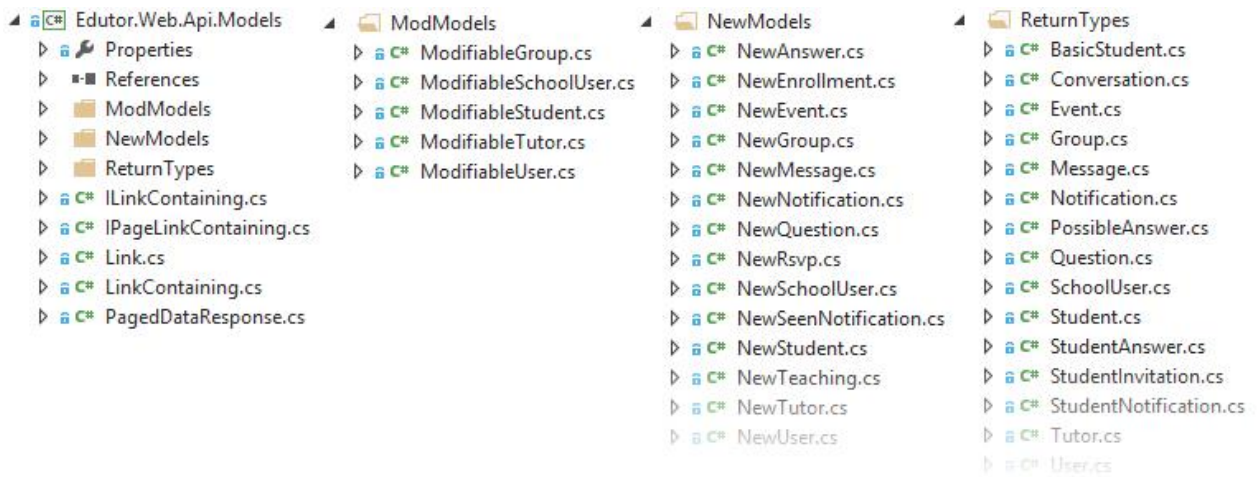


Ilustración 108 Composición del proyecto Models

Para ejemplificar, se tomaron dos clases que hacen referencia a una misma entidad dentro de diversas etapas del sistema, por un lado tenemos la clase *Student* y la clase *NewStudent*, la primera pertenece a la capa del modelo y la segunda al controlador.

```

namespace Edutor.Data.Entities
{
    61 referencias | 3 cambios
    public class Student : IVersionedEntity
    {
        32 referencias | 1 cambio
        public virtual int StudentId { get; set; }
        2 referencias | 1 cambio
        public virtual int TutorId { get; set; }
        4 referencias | 1 cambio
        public virtual string TutorRelationship { get; set; }
        5 referencias | 1 cambio
        public virtual bool IsActive { get; set; }
        4 referencias | 1 cambio
        public virtual string Address { get; set; }
        4 referencias | 1 cambio
        public virtual string Phone { get; set; }
        6 referencias | 1 cambio
        public virtual string Curp { get; set; }
        7 referencias | 1 cambio
        public virtual string Name { get; set; }
        7 referencias | 1 cambio
        public virtual string Token { get; set; }
        10 referencias | 1 cambio
        public virtual User Tutor { get; set; }
        3 referencias | 1 cambio
        public virtual IList<Group> Groups { get; set; }
        33 referencias | 1 cambio
        public virtual byte[] Version { get; set; }
    }
}

namespace Edutor.Web.Api.Models.NewModels
{
    4 referencias | 6 cambios
    public class NewStudent
    {
        [Required]
        0 referencias | 2 cambios
        public int TutorId { get; set; }

        [StringLength(3, MinimumLength = 3)]
        0 referencias | 4 cambios
        public string TutorRelationship { get; set; }

        [Required]
        [StringLength(512)]
        0 referencias | 4 cambios
        public string Address { get; set; }

        [StringLength(10)]
        0 referencias | 3 cambios
        public string Phone { get; set; }

        [Required]
        [StringLength(18, MinimumLength = 18)]
        0 referencias | 3 cambios
        public string Curp { get; set; }

        [Required]
        [StringLength(60)]
        0 referencias | 3 cambios
        public string Name { get; set; }
    }
}

```

Ilustración 109 Comparativa entre clases

En la imagen, se puede observar que a pesar de que las dos clases representan a un estudiante (y por ello comparten propiedades como *Curp* y *Name*) no tienen las mismas, y es que la que tiene el prefijo *New* es la que el cliente enviará al servicio web para interactuar como parámetro.

Podría parecer un redundante pero tiene múltiples razones de ser, una de ellas tiene que ver mucho con la seguridad del servicio. En la clase *NewStudent* se define solo un subconjunto de propiedades sobre las que se permitirá la interacción, es decir, no se otorga acceso directamente a todas las propiedades de la entidad *Student*, previniendo así que el cliente que consuma el servicio modifique, voluntaria o involuntariamente, propiedades no deseadas en la capa de datos.

Otra razón es que al hacer más ligeras nuestras entradas, al quitar propiedades que no son necesarias para la operación a realizar, también hacemos que la comunicación sea más rápida y que se consuma menor ancho de banda para ser transportado.

Links de hiper media

Para facilitar la navegación dentro de los recursos que ofrece la API, existe la clase *Link* que está relacionada con todos los tipos de retorno, ya que todos ellos contienen una colección de links que apuntan hacia elementos relacionados con la consulta que se realizó. La implementación de dicha clase es bastante sencilla, puesto que solo se compone de la *Url*, el tipo de relación entre el link y el recurso que lo contiene y el método *HTTP* con el que se debe llamar para que se ejecute.

Validación de datos para los modelos de entrada

Cuando se expone una API pública es importante realizar una validación adecuada de los datos que se aceptarán, puesto que es probable que intencional o no intencionalmente alguien haga una petición con datos inválidos que pueden provocar comportamientos inesperados dentro del sistema expuesto. Edutor emplea la validación de datos a través de la librería *DataAnnotations*, proporcionada por el framework *.NET*.

La forma en que se especifican las reglas de validación es mediante la utilización de atributos decoradores. Los decoradores de la librería sirven para especificar una gran variedad de reglas, por citar un ejemplo, se mostrará las reglas de validación de una entidad del sistema.

Se empeará las clase `NewUser`:

```
2 referencias | 5 cambios
public class NewUser
{
    [Required]
    [StringLength(70)]
    0 referencias | 4 cambios
    public string Name { get; set; }

    [Required]
    [StringLength(18, MinimumLength = 18)]
    0 referencias | 3 cambios
    public string Curp { get; set; }

    [EmailAddress]
    [Required]
    [StringLength(60)]
    0 referencias | 4 cambios
    public string Email { get; set; }

    [Required]
    [StringLength(512)]
    0 referencias | 4 cambios
    public string Address { get; set; }
}
```

Ilustración 110 Fragmento de la clase `NewUser`

Las reglas del negocio dictan que el nombre del usuario en todos casos es requerido, y este no debe exceder de 70 caracteres de longitud, lo cual es especificado usando los atributos `Required` y `StringLength`. En México, la CURP siempre una cadena de 18 caracteres, y nuevamente para las reglas del negocio es un valor requerido, esta vez se usa el atributo `StringLength` para establecer una longitud máxima de 18 y la mínima del mismo valor, el atributo `Required` la hace necesaria para que el modelo sea válido.

Para el Email, las reglas del negocio especifican que debe ser una cadena obligatoria de a lo más 60 caracteres de longitud, nuevamente los atributos ya conocidos hacen su aparición, pero esta vez se añadió un tercero, que es `EmailAddress` que ayudará a comprobar que la cadena que se ingresó sea en realidad un correo electrónico válido, esto nos evita tener que emplear una expresión regular para realizar dicha validación.

Ahora se muestra un fragmento de la clase `NewTutor`

```
public class NewTutor : NewUser
{
    [StringLength(20)]
    0 referencias | 3 cambios
    public string Job { get; set; }

    [StringLength(10)]
    0 referencias | 3 cambios
    public string JobTelephone { get; set; }
}
```

Ilustración 111 Fragmento de la clase NewTutor

Como las reglas del negocio indican, un tutor puede o no tener un empleo, por tanto las propiedades dentro del modelo no están marcadas como requeridas, sin embargo, en caso de que estas tengan un valor, estarán limitadas a cierta longitud de caracteres.

Mediante los atributos de validación únicamente se especifican las reglas, pero no se realiza la validación, la validación se realiza en el componente que a continuación se explica.

Edutor.Web.Api

De todos los paquetes, este es el único que está expuesto directamente, ya que es el que recibe las peticiones y posteriormente las enruta a los controladores asignados para procesarlas según sea el caso, y es precisamente por eso que es uno de los más detallados y con una complejidad mayor.

Para comenzar a explicarlo, podemos iniciar por la carpeta *Controllers* que como su nombre indica, almacena los controladores de la aplicación, los controladores son los siguientes

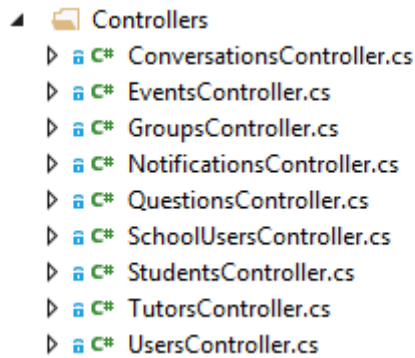


Ilustración 112 Composición del controlador

Existe un controlador por cada uno de los extremos a través de los que se realizarán las interacciones entre profesores y tutores, hablando de programación, cada uno de ellos contiene métodos que quedan expuestos a través de una URL.

Descripción de un controlador

Como se mencionó antes, cada controlador recibe interacciones que le llegan directamente desde peticiones HTTP, por lo cual es importante describir un controlador, para tener una noción de cómo es que funcionan, para ello se tomará el controlador `QuestionsController`.

```

namespace Edutor.Web.Api.Controllers
{
    /// <summary>
    /// Conjunto de extremos REST que permiten operar con los servicios de creación y i
    /// </summary>
    [UnitOfWorkActionFilter]
    1 referencia | 11 cambios
    public class QuestionsController : ApiController
    {
        private readonly IPostQuestionMaintenanceProcessor _postQuestion;
        private readonly IPutQuestionsUpdateProcessor _updateQuestion;
        private readonly IGetQuestionsInquiryProcessor _getQuestions;
        private readonly IGetStudentsInquiryProcessor _getStudents;
        private readonly IPagedDataRequestFactory _pagedDataRequestFactory;

        0 referencias | 1 cambio
        public QuestionsController(IPostQuestionMaintenanceProcessor postQuestion,
            IGetQuestionsInquiryProcessor getQuestions,
            IGetStudentsInquiryProcessor getStudents,
            IPutQuestionsUpdateProcessor updateQuestion,
            IPagedDataRequestFactory pagedDataRequestFactory)
        {
            _postQuestion = postQuestion;
            _getQuestions = getQuestions;
            _getStudents = getStudents;
            _pagedDataRequestFactory = pagedDataRequestFactory;
            _updateQuestion = updateQuestion;
        }
    }
}

```

Ilustración 113 Fragmento de la clase QuestionsController

Para comenzar, el controlador hereda de la clase ApiController se debe tomar como recomendación nombrar la clase como [Entidad]Controller, posteriormente se declaran algunos miembros que ayudan al controlador a ejecutar las acciones que expone al público. Son asignadas en con constructor de la clase, usando el patrón de diseño Inyección de Dependencias.

```

[Route("questions")]
[HttpPost]
[ResponseType(typeof(Question))]
[Authorize(Roles = Constants.RoleNames.SchoolUser)]
[ValidationActionFilter]
0 referencias | 5 cambios
public IHttpActionResult AddQuestion(NewQuestion question)
{
    var ret = _postQuestion.AddQuestion(question);
    // Genera la respuesta con su correspondiente código de estatus HTTP
    var result = new ModelPostedActionResult<Question>(Request, ret);
    return result;
}

```

Ilustración 114 Código del método AddQuestion

El método `AddQuestion` está decorado con varios atributos, el primero `Route` le indica al framework que el método es accesible en la ruta `http://www.ejemplo.com/questions`, el segundo atributo es `HttpPost` el cual indica que el método debe ser solo accesible cuando se ejecute una petición `Post`, el tercero `ResponseType` especifica que el tipo de retorno para la petición `HTTP` (a pesar de que el método internamente regresa `IHttpActionResult`) es `Question`.

Otro de los atributos importantes es `Authorize` que indica que la petición debe ser autenticada, el atributo también recibe como parámetro `Roles` con el valor de la constante `SchoolUser` que especifica que una vez autenticada la petición debe estar autorizada para ejercer las tareas de un usuario escolar. El atributo `ValidationActionFilter` especifica que el valor del parámetro `question` debe ser validado conforme a lo especificado en la clase `NewQuestion` y en caso de no estarlo el método no se ejecutará.

```
[Route("questions/{questionId:int}")]
[HttpGet]
[Authorize(Roles = Constants.RoleNames.All)]
0 referencias | 4 cambios
public Question GetQuestion(int questionId)
{
    return _getQuestions.GetQuestion(questionId);
}
```

Ilustración 115 Código del método `GetQuestion`

El método `GetQuestion` es aún más sencillo, gracias a la inyección de dependencias, este solo tiene dos atributos que lo decoran, en este caso no es necesario `ResponseType` puesto que tanto interna como externamente el tipo de retorno es `Question`. El atributo `Route`, como ya se vio especifica a través de qué ruta será accesible, para este caso una petición a `http://www.ejemplo.com/questions/3` hará que se ejecute el método con el parámetro

questionId con un valor de 3. `HttpGet` simplemente indica que la petición debe ser del tipo `Get`.

Por último, aparece nuevamente el atributo `Authorize` pero esta vez con el valor de `Roles` igual a la constante `All` que indica que todos los usuarios pueden acceder a ese método, siempre y cuando estén autenticados en el servicio.

Todos los controladores dentro de la aplicación están implementados de manera similar, es decir, empleando atributos decoradores para definir las rutas, los tipos de petición, tipos de respuesta de cada método, validaciones y la seguridad.

Autenticación y autorización

La mayoría de peticiones que se realizan a la API de Edutor deben ser autenticadas y autorizadas para poder ser procesadas por el servicio. De entrada se podría pensar que autenticación y autorización son lo mismo, pero no es así, en particular, para Edutor entenderemos lo siguiente:

- **Autenticación.** El proceso de autenticación implica corroborar que el cliente que realiza la petición existe en los registros de la aplicación, es decir que está registrado en nuestra base de datos independientemente del rol o roles que el cliente posea.
- **Autorización.** Este proceso ocurre una vez que el cliente es autenticado e implica corroborar que el rol que tiene asignado es el necesario y suficiente para ejecutar la acción que solicita.

Proceso de autorización y autenticación en el servidor

En cuanto una petición es enrutada hacia un recurso que está protegido mediante el atributo `Authorize`, la aplicación realizará el proceso inverso al que realizan los clientes, es decir, toma el mensaje de petición HTTP y comprueba la existencia de un encabezado de autorización del tipo básico en caso de no existir rechaza la petición con un error 401.

```

private bool CanHandleAuth(HttpRequestMessage request)
{
    return (request.Headers != null
        && request.Headers.Authorization != null
        && request.Headers.Authorization.Scheme
            .Equals(Constants.SchemeTypes.Basic, StringComparison.OrdinalIgnoreCase));
}

```

Ilustración 116 Código del método CanHandleAuth

Posteriormente lo toma y lo decodifica, obteniendo las credenciales enviadas por el cliente. Una vez que cuenta con las credenciales, el *token* o la combinación usuario-contraseña según sea el caso, el sistema busca en la base de datos, si existe alguna coincidencia procede al siguiente paso, en caso de que no, significa que el cliente no podrá ser autenticado y su petición debe ser rechazada.

Si algún registro coincidió, se recupera y se extrae de él la información necesaria para asignarle los roles que determinan el nivel de autorización con el que se cuenta. Por último, si el nivel de autorización no es el suficiente, la petición es rechazada con un código de error 401.

```

public virtual IPrincipal GetPrincipal(Student student)
{
    var identity = new GenericIdentity(student.Tutor.UserId.ToString(), Constants.SchemeTypes.Basic);

    identity.AddClaim(new Claim(ClaimTypes.Role, Constants.RoleNames.Tutor));
    identity.AddClaim(new Claim(Constants.CustomClaimTypes.StudentId, student.StudentId.ToString()));
    identity.AddClaim(new Claim(Constants.CustomClaimTypes.TutorId, student.Tutor.UserId.ToString()));
    identity.AddClaim(new Claim(Constants.CustomClaimTypes.UserId, student.Tutor.UserId.ToString()));

    return new ClaimsPrincipal(identity);
}

```

Ilustración 117 Fragmento de código para la asignación de roles

Paginación

Con gran frecuencia suele suceder que la información que se desea obtener es demasiada, por ejemplo: obtener una lista de todos los alumnos inscritos en el plantel escolar. Dicha petición no implicaría mayor problema para casos donde la cantidad de alumnos sea pequeña, que no es el común de los casos.

El costo de consultar en una sola operación es bastante alto, además de que consume mayor ancho de banda y puede provocar mayores gastos en caso de que el cliente esté funcionando bajo un plan de datos móviles, otra desventaja es que entre más grande sea la respuesta, el cliente que la solicitó tendrá mayor dificultad para procesarla una vez recibida.

En Edutor, se implementa la paginación de resultados mediante el uso de la clase genérica `PagedDataResponse` la clase envuelve a una colección de objetos cualquiera y una colección de objetos de la clase `Link` que contienen los hipervínculos de navegación entre las páginas de la lista. También incluye propiedades para especificar el conteo de páginas, el número total de elementos que se presentan y el número de páginas que se está mostrando en ese momento.

Para realizar la paginación se analiza la URL de la petición, buscando en ella los parámetros `pageSize` y `pageNumber` que indicarán la respectivamente el tamaño de la página a consultar y el número que se desea obtener.

```
public PagedDataRequest Create(Uri requestUri)
{
    int? pageNumber,
        pageSize;

    try
    {
        var valueCollection = requestUri.ParseQueryString();
        pageNumber = PrimitiveTypeParser.Parse<int?>(valueCollection["pageNumber"]);
        pageSize = PrimitiveTypeParser.Parse<int?>(valueCollection["pageSize"]);
    }
    catch (Exception e)
    {
        _log.Error("Error al interpretar la entrada", e);
        throw new HttpException((int)HttpStatusCode.BadRequest, e.Message);
    }
    pageNumber = pageNumber.GetBoundedValue(defaultValue: 1, Constants.Paging.MinPageNumber);
    pageSize = pageSize.GetBoundedValue(DefaultPageSize, min: 1, MaxPageSize);

    return new PagedDataRequest(pageNumber.Value, pageSize.Value);
}
```

Ilustración 118 Fragmento de código para la creación de paginación

El método `Create` recibe como parámetro el `Uri` de la petición, posteriormente, usando el método `ParseQueryString` obtiene un diccionario llave-valor con los parámetros enviados a través de ella, solo para intentar convertir los valores de dichos parámetros a números enteros que más adelante se usarán para realizar la consulta a la base de datos. Mediante los métodos `GetBoundedValue` se impide que se realicen consultas con valores no permitidos como -1, 0 o que estos excedan el máximo valor permitido.

Librería cliente de Edutor

Para la etapa de pruebas se creó una librería cliente de la API, que consiste de otro componente de software separado del desarrollo principal, cuya tarea consiste en realizar las peticiones HTTP por el desarrollador que la utiliza. Para este caso el componente fue desarrollado completamente en C#, haciendo uso de las mismas herramientas que se emplearon durante el desarrollo principal, aunque al ser un servicio web, la librería se pudo haber desarrollado en cualquier otro lenguaje.

La librería cliente envuelve las peticiones HTTP, otorgando al desarrollador de una capa más de abstracción para consumir el servicio, la que hace que el servicio web pueda ser operado a través de métodos y no tenga que escribir y realizar las peticiones de manera tradicional. La importancia de la existencia de este componente radica en que gracias a él se pudieron desarrollar pruebas automatizadas y crear aplicaciones que se pudieran llevar a las escuelas para implementar el sistema.

Capítulo VI. Implementación y pruebas

Implementación

Servidor Web

En el capítulo IV se analizaron las tecnologías de desarrollo y entre ellas se eligió emplear Microsoft Azure, que es una plataforma relativamente nueva en tiempos y para otros desarrollos dentro de la escuela. Como proveedor de servicios en la nube, Microsoft Azure, nos ofrece una variedad de paquetes de servidores web, entre ellos la opción de montar un sitio web en servidores a medida a la vez que nos garantiza la seguridad de los datos que se manejan dentro de la misma.

Para montar la aplicación se eligió la opción **Aplicación Web** cuya funcionalidad está específicamente dedicada a sitios y servicios web y a la cual se accede vía FTP o mediante una funcionalidad conocida como *Web Deploy* para publicar y depurar el servicio. En cuanto a las especificaciones de hardware del servidor, cabe destacar que son escalables, pero la configuración para la etapa de pruebas se detalla en la siguiente tabla

Característica	Detalle
Sistema operativo	Azure Guest OS (Windows Server 2012)
Capacidad de almacenamiento	10 Gb
Servidor Web	IIS
Bases de datos	0 (Se empleará un servidor exclusivo)
Almacenamiento de bases de datos	0
Memoria RAM	3 Gb
Asistencia	Telefónico, 24 / 7
Ubicación geográfica	Centro y Sur de EE. UU.
Soporte para .NET V4.5	Sí
Costo	910.66 / mes

Tabla 72 Características del servidor de pruebas

A destacar que la manera en que fue desarrollado el sistema permite su implantación en servidores propios de cada plantel escolar, otorgando así de flexibilidad en la implantación.

Servidor de base de datos

Así como para el servidor web, se eligió usar una solución dedicada dentro de las que nos ofrece la plataforma de servicios en nube elegida, en este caso el servicio elegido es **Base de datos SQL** el cual nos otorga un servidor dedicado para ejercer la funcionalidad de bases de datos, también nos da la posibilidad de replicar los datos además de las características de seguridad de SQL Server, los detalles del servidor se muestran en la siguiente tabla

Característica	Detalle
Sistema operativo	Azure Guest OS (Windows Server 2012)
Capacidad de almacenamiento	250 Gb
SGBD	SQL Azure
Bases de datos	1
Almacenamiento de bases de datos	250 Gb
Memoria RAM	3 Gb
Asistencia	Telefónico, 24 / 7
Ubicación geográfica	Centro y Sur de EE. UU.
Soporte para .NET V4.5	[No aplica]
Costo	230.00 / mes

Tabla 73 Características del servidor de base de datos

Servidor Web para la etapa de desarrollo

Durante la etapa de desarrollo se empleó un servidor similar al de producción, ya que el equipo que desarrolla cuenta con el beneficio de Microsoft BizSpark para poder hacerlo sin incurrir en gasto alguno. Se empleó, además, el equipo de los integrantes del equipo, las características de uno de ellos se muestra a continuación

Característica	Detalle
Sistema operativo	Windows 8.1
Capacidad de almacenamiento	918 Gb
Servidor Web	IIS 8.5

Bases de datos	1
Almacenamiento de bases de datos	20 Gb
Memoria RAM	8 Gb
Asistencia	[No aplica]
Ubicación geográfica	[No aplica]
Soporte para .NET V4.5	Sí
Costo	[No aplica]

Tabla 74 Características del servidor web para la etapa de desarrollo

En cuanto al software que cumple la funcionalidad de servidor web se menciona que se usará IIS, un servicio que únicamente funciona en sistemas operativos Windows, dicho software debe ser habilitado por el usuario puesto que viene pre-instalado, tanto en la versión de servidor del sistema operativo como en las versiones de escritorio. En cuanto a las versiones de los frameworks de desarrollo, estas también vienen preinstaladas en el sistema operativo (de Windows 8 en adelante).

Servidor de bases de datos para la etapa de desarrollo

Al igual que para el servidor web, se empleará un ambiente similar al que se empleará en la etapa de producción así como los equipos de los integrantes, uno de ellos con las siguientes características

Característica	Detalle
Sistema operativo	Windows 8.1
Capacidad de almacenamiento	918 Gb
SGBD	Microsoft SQL Server Enterprise (V12.0.2)
Bases de datos	1
Almacenamiento de bases de datos	20 Gb
Memoria RAM	8 Gb
Asistencia	[No aplica]
Ubicación geográfica	[No aplica]
Soporte para .NET V4.5	[No aplica]
Costo	[No aplica]

Tabla 75 Características del servidor de base de datos durante el desarrollo

En este caso, el sistema gestor de bases de datos es un programa que se debe obtener por separado, ya sea en un disco de instalación o vía web, durante la instalación se recomienda elegir el modo de autenticación mixta, para poder acceder a ella mediante una contraseña y un nombre de usuario.

También se recomienda la instalación de SQL Server Management Studio, que nos permitirá gestionar y supervisar la base de datos tanto en la etapa de desarrollo como en la de pruebas.

Aprovisionamiento de los servidores necesarios

Para realizar la implementación de servidores, es necesario acceder al portal de gestión de Windows Azure (<http://portal.azure.com>) con la cuenta de administrador de servicios y desde ahí realizar dicha acción.

Aprovisionamiento del servidor de base de datos

Una vez que se accedió al portal de gestión de Azure, a en la esquina izquierda superior se debe elegir la opción de *NUEVO* tras lo cual aparecerá un panel del que se elige la opción *Data + Storage* y en el nuevo panel seleccionar la opción de *SQL Database*:

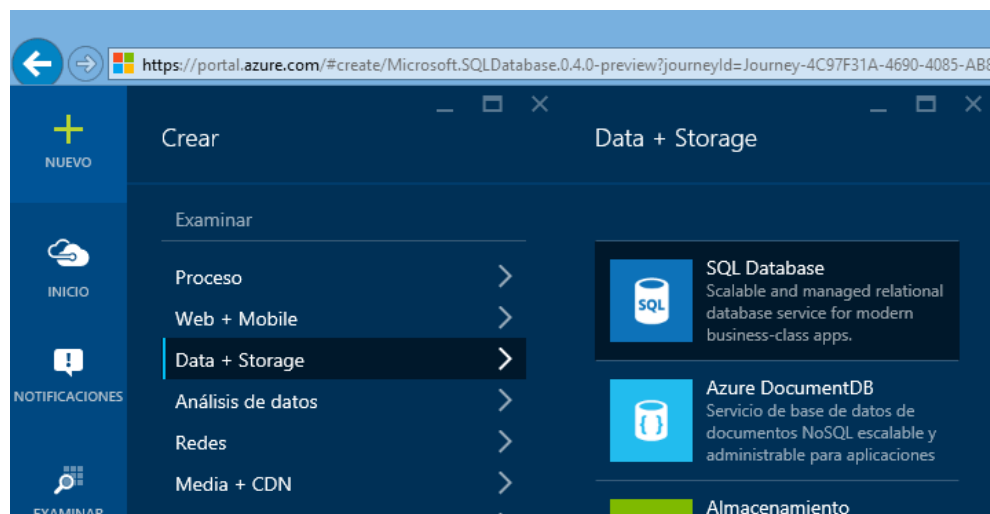


Ilustración 119 Creación de un servidor de base de datos

Tras lo cual se abrirá un tercer panel de configuración en el cual se puede elegir el tipo, servidor y otras características de la base de datos. Es importante que al asignar un nombre de inicio de sesión y contraseña al servidor se guarden dichos valores puesto que son requeridos más adelante al crear la base de datos y durante la conexión con la aplicación.

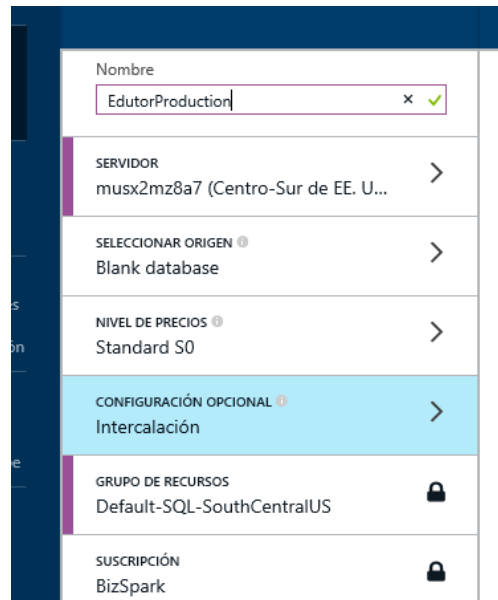


Ilustración 120 Configuración del servidor de base de datos

Una vez configuradas las opciones para nuestra base de datos, se debe elegir crearla y esperar unos cuantos minutos a que la operación sea realizada, una vez completado este proceso, se puede continuar con el aprovisionamiento.

Aprovisionamiento del servidor de aplicaciones web

Nuevamente se debe acceder al portal de gestión y elegir la opción *NUEVO* después, en el panel que se abrió elegir *Web + Mobile* y en él seleccionar *Aplicación Web*.

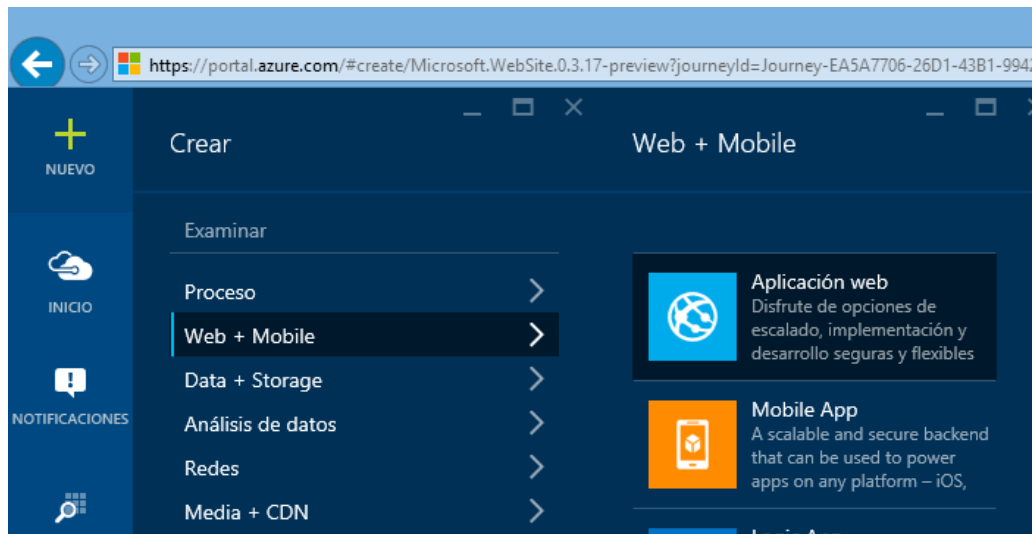


Ilustración 121 Creación de un servidor de aplicaciones web

Nuevamente se debe acceder a un tercer panel de configuración perteneciente al servidor de aplicaciones web que se está a punto de crear. Debe ser configurado según las necesidades que se tengan, procurando colocarlo en la misma ubicación geográfica que la base de datos, para que así el tiempo de transferencia de datos entre ellos sea el mínimo.

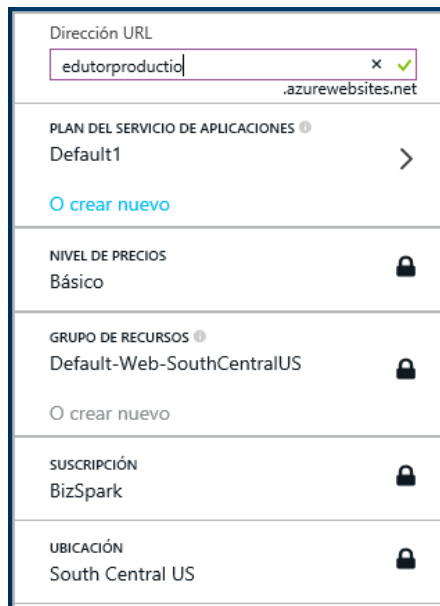


Ilustración 122 Configuración del servidor de aplicaciones web

Una vez completados los pasos de aprovisionamiento se puede continuar con la implementación.

Creación de la base de datos

Una vez preparado el entorno para la implementación, se debe continuar con la creación de la base de datos. Usando el entorno de desarrollo, se debe seleccionar el proyecto de bases de datos llamado **Edutor.Db**. Dicho proyecto está formado por dos carpetas, *Scripts* y *Tables* cuya función es más que explícita en los nombres. A pesar de que los tanto los scripts de creación de tablas y de inserción de datos están definidos en archivos separados estos se conjuntan en un único archivo al momento de la publicación de la base de datos

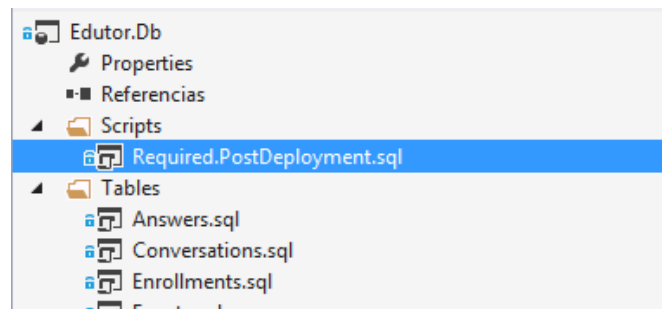
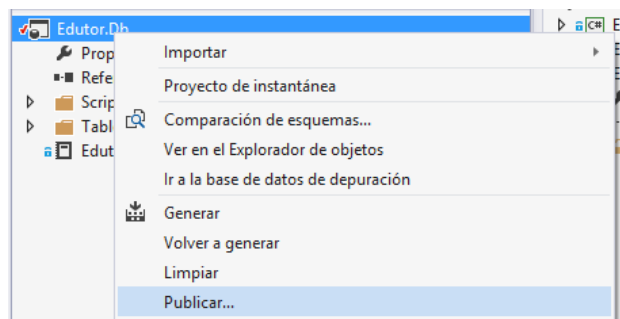


Ilustración 123 Script de información post-creación de base de datos

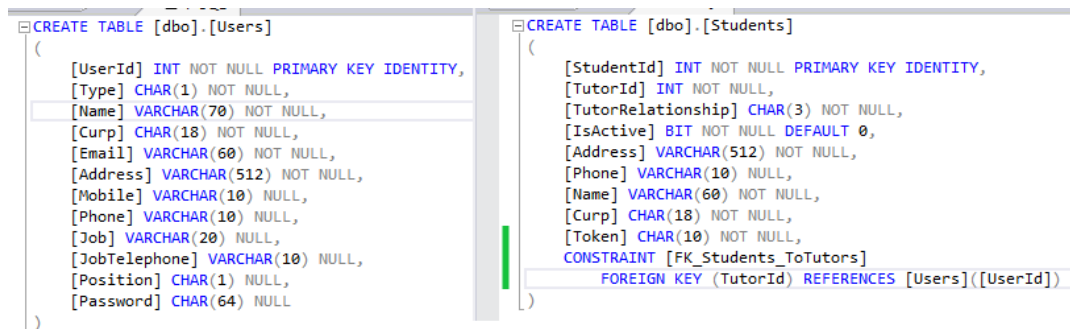
En el script de inserción de datos *Required.PostDeployment.sql*, que se ejecuta una vez que las tablas han sido creadas, se insertan dos registros importantes, el del usuario administrador, en este caso uno de los miembros del equipo, y un grupo comodín.

Para la publicación de la base de datos debemos desplegar el menú contextual del proyecto y elegir la opción *Publicar* de entre todas las que aparecen en él.



Elegimos el servidor y la base de datos creada en un paso anterior y ahí publicamos la base de datos, indicando el servidor y las credenciales para acceder con los permisos suficientes. Del entorno de desarrollo se obtiene un mensaje de indicando si la publicación fue exitosa.

A destacar la tabla de usuarios y la de estudiantes, los usuarios escolares tendrán una contraseña asignada mientras que los tutores, a pesar de tener una contraseña por ser usuarios, usarán un *token* específico para cada uno de sus tutorados. Tanto el *token* como la contraseña serán las credenciales de validación requeridas para efectuar acciones dentro del sistema.



```
CREATE TABLE [dbo].[Users]
(
  [UserId] INT NOT NULL PRIMARY KEY IDENTITY,
  [Type] CHAR(1) NOT NULL,
  [Name] VARCHAR(70) NOT NULL,
  [Curp] CHAR(18) NOT NULL,
  [Email] VARCHAR(60) NOT NULL,
  [Address] VARCHAR(512) NOT NULL,
  [Mobile] VARCHAR(10) NULL,
  [Phone] VARCHAR(10) NULL,
  [Job] VARCHAR(20) NULL,
  [JobTelephone] VARCHAR(10) NULL,
  [Position] CHAR(1) NULL,
  [Password] CHAR(64) NULL
)

CREATE TABLE [dbo].[Students]
(
  [StudentId] INT NOT NULL PRIMARY KEY IDENTITY,
  [TutorId] INT NOT NULL,
  [TutorRelationship] CHAR(3) NOT NULL,
  [IsActive] BIT NOT NULL DEFAULT 0,
  [Address] VARCHAR(512) NOT NULL,
  [Phone] VARCHAR(10) NULL,
  [Name] VARCHAR(60) NOT NULL,
  [Curp] CHAR(18) NOT NULL,
  [Token] CHAR(10) NOT NULL,
  CONSTRAINT [FK_Students_ToTutors]
  FOREIGN KEY (TutorId) REFERENCES [Users]([UserId])
)
```

Ilustración 125 Tablas Users y Students

Gestión de roles a nivel de la base de datos

Los roles de los usuarios dentro del sistema son tres: Tutores, Profesores y Administradores, para la gestión de ellos se utilizó el mismo framework de desarrollo que, auxiliado por los campos *Type* y *Position* de la tabla *Users*, permite controlar los niveles de autorización y las acciones que cada rol puede realizar.

Los valores posibles para la columna *Type* son: ‘T’ para un usuario tutor y ‘S’ para un usuario escolar, a su vez, el usuario es usuario escolar, deberá tener uno de los siguientes valores en la columna *Position*: ‘P’ para profesores y ‘A’ para administradores.

Conexión con la base de datos

Una vez que se ha publicado la base de datos en el servidor, se debe establecer la conexión con la aplicación web, para ello es necesario acceder al archivo `connections.config` localizado en la raíz del proyecto web y modificar la cadena de conexión para que haga referencia al servidor de bases de datos y a la base de datos creada. La cadena de conexión debe ser parecida a esta:

```
<add name="edutorDb"
connectionString="server=musx2mz8a7.database.windows.net,1433;database=dataedutor;uid=dbadmin;password=DataEdu70r:);" />
```

Donde:

- `"edutorDb"` es el nombre de la cadena, este debe ser siempre el mismo para el entorno productivo, también podemos crear otra cadena, con el nombre `"edutorDbTest"` que podemos usar para cuando se esté depurando el servicio.
- `musx2mz8a7.database.windows.net,1433` es la dirección del servidor y el puerto en a través del cual se establecerá la conexión. Estos valores pueden cambiar para cuando sea necesario hacer referencia a otro servidor.
- `dataedutor` es el nombre de la base de datos, este valor también puede cambiar de acuerdo a las necesidades que se tengan.
- `Dbadmin` y `DataEdu70r:)` son el nombre de usuario y la contraseña con las se realizará la conexión al servidor.

La conexión al servidor se verifica la primera vez que se pone en marcha el servicio web, en caso de ser inválida lo indicará con un mensaje de error y no permitirá la interacción hasta que dicho error sea corregido.

Publicación de la aplicación web

Del mismo modo que se realizó la publicación de la base de datos a través de la herramienta de desarrollo, se puede realizar la publicación del servicio web al servidor previamente configurado en la etapa de aprovisionamiento, para ello es necesario regresar al portal de administración de Azure y elegir nuestra aplicación web y descargar el perfil de publicación.



Ilustración 126 Obtención del perfil de publicación

El perfil de publicación es un archivo con extensión *.PublishSettings* que contiene la información necesaria para que Visual Studio pueda conectarse al servidor remoto y publicar la aplicación de manera sencilla. Para esto, se debe acceder al menú contextual del proyecto web, en este caso **Edutor.Web.Api**, y elegir la opción *Publicar*.

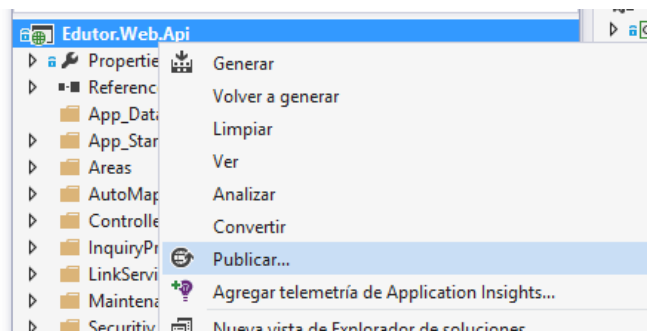


Ilustración 127 Publicación del proyecto de aplicación web

Tras lo cual aparecerá un cuadro de diálogo en el que se debe seleccionar *Perfil > Importar*, se debe navegar hasta el archivo descargado y elegirlo, al hacer esto se cargará la configuración necesaria para el despliegue de la aplicación web en el servidor remoto.

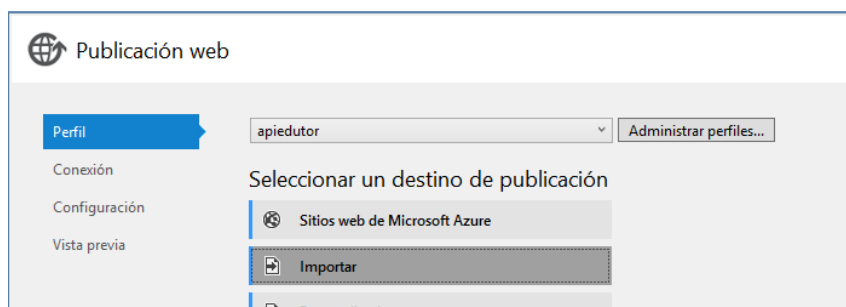


Ilustración 128 Importar perfil de publicación

Para confirmar que la publicación fue realizada correctamente un navegador se abrirá mostrando la aplicación web funcionando.

Resumen de implementación del sistema

A modo de resumen se presenta la siguiente lista de pasos:

1. Aprovisionar el servidor de base de datos usando el portal de Azure
2. Crear la base de datos en el servidor
3. Aprovisionar el servidor de aplicaciones web en el portal de Azure
4. Publicar en el servidor el proyecto de base de datos **Edutor.Db** utilizando Visual Studio
5. Establecer la cadena de conexión a la base de datos en el archivo *connections.config*
6. Publicar en el servidor de aplicaciones web el proyecto **Edutor.Web.Api**

Pruebas de la API de Edutor

Herramientas

Fiddler

Una de las herramientas empleadas para la etapa de pruebas fue la aplicación *Fiddler* que es un monitor y depurador de tráfico HTTP, el cual se usó para realizar peticiones directamente a través de la interfaz de la aplicación así como para capturar el tráfico entrante y saliente al servidor web que se usó durante la fase de desarrollo.

Peticiones a través de Fiddler

Se realizaron diversas peticiones individuales al servidor de pruebas empleando Fiddler. Para el caso de las peticiones a las que solo tiene acceso un administrador se usó el administrador que se inserta al realizar la instalación de Edutor en un servidor nuevo. Para otros casos se efectuaron las peticiones

Cuando la respuesta a la petición es muy larga, se optó por paginar la petición cuando se podía, cuando no hubo otra opción, se cortó a fin de hacer más simple la lectura.

Grupos

Consulta para obtener todos los grupos

```
GET http://api.edutor-tt.com/groups?pageNumber=1&pageSize=2 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Type: application/json
```

Respuesta

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 1059
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie: ARRAffinity=31abbedeb17be6e5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Domain=api.edutor-tt.com
Date: Tue, 02 Jun 2015 02:43:32 GMT
```

```
{"pageCount":8,"pageNumber":1,"pageSize":2,"items":[{"groupId":0,"name":"Plante1","subject":"Grupo por default","fromDate":"2015-01-
```

```
01T00:00:00", "toDate": null, "studentsCount": 0, "teachersCount": 0, "links": [{"rel": "self",
"href": "http://api.edutor-
tt.com/groups/0", "method": "GET"}, {"rel": "students", "href": "http://api.edutor-
tt.com/groups/0/students", "method": "GET"}, {"rel": "schoolusers", "href": "http://api.edut
or-
tt.com/groups/0/schoolusers", "method": "GET"}]}, {"groupId": 1, "name": "1L", "subject": "Gbo
ngan", "fromDate": "2015-01-
28T00:00:00", "toDate": null, "studentsCount": 8, "teachersCount": 1, "links": [{"rel": "self",
"href": "http://api.edutor-
tt.com/groups/1", "method": "GET"}, {"rel": "students", "href": "http://api.edutor-
tt.com/groups/1/students", "method": "GET"}, {"rel": "schoolusers", "href": "http://api.edut
or-
tt.com/groups/1/schoolusers", "method": "GET"}]}, {"links": [{"rel": "currentPage", "href": "
http://api.edutor-
tt.com/groups?pageNumber=1&pageSize=2", "method": "GET"}, {"rel": "nextPage", "href": "http:
//api.edutor-tt.com/groups?pageNumber=2&pageSize=2", "method": "GET"}]}]
```

Mediante esa petición se obtienen todos los grupos registrados, la petición únicamente puede ser accedida por un usuario administrador, se comprueba al realizar la misma petición pero sin credenciales:

Petición no autorizada

```
GET http://api.edutor-tt.com/groups?pageNumber=1&pageSize=2 HTTP/1.1
User-Agent: Edutor-Client/1.0
Host: api.edutor-tt.com
Content-Type: application/json
```

Respuesta a petición no autorizada

```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 61
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6be5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214; Path=/; Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 02:43:26 GMT
```

```
{"message": "Authorization has been denied for this request."}
```

Consulta para agregar un nuevo grupo

```
POST http://api.edutor-tt.com/groups?pageNumber=1&pageSize=2 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Type: application/json
Content-Length: 127
```

```
{
  "name": "4B",
  "subject": "Grupo de cuarto",
  "fromDate": "2015-01-28T00:00:00",
  "toDate": "2015-06-28T00:00:00"
}
```

Respuesta

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 418
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/groups/15
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 02:49:57 GMT

{"groupId":15,"name":"4B","subject":"Grupo de cuarto","fromDate":"2015-01-
28T00:00:00","toDate":"2015-06-
28T00:00:00","studentsCount":0,"teachersCount":0,"links":[{"rel":"self","href":"http://
/api.edutor-
tt.com/groups/15","method":"GET"}, {"rel":"students","href":"http://api.edutor-
tt.com/groups/15/students","method":"GET"}, {"rel":"schoolusers","href":"http://api.edu
tor-tt.com/groups/15/schoolusers","method":"GET"}]}
```

El código de estatus de la respuesta es 201, que significa que un recurso nuevo fue creado en el servidor, y es con el recurso nuevo con el que el servidor responde.

Petición no autorizada

Al igual que la consulta de grupos, el añadir un nuevo grupo también está restringido para administradores

```
POST http://api.edutor-tt.com/groups HTTP/1.1
User-Agent: Edutor-Client/1.0
Host: api.edutor-tt.com
Content-Type: application/json
Content-Length: 127

{
  "name": "4B",
  "subject": "Grupo de cuarto",
  "fromDate": "2015-01-28T00:00:00",
  "toDate": "2015-06-28T00:00:00"
}
```

La respuesta es un error 401, No autorizado

```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 61
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:00:16 GMT

{"message":"Authorization has been denied for this request."}
```

Petición con errores en el modelo

Se realizó la prueba de tratar de agregar un grupo sin nombre, cosa que viola nuestras reglas de negocio

```
POST http://api.edutor-tt.com/groups HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Type: application/json
Content-Length: 110
```

```
{
  "subject": "Grupo de cuarto",
  "fromDate": "2015-01-28T00:00:00",
  "toDate": "2015-06-28T00:00:00"
}
```

Se obtuvo como respuesta un código 400, que significa *Bad Request* (petición errónea), y el detalle de los errores que el sistema encontró en la petición.

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 59
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Domain=api.edutor-tt.com
Date: Tue, 02 Jun 2015 02:52:50 GMT
```

```
{"message":"newGroup.Name: The Name field is required. ;"}
```

Consulta para modificar un grupo

Para el caso, tratamos de modificar el grupo con identificador 15

```
PATCH http://api.edutor-tt.com/groups HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Type: application/json
Content-Length: 142
```

```
{
  "groupId":15,
  "name": "5B",
  "subject": "Grupo de cuarto",
  "fromDate": "2015-01-28T00:00:00",
  "toDate": "2015-06-28T00:00:00"
}
```

Una vez ejecutada la petición, la respuesta es la siguiente

HTTP/1.1 200 OK

```
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 418
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/groups/15
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=a37f273645ad4151f42d38dfbf0a608ddce6db3d0e8a15dbca8532a8cc6a23b2;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:11:41 GMT
```

```
{"groupId":15,"name":"5B","subject":"Grupo de cuarto","fromDate":"2015-01-
28T00:00:00","toDate":"2015-06-
28T00:00:00","studentsCount":0,"teachersCount":0,"links":[{"rel":"self","href":"http://
/api.edutor-
tt.com/groups/15","method":"GET"}, {"rel":"students","href":"http://api.edutor-
tt.com/groups/15/students","method":"GET"}, {"rel":"schoolusers","href":"http://api.edu
tor-tt.com/groups/15/schoolusers","method":"GET"}]}
```

Al igual que las peticiones anteriores, la modificación de grupos también está sujeta a autenticación y autorización, pero la petición no autorizada se omite por brevedad en el reporte.

Consulta para vincular un estudiante con un grupo

Supongamos que se desea vincular el estudiante con identificador 6 al grupo recién creado, el de 5B. La petición es la siguiente

```
PUT http://api.edutor-tt.com/groups/15/students/6 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Mientras que la respuesta es un código de estatus 204, *No Content* (Sin contenido), que indica una petición exitosa.

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:23:05 GMT
```

Consulta para vincular un usuario escolar con un grupo

Supongamos que se desea vincular el estudiante con identificador 6 al grupo que recién creado, el de 5B. La petición es la siguiente

```
PUT http://api.edutor-tt.com/groups/15/schoolusers/42 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Mientras que la respuestas es un código de éxito, pero que indica que la respuesta no tiene contenido

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6be5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:28:28 GMT
```

Consulta para conocer los alumnos inscritos a un grupo

Ahora se comprueba que en realidad el alumno y el profesor hayan sido añadidos al grupo, la petición para conocer los estudiantes de cierto grupo es:

```
PUT http://api.edutor-tt.com/groups/15/students HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Y la respuesta, la lista paginada en donde se encuentra el estudiante con el id igual a 6

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 522
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=a37f273645ad4151f42d38dfbf0a608ddce6db3d0e8a15dbca8532a8cc6a23b2;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:31:43 GMT
```

```
{"pageCount":1,"pageNumber":1,"pageSize":20,"items":[{"studentId":6,"tutorId":4,"tutor
Relationship":"unc","isActive":false,"address":"P.O. Box 349, 5540 Ornare
Road","phone":"7824543778","curp":"MVB433575YIVUC6IM","name":"Jane
Guzman","links":[{"rel":"self","href":"http://api.edutor-
tt.com/students/6","method":"GET"}, {"rel":"tutor","href":"http://api.edutor-
tt.com/tutors/4","method":"GET"}]}, {"rel":"currentPage","href":"http://api.e
dutor-tt.com/groups/15/students?pageNumber=1&pageSize=20","method":"GET"}]}
```

Para conocer los profesores, la petición es la siguiente

```
GET http://api.edutor-tt.com/groups/15/schoolusers HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUKxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Y la respuesta, en la que se puede encontrar el usuario escolar con id igual a 42

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 554
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:34:27 GMT

{"pageCount":1,"pageNumber":1,"pageSize":20,"items":[{"position":"P","userId":42,"name
":"Ul1a
Stout","curp":"ABKZ607668NVIQTL64","email":"penatibus.et@Fusce.org","address":"614-
9864 Metus
St.","mobile":"0259591233","phone":"2269749034","type":"S","links":[{"rel":"self","hre
f":"http://api.edutor-
tt.com/schoolusers/42","method":"GET"},{"rel":"groups","href":"http://api.edutor-
tt.com/schoolusers/42/groups","method":"GET"}]},{"rel":"currentPage","href":
"http://api.edutor-
tt.com/groups/15/schoolusers?pageNumber=1&pageSize=20","method":"GET"}]}
```

Consulta para eliminar un estudiante o un profesor de un grupo

La petición es similar a la anterior, únicamente cambia el verbo HTTP utilizado

```
DELETE http://api.edutor-tt.com/groups/15/students/6 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUKxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

La respuesta es un código que indica que la petición se realizó con éxito

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=a37f273645ad4151f42d38dfbf0a608ddce6db3d0e8a15dbca8532a8cc6a23b2;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:38:19 GMT
```

De igual manera, esta es la petición que se realiza para eliminar un profesor de un grupo

```
DELETE http://api.edutor-tt.com/groups/15/schoolusers/42 HTTP/1.1
User-Agent: Edutor-Client/1.0
```

```
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Que devuelve el código de No Contenido, que indica una petición exitosa

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:42:46 GMT
```

Consulta para eliminar un grupo del sistema

Para eliminar un grupo, es necesario realizar una petición como la siguiente

```
DELETE http://api.edutor-tt.com/groups/15 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

El resultado es:

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=a37f273645ad4151f42d38dfbf0a608ddce6db3d0e8a15dbca8532a8cc6a23b2;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:44:31 GMT
```

Petición para eliminar un grupo con relaciones a otras entidades

Cuando se trata de eliminar un grupo que tiene alumnos, profesores u otro tipo de relación con otros elementos dentro del sistema, la API debe responder con un código de error, tal es el caso del grupo con id 5, al intentar eliminarlo con la siguiente petición

```
DELETE http://api.edutor-tt.com/groups/5 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Se obtiene como respuesta un código 409, que indica que hubo un conflicto al procesar la petición.

```
HTTP/1.1 409 Conflict
```



```
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 80
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=a37f273645ad4151f42d38dfbf0a608ddce6db3d0e8a15dbca8532a8cc6a23b2;Path=/;Domain=
api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:46:58 GMT
```

```
{"message":"El usuario / estudiante no puede ser eliminado, está en conflicto"}
```

Se corrobora su existencia al consultar la API mediante la siguiente petición:

```
DELETE http://api.edutor-tt.com/groups/5 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
Host: api.edutor-tt.com
Content-Length: 0
```

Que devuelve

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 396
Content-Type: application/json; charset=utf-8
Expires: -1
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
Date: Tue, 02 Jun 2015 03:52:28 GMT
```

```
{"groupId":5,"name":"3G","subject":"Rivire-du-Loup","fromDate":"2015-05-
05T00:00:00","toDate":null,"studentsCount":6,"teachersCount":5,"links":[{"rel":"self",
"href":"http://api.edutor-
tt.com/groups/5","method":"GET"},{"rel":"students","href":"http://api.edutor-
tt.com/groups/5/students","method":"GET"},{"rel":"schoolusers","href":"http://api.edut
or-tt.com/groups/5/schoolusers","method":"GET"}]}
```

De igual manera, el recurso de eliminar es accesible solo para administradores, se comprueba realizando una petición con credenciales de tutor:

```
DELETE http://api.edutor-tt.com/groups/5 HTTP/1.1
User-Agent: Edutor-Client/1.0
Authorization: Basic VDPFMjYwNDl1Njkx
Host: api.edutor-tt.com
```

A la que se obtiene el código de no autorizado:

```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 0
Expires: -1
WWW-Authenticate: basic
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=31abbedeb17be6ebe5d04e4fcd902071de2a016a311f0d7dd61e5a62cdb7c214;Path=/;Do
main=api.edutor-tt.com
```

Date: Tue, 02 Jun 2015 03:54:37 GMT

Estudiantes

Consulta para activar un estudiante

POST http://api.edutor-tt.com/students/E1717bb1aa/activate HTTP/1.1
User-Agent: Edutor-Client/1.0
Host: api.edutor-tt.com
Content-Length: 0
Connection: Keep-Alive

Respuesta

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 339
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=af2e235b27316233eee3bf0f7a4ae44a38028b89253bac6488bafce70348a9b3;Path=/;Do
main=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:38:18 GMT

```
{"studentId":1,"tutorId":2,"tutorRelationship":"aun","isActive":true,"address":"Ap  
#388-1548 Neque Ave","phone":"8816363220","curp":"EFUN614645IWRX8GJ","name":"Phelan  
H. Dillard","links":[{"rel":"self","href":"http://api.edutor-  
tt.com/students/1","method":"GET"}, {"rel":"tutor","href":"http://api.edutor-  
tt.com/tutors/2","method":"GET"}]}
```

Consulta para agregar un estudiante

POST http://api.edutor-tt.com/students HTTP/1.1
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 142
Expect: 100-continue

```
{"TutorId":39,"TutorRelationship":"mom","Address":"193-5649 Sed  
St.,"Phone":"1559287690","Curp":"IONK373817PGJKD2GZ","Name":"Nero A. Phelps"}
```

Respuesta

HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 335
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/students/92
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f;Path=/;Do
main=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:06:23 GMT

```
{"studentId":92,"tutorId":39,"tutorRelationship":"mom","isActive":false,"address":"193-5649 Sed St.,"phone":"1559287690","curp":"IONK373817PGJKD2GZ","name":"Nero A. Phelps","links":[{"rel":"self","href":"http://api.edutor-tt.com/students/92","method":"GET"}, {"rel":"tutor","href":"http://api.edutor-tt.com/tutors/39","method":"GET"}]}
```

Tutores

Consulta para agregar un tutor

```
POST http://api.edutor-tt.com/tutors HTTP/1.1
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 193
Expect: 100-continue
```

```
{"Job":"","JobTelephone":null,"Name":"Mannix E. Burgess","Curp":"MPKU117517SSHUXN51","Email":"diam.nunc@Quisquetincidunt.com","Address":"Ap #826-8094 Ut Rd.,"Mobile":"0026534883","Phone":null}
```

Respuesta

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 387
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/tutors/22
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f; Path=/; Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:02:50 GMT
```

```
{"job":"","jobTelephone":null,"userId":22,"name":"Mannix E. Burgess","curp":"MPKU117517SSHUXN51","email":"diam.nunc@Quisquetincidunt.com","address":"Ap #826-8094 Ut Rd.,"mobile":"0026534883","phone":null,"type":"T","links":[{"rel":"self","href":"http://api.edutor-tt.com/tutors/22","method":"GET"}, {"rel":"students","href":"http://api.edutor-tt.com/tutors/22/students","method":"GET"}]}
```

Usuarios escolares

Consulta para agregar un usuario escolar

```
POST http://api.edutor-tt.com/schoolusers HTTP/1.1
Authorization: Basic UzpGRUJBOTExMjA2SERGUkxOMDk6MQ==
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 184
Expect: 100-continue
```

```
{"Position":"A","Name":"Lane Ayers","Curp":"TPIG882507TQQPLA47","Email":"semper@interdumliberoduo.edu","Address":"P.O. Box 511, 8806 In Ave","Mobile":"0085065265","Phone":"8859033169"}
```

Respuesta

HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 387
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/schoolusers/102
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f; Path=/; Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:09:05 GMT

```
{"position": "A", "userId": 102, "name": "Lane  
Ayers", "curp": "TPIG882507TQQPLA47", "email": "semper@interdumliberoduo.edu", "address": "P.O.  
Box 511, 8806 In  
Ave", "mobile": "0085065265", "phone": "8859033169", "type": "S", "links": [{"rel": "self", "href": "h  
ttp://api.edutor-  
tt.com/schoolusers/102", "method": "GET"}, {"rel": "groups", "href": "http://api.edutor-  
tt.com/schoolusers/102/groups", "method": "GET"}]}
```

Eventos

Consulta para agregar un evento

POST http://api.edutor-tt.com/events HTTP/1.1
Authorization: Basic UzpET1JENjI0NjEwUWVparVZOOTk6NTA=
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 403
Expect: 100-continue

```
{"SchoolUserId": 50, "GroupId": 2, "Name": "Muerte de Frida Khalo", "Date": "2015-07-  
08T00:00:00", "Description": "Frida Kahlo murió en Coyoacán el 13 de julio de 1954. No  
se realizó ninguna autopsia. Sus restos fueron velados en el Palacio de Bellas Artes  
de la Ciudad de México y se cubrió su féretro con la bandera del Partido Comunista  
Mexicano, un hecho que la prensa nacional criticó profusamente."}
```

Respuesta

HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 782
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/events/7
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f; Path=/; Do
main=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:14:26 GMT

```
{"eventId": 7, "groupId": 2, "schoolUserId": 50, "name": "Muerte de Frida  
Khalo", "date": "2015-07-08T00:00:00", "creationDate": "2015-06-  
21T21:14:26.9155003Z", "description": "Frida Kahlo murió en Coyoacán el 13 de julio de  
1954. No se realizó ninguna autopsia. Sus restos fueron velados en el Palacio de  
Bellas Artes de la Ciudad de México y se cubrió su féretro con la bandera del Partido  
Comunista Mexicano, un hecho que la prensa nacional criticó  
profusamente.", "attendeesComing": 0, "attendeesNotComing": 0, "attendeesNoAnswer": 0, "links  
": [{"rel": "self", "href": "http://api.edutor-  
tt.com/events/7", "method": "GET"}, {"rel": "attendees", "href": "http://api.edutor-
```

```
tt.com/events/7/attendees", "method": "GET"}, {"rel": "schooluser", "href": "http://api.edutor-tt.com/schoolusers/50", "method": "GET"}]]}
```

Consulta para confirmar la asistencia a un evento

```
PUT http://api.edutor-tt.com/events/7/attendees/1 HTTP/1.1
Authorization: Basic VDPfMTCxN2JiMWFh
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 39
Expect: 100-continue
Connection: Keep-Alive
```

```
{"Rsvp":true,"StudentId":1,"EventId":7}
```

Respuesta

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f;Path=/;Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:01:13 GMT
```

Notificaciones

Consulta para agregar una notificación

```
POST http://api.edutor-tt.com/notifications HTTP/1.1
Authorization: Basic UzpDSUJZNDU3MTk4WVRXS1JVNju6NDc=
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 101
Expect: 100-continue
```

```
{"SchoolUserId":47,"GroupId":14,"Text":"Les pido de favor que envíen el dinero para los materiales"}
```

Respuesta

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 424
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/notifications/24
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f;Path=/;Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:18:13 GMT
```

```
{"notificationId":24,"schoolUserId":47,"groupId":14,"text":"Les pido de favor que envíen el dinero para los materiales","totalStudents":0,"seenStudents":0,"links":[{"rel":"self","href":"http://api.edutor-tt.com/notifications/24","method":"GET"}, {"rel":"details","href":"http://api.edutor-tt.com/notifications/24/details","method":"GET"}, {"rel":"schooluser","href":"http://api.edutor-tt.com/schoolusers/47","method":"GET"}]}
```

Consulta para marcar una notificación como vista

```
PUT http://api.edutor-tt.com/notifications/2/details/1 HTTP/1.1
Authorization: Basic VDPfMTcxN2JiMWFh
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 34
Expect: 100-continue
Connection: Keep-Alive
```

```
{"StudentId":1,"NotificationId":2}
```

Respuesta

```
HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=af2e235b27316233eee3bf0f7a4ae44a38028b89253bac6488bafce70348a9b3;Path=/;Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:01:13 GMT
```

Preguntas

Consulta para agregar una pregunta

```
POST http://api.edutor-tt.com/questions HTTP/1.1
Authorization: Basic UzpFRVFLNjUxODQ4RUpPTFFIMjA6NTM=
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 163
Expect: 100-continue
```

```
{"SchoolUserId":53,"GroupId":13,"Text":"¿Le pareció complicada la tarea de español?","ExpirationDate":"2015-07-14T00:00:00-05:00","PossibleAnswers":["Si","No"]}
```

Respuesta

```
HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 448
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/questions/9
Server: Microsoft-IIS/8.0
```

X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=falbd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f;Path=/;Do
main=api.edutor-tt.com
Date: Sun, 21 Jun 2015 21:18:16 GMT

```
{"schoolUserId":53,"groupId":13,"questionId":9,"text":"¿Le pareció complicada la tarea de español?","expirationDate":"2015-07-14T05:00:00+00:00","possibleAnswers":[],"totalAnswerCount":0,"links":[{"rel":"self","href":"http://api.edutor-tt.com/questions/9","method":"GET"}, {"rel":"answers","href":"http://api.edutor-tt.com/questions/9/answers","method":"GET"}, {"rel":"schooluser","href":"http://api.edutor-tt.com/schoolusers/53","method":"GET"}]}
```

Consulta para responder a una pregunta

PUT http://api.edutor-tt.com/questions/5/answers/1 HTTP/1.1
Authorization: Basic VDPFMTcXN2JiMWFh
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 51
Expect: 100-continue
Connection: Keep-Alive

```
{"SelectedAnswerId":2,"StudentId":1,"QuestionId":5}
```

Respuesta

HTTP/1.1 204 No Content
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=af2e235b27316233eee3bf0f7a4ae44a38028b89253bac6488bafce70348a9b3;Path=/;Do
main=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:01:12 GMT

Conversaciones

Consulta para agregar un mensaje como tutor

POST http://api.edutor-tt.com/conversations HTTP/1.1
Authorization: Basic VDPFMTcXN2JiMWFh
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 39
Expect: 100-continue

```
{"ToId":62,"FromId":2,"Text":"Hola :)"}}
```

Respuesta

HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache

Content-Length: 319
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/conversations/1/messages/2
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f;Path=/;Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:17:05 GMT

```
{"conversationId":1,"messageId":2,"toId":62,"fromId":2,"text":"Hola :)", "sentDate":"2015-06-21T22:17:06.9108727Z","seenDate":null,"links":[{"rel":"self","href":"http://api.edutor-tt.com/conversations/1/messages/2","method":"GET"}, {"rel":"conversation","href":"http://api.edutor-tt.com/conversations/1","method":"GET"}]}
```

Consulta para agregar un mensaje como usuario escolar

POST http://api.edutor-tt.com/conversations HTTP/1.1
Authorization: Basic UzpKTktFOTY4NTI4T0VNWllNMzg6NjI=
User-Agent: Edutor-Client/1.0
Content-Type: application/json; charset=utf-8
Host: api.edutor-tt.com
Content-Length: 39
Expect: 100-continue
Connection: Keep-Alive

```
{"ToId":2,"FromId":62,"Text":"Hola :)"}
```

Respuesta

HTTP/1.1 201 Created
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 319
Content-Type: application/json; charset=utf-8
Expires: -1
Location: http://api.edutor-tt.com/conversations/1/messages/1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=af2e235b27316233eee3bf0f7a4ae44a38028b89253bac6488bafce70348a9b3;Path=/;Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:17:04 GMT

```
{"conversationId":1,"messageId":1,"toId":2,"fromId":62,"text":"Hola :)", "sentDate":"2015-06-21T22:17:06.3820702Z","seenDate":null,"links":[{"rel":"self","href":"http://api.edutor-tt.com/conversations/1/messages/1","method":"GET"}, {"rel":"conversation","href":"http://api.edutor-tt.com/conversations/1","method":"GET"}]}
```

Consulta para obtener una conversación

GET http://api.edutor-tt.com/conversations/1 HTTP/1.1
Authorization: Basic VDpFMTCxN2JiMWFh
User-Agent: Edutor-Client/1.0
Host: api.edutor-tt.com

Respuesta

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 939
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=af2e235b27316233eee3bf0f7a4ae44a38028b89253bac6488bafce70348a9b3;Path=/;Domain=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:17:05 GMT
```

```
{
  "conversationId":1,"senderId":62,"senderName":"Cassidy O.
  Gutierrez","recipientId":2,"recipientName":"Cairo
  Chaney","lastMessages":[{"conversationId":1,"messageId":1,"toId":2,"fromId":62,"text":
  "Hola :)", "sentDate":"2015-06-
  21T22:17:06","seenDate":null,"links":[{"rel":"self","href":"http://api.edutor-
  tt.com/conversations/1/messages/1","method":"GET"}, {"rel":"conversation","href":"http:
  //api.edutor-
  tt.com/conversations/1","method":"GET"}]}, {"conversationId":1,"messageId":2,"toId":62,
  "fromId":2,"text":"Hola :)", "sentDate":"2015-06-
  21T22:17:06","seenDate":null,"links":[{"rel":"self","href":"http://api.edutor-
  tt.com/conversations/1/messages/2","method":"GET"}, {"rel":"conversation","href":"http:
  //api.edutor-
  tt.com/conversations/1","method":"GET"}]}], "links":[{"rel":"self","href":"http://api.e
  dutor-
  tt.com/conversations/1","method":"GET"}, {"rel":"messages","href":"http://api.edutor-
  tt.com/conversations/1/messages","method":"GET"}]}
```

Consulta para obtener los mensajes de una conversación

```
GET http://api.edutor-tt.com/conversations/1/messages HTTP/1.1
Authorization: Basic UzpKtktFOTY4NTI4T0VNWl1lNMzg6NjI=
User-Agent: Edutor-Client/1.0
Host: api.edutor-tt.com
```

Respuesta

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 807
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie:
ARRAffinity=fa1bd11a4cd6e4f88ba526273ec0ad27734e9f8d6a4ea5d094811f39dec6518f;Path=/;Do
main=api.edutor-tt.com
Date: Sun, 21 Jun 2015 22:17:05 GMT
```

```
{
  "pageCount":1,"pageNumber":1,"pageSize":20,"items":[{"conversationId":1,"messageId":1
  ,"toId":2,"fromId":62,"text":"Hola :)", "sentDate":"2015-06-
  21T22:17:06","seenDate":null,"links":[{"rel":"self","href":"http://api.edutor-
  tt.com/conversations/1/messages/1","method":"GET"}, {"rel":"conversation","href":"http:
  //api.edutor-
  tt.com/conversations/1","method":"GET"}]}, {"conversationId":1,"messageId":2,"toId":62,
  "fromId":2,"text":"Hola :)", "sentDate":"2015-06-
  21T22:17:06","seenDate":null,"links":[{"rel":"self","href":"http://api.edutor-
  tt.com/conversations/1/messages/2","method":"GET"}, {"rel":"conversation","href":"http:
  //api.edutor-
  tt.com/conversations/1","method":"GET"}]}], "links":[{"rel":"currentPage","href":"http:
  //api.edutor-
  tt.com/conversations/1/messages?pageNumber=1&pageSize=20","method":"GET"}]}
```

Aplicación administrativa

Se creó una aplicación que sirve para dos propósitos: el primero, poder realizar peticiones a la API de manera automatizada y empleando el componente cliente previamente desarrollado. Y el segundo: ofrecer de una manera sencilla de realizar la carga de los registros de la escuela.

La ejecución de esta aplicación requiere la existencia de múltiples archivos tabulares delimitados los valores con un *pipe* '|', que contengan la información que se desea cargar. Se describen los archivos a continuación:

- **UsuariosEscolares.csv:** este almacena la información de los usuarios escolares (profesores y administradores). Descripción: curp del usuario|nombre completo|dirección|correo electrónico|teléfono de casa|teléfono móvil|posición dentro de la escuela

Ejemplo: El registro para el **Administrador, Antonio Feregrino Bolaños**, con CURP **FEBA911206HDFRLN07** cuya dirección es **Av. Juan de Dios Bátiz esq. Av. Miguel Othón de Mendizábal, Col. Lindavista**, correo electrónico **direccion_escom@ipn.mx** y números telefónicos **57296001** y **57296000**.

```
FEBA911206HDFRLN07|Antonio Feregrino Bolaños|Av. Juan de Dios Bátiz esq.  
Av. Miguel Othón de Mendizábal, Col.  
Lindavista|direccion_escom@ipn.mx|57296001|57296000|A
```

- **Tutores.csv:** este almacena la información de los tutores de los alumnos pertenecientes a la escuela. Descripción: curp del tutor|nombre completo|dirección |correo electrónico|teléfono de casa|teléfono móvil|nombre de trabajo|teléfono de trabajo

Ejemplo: El registro para el **Tutor Zenon Uribe Sanjuas**, con CURP **UISZ570915HDGRNN04** cuya dirección es **Nezahualcoyotl 109 Piso 8, Centro, 06082 México, D.F.** con correo electrónico **zenon1098@hotmail.com**, de profesión **Astrónomo** y números telefónicos **57296003**, **57296004** y **57296005**

UISZ570915HDGRNN04|Zenon Uribe Sanjuas|Nezahualcoyotl 109 Piso 8,
Centro, 06082 México,
D.F.|zenon1098@hotmail.com|57296003|57296004|Astrónomo|57296005

- **Alumnos.csv:** almacena la información de los alumnos pertenecientes a la escuela. Descripción: curp del tutor|curp del estudiante|nombre completo|relación con el tutor|dirección|teléfono de casa

Ejemplo: El registro para el **Estudiante Ununquadio Uribe Rodríguez**, con CURP **UIRU040315HASRDN01**, cuya dirección es **Nezahualcoyotl 109 Piso 8, Centro, 06082 México, D.F.** con número telefónico **57296005**, su tutor es su **padre** de CURP **UISZ570915HDGRNN04**

UISZ570915HDGRNN04|UIRU040315HASRDN01|Ununquadio Uribe
Rodríguez|pad|Nezahualcoyotl 109 Piso 8, Centro, 06082 México,
D.F.|57296005

- **Grupos.csv:** Almacena la información de los grupos que se registrarán para el periodo escolar. Descripción: nombre del grupo|descripción del grupo|fecha en la que inicia el grupo

Ejemplo: El registro para el grupo **3B**, de la asignatura **Español**, que comienza el día **2 de agosto de 2015**

3B|Español|2015-08-02

- **GruposAlumnos.csv:** Almacena la información correspondiente al vínculo 'pertenece al grupo'. Descripción: nombre del grupo|curp del estudiante

3B|UIRU040315HASRDN01

- **GruposProfesores.csv:** Almacena la información correspondiente al vínculo 'da clases en el grupo'

Los archivos anteriores deben ser almacenados en la misma carpeta con permisos de lectura dentro de la computadora en la que se ejecuta la aplicación, ya que esta lee directamente los archivos del sistema de archivos.

La aplicación administrativa tiene la limitante de tener que ser ejecutada una sola vez por cada carga de datos que se desee hacer, y esta debe ser en una instalación nueva de Edutor, de otro modo fallaría la ejecución con resultados inesperados.

Una vez ejecutada nos solicita ingresar la ruta de la carpeta en la que se guardaron los archivos con los datos a subir.



Ilustración 129 Solicitud de la ruta de los archivos para carga

Una vez que la ruta ha sido escrita, la aplicación procede con la inserción de registros, comenzando por los usuarios escolares

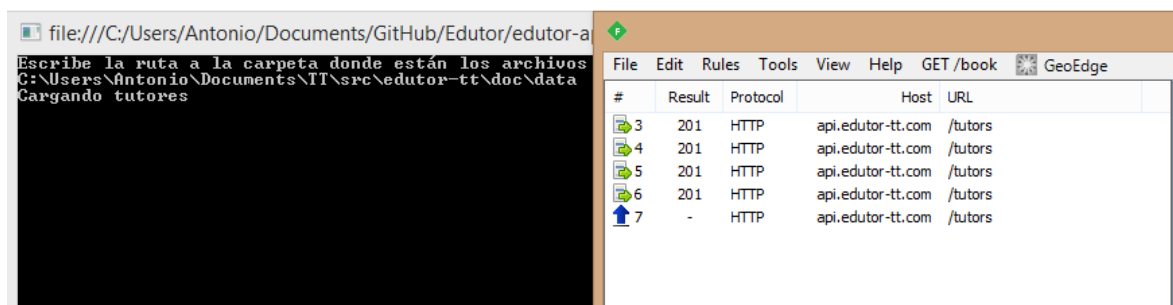


Ilustración 130 Peticiones al extremo Tutores

Usamos Fiddler para ver las peticiones que se realizan al servidor mientras se ejecuta la aplicación, en la imagen anterior se observa que se han realizado 4 peticiones exitosas, las 4 han regresado con un código de estatus 201, que indica que un recurso ha sido creado. Para la prueba se insertaron 40 tutores.

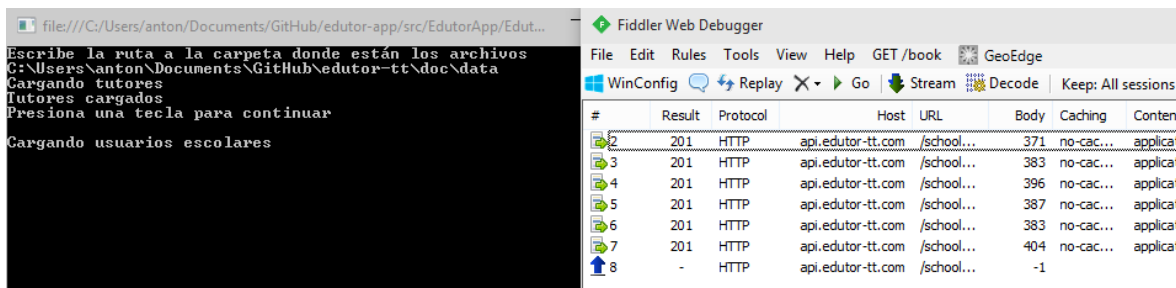


Ilustración 131 Peticiones al extremo SchoolUsers

Lo mismo aplica para las siguientes inserciones, como la de usuarios escolares, de los cuales se insertan 60, 40 profesores y 20 administrativos.

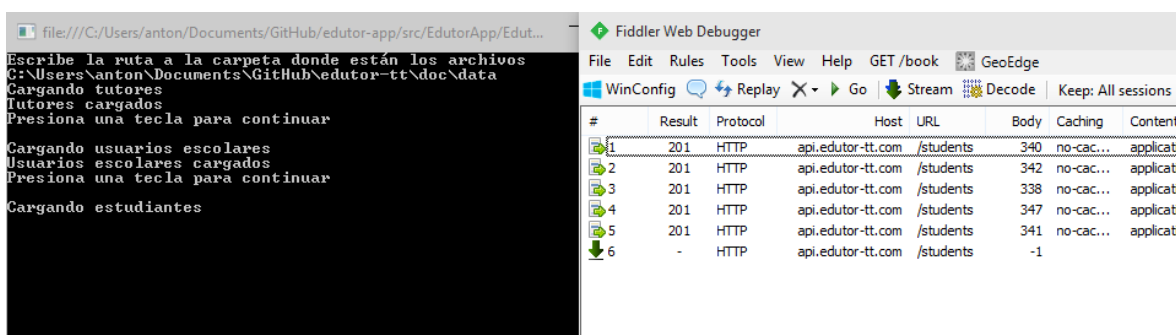


Ilustración 132 Peticiones al extremo Students

Para el caso de los estudiantes se hicieron 100 peticiones POST, y todas volvieron con el código 201. Posteriormente se insertan los grupos, para la prueba se emplearon 15 grupos

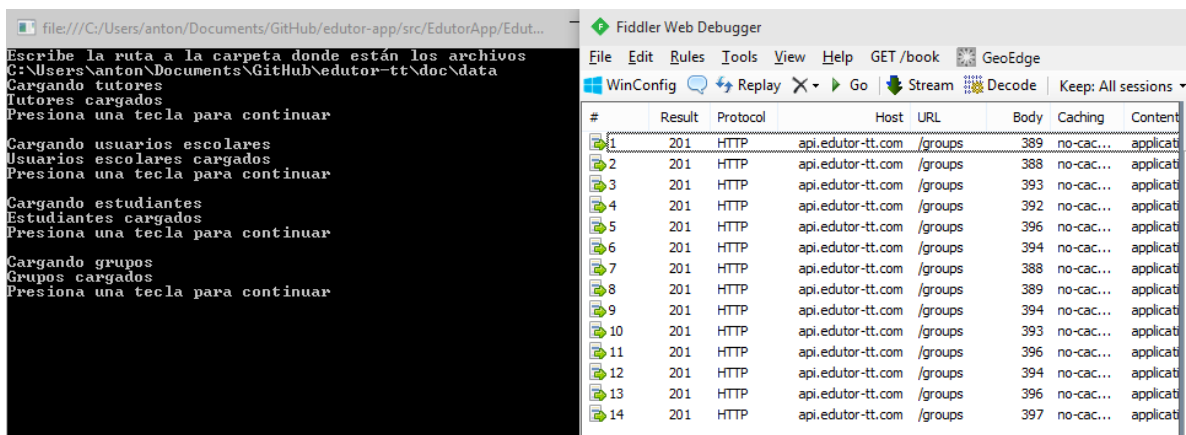


Ilustración 133 Peticiones al extremo Groups

Una parte vital es la creación de las relaciones grupo-profesor y grupo-alumnos, la aplicación también crea estas relaciones

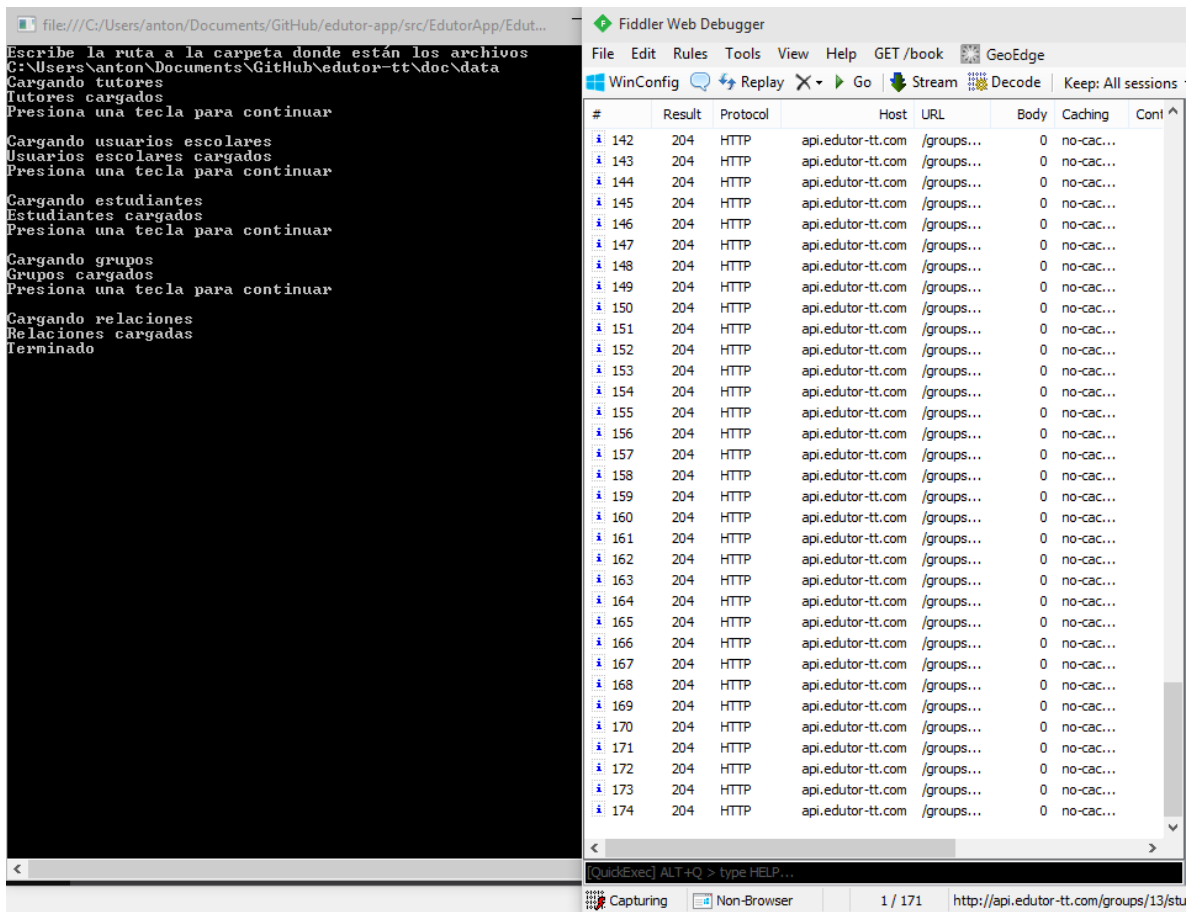


Ilustración 134 Peticiones al extremo Groups para vincular alumnos

En este caso, y según lo definido en la documentación de la API, la petición exitosa regresa con un código de estatus 204, que significa que ha sido completada con éxito pero que no tiene nada que devolver como respuesta.

La aplicación ejecuta las peticiones de manera secuencial, es decir, primero ejecutará la carga de tutores, profesores, estudiantes, grupos y las relaciones. Adicionalmente, la aplicación administrativa permite la carga de datos por default, utilizados por las apps demos para mostrar datos cargados en la base de datos. Para dicha carga también se ayuda de archivos similares a los descritos anteriormente, los archivos se describen a continuación:

- **Eventos.csv:** Este archivo almacena los eventos de prueba. Descripción: nombre del grupo|curp del usuario que lo organiza|fecha en la que realizará el evento|nombre del evento|descripción del evento

Ejemplo: El evento **Creación de Petróleos Mexicanos**, y su descripción **El 7 de junio de 1938, mediante Decreto del Congreso de la Unión, se creó Petróleos Mexicanos...** que se llevará a cabo el día **19 de Julio de 2015** será organizado por el profesor de CURP **JNKE968528OEMZYM38** para el grupo **1L**.

```
1L|JNKE968528OEMZYM38|2015-07-19|Creación de petróleos mexicanos|El 7 de junio de 1938, mediante Decreto del Congreso de la Unión, se creó Petróleos Mexicanos y comenzó a operar a partir del 20 de julio del mismo año, para ser la única compañía que pudiera explotar y administrar los yacimientos de petróleo encontrados en el territorio mexicano.
```

- **Notificaciones.csv:** Este archivo almacena las notificaciones de prueba. Descripción: nombre del grupo|curp del usuario que la envía|texto de la notificación

Ejemplo: La notificación **El día de mañana no habrá clases del grupo 3B** para el grupo **3B** creada por el usuario con CURP **JNKE968528OEMZYM38**

```
3B|JNKE968528OEMZYM38|Tarea de español para mañana
```

- **Preguntas.csv:** Este archivo almacena las preguntas de prueba. Descripción: nombre del grupo|curp del usuario que realiza la pregunta|texto de la pregunta|número de respuestas predefinidas|respuesta predefinida1|[respuesta predefinida2]...

Ejemplo: La pregunta **¿Tuvieron dificultad para resolver la tarea de matemáticas?** Con **3** respuestas predefinidas (**Sí, No, No se realizó**) para el grupo **3B** del usuario con CURP **JNKE968528OEMZYM38**

```
3B|JNKE968528OEMZYM38|¿Tuvieron dificultad para resolver la tarea de matemáticas?|3|Si|No|No se realizó
```

Los archivos de datos de prueba se deben almacenar en la misma carpeta que los de administración. La aplicación preguntará si es que se desea cargar los datos de prueba de los archivos o terminar la ejecución de las inserciones sin agregar más datos.


```
file:///C:/Users/Antonio/Documents/GitHub/Edutor/edutor-app/src/EdutorApp...
Escribe la ruta a la carpeta donde están los archivos
C:\Users\Antonio\Documents\IT\src\edutor-tt\doc\data
Cargando tutores
Tutores cargados
Presiona una tecla para continuar

Cargando usuarios escolares
Usuarios escolares cargados
Presiona una tecla para continuar

Cargando estudiantes
Estudiantes cargados
Presiona una tecla para continuar

Cargando grupos
Grupos cargados
Presiona una tecla para continuar

Cargando relaciones
Relaciones cargadas
¿Deseas cargar los datos de muestra? (s/n)
_
```

Ilustración 135 Solicitud para cargar datos de muestra

De elegir sí, continúa agregando los eventos demo. Algo interesante a notar es que para esta etapa, el programa de pureabas comienza a realizar peticiones simultáneas a nombre de cada profesor, esto se debe a que tiene que usar las credenciales específicas de cada uno para acceder a los extremos de la API, las capturas siguientes muestran lo dicho:

file:///C:/Users/Antonio/Documents/GitHub/Edutor/edutor-app/src

```

Escribe la ruta a la carpeta donde están los archivos
C:\Users\Antonio\Documents\IT\src\edutor-tt\doc\data
Cargando tutores
Tutores cargados
Presiona una tecla para continuar

Cargando usuarios escolares
Usuarios escolares cargados
Presiona una tecla para continuar

Cargando estudiantes
Estudiantes cargados
Presiona una tecla para continuar

Cargando grupos
Grupos cargados
Presiona una tecla para continuar

Cargando relaciones
Relaciones cargadas
¿Deseas cargar los datos de muestra? (s/n)
s
Cargando eventos
Events for: ABKZ607668NUIQTL64
Events for: CIBY457198YTWKRU65
Events for: DNEY145989MBPSYA98
Events for: DORD624610QZZEUN99
Events for: DPUT562123IZFUWF82
Events for: EEQR651848EJOLQH20
Events for: GLBQ347762JXUITH87
Events for: JJUE1362740LPMEL24
Events for: JNKE9685280EMZYM38

```

#	Result	Protocol	Host	URL
1	-	HTTP	api.edutor-tt.com	/events
2	-	HTTP	api.edutor-tt.com	/events
3	-	HTTP	api.edutor-tt.com	/events
4	-	HTTP	api.edutor-tt.com	/events
5	-	HTTP	api.edutor-tt.com	/events
6	-	HTTP	api.edutor-tt.com	/events
7	-	HTTP	api.edutor-tt.com	/events
8	-	HTTP	api.edutor-tt.com	/events
9	-	HTTP	api.edutor-tt.com	/events

Ilustración 136 Peticiones al extremo Events

```

Events for: EEQR651848EJOLQH20
Events for: GLBQ347762JXUITH87
Events for: JJUE1362740LPMEL24
Events for: JNKE9685280EMZYM38
Eventos cargados

Cargando notificaciones
Notifications for: ABKZ607668NUIQTL64
Notifications for: CIBY457198YTWKRU65
Notifications for: DNEY145989MBPSYA98
Notifications for: JNKE9685280EMZYM38
Notifications for: DORD624610QZZEUN99
Notifications for: DPUT562123IZFUWF82
Notifications for: EEQR651848EJOLQH20
Notifications for: GLBQ347762JXUITH87
Notifications for: JJUE1362740LPMEL24

```

18	201	HTTP	api.edutor-tt.com	/events
19	201	HTTP	api.edutor-tt.com	/events
20	-	HTTP	api.edutor-tt.com	/notifications
21	-	HTTP	api.edutor-tt.com	/notifications
22	-	HTTP	api.edutor-tt.com	/notifications
23	-	HTTP	api.edutor-tt.com	/notifications
24	-	HTTP	api.edutor-tt.com	/notifications
25	-	HTTP	api.edutor-tt.com	/notifications
26	-	HTTP	api.edutor-tt.com	/notifications
27	-	HTTP	api.edutor-tt.com	/notifications
28	-	HTTP	api.edutor-tt.com	/notifications

Ilustración 137 Peticiones al extremo Notifications

```

Notifications for: GLBQ347762JXUITH87
Notifications for: JJUE1362740LPMEL24
Eventos cargados

Cargando preguntas
Questions for: ABKZ607668NUIQTL64
Questions for: CIBY457198YTWKRU65
Questions for: DNEY145989MBPSYA98
Questions for: JNKE9685280EMZYM38
Questions for: DORD624610QZZEUN99
Questions for: DPUT562123IZFUWF82
Questions for: EEQR651848EJOLQH20
Questions for: GLBQ347762JXUITH87
Questions for: JJUE1362740LPMEL24
Preguntas cargadas
Terminado

```

58	201	HTTP	api.edutor-tt.com	/questions
59	201	HTTP	api.edutor-tt.com	/questions
60	201	HTTP	api.edutor-tt.com	/questions
61	201	HTTP	api.edutor-tt.com	/questions
62	201	HTTP	api.edutor-tt.com	/questions
63	201	HTTP	api.edutor-tt.com	/questions
64	201	HTTP	api.edutor-tt.com	/questions
65	201	HTTP	api.edutor-tt.com	/questions
66	201	HTTP	api.edutor-tt.com	/questions
67	201	HTTP	api.edutor-tt.com	/questions
68	201	HTTP	api.edutor-tt.com	/questions
69	201	HTTP	api.edutor-tt.com	/questions
70	201	HTTP	api.edutor-tt.com	/questions
71	201	HTTP	api.edutor-tt.com	/questions
72	201	HTTP	api.edutor-tt.com	/questions

Ilustración 138 Peticiones al extremo Questions

Resultados

Se desarrolló un par de aplicaciones demo con fines de pruebas, cabe destacar que dichas aplicaciones se realizaron empleando una tecnología que permite generar un mismo producto para las tres plataformas con mayor cuota de usuarios, iOS, Windows Phone y Android empleando la interfaz nativa de cada una de ellas, facilitando así la adopción. Se adjuntan algunas capturas de pantalla de dichos demos.

Es importante mencionar que para las pruebas que se realizaron con los padres de familia y el profesor se utilizó Android como plataforma standard para ambas aplicaciones demo, a fin de probar la API en un ambiente productivo y más cercano a la realidad.

Aplicación para tutores

La aplicación está diseñada específicamente para ser usada por tutores, por lo que permite ingresar uno o más *tokens* de estudiantes.

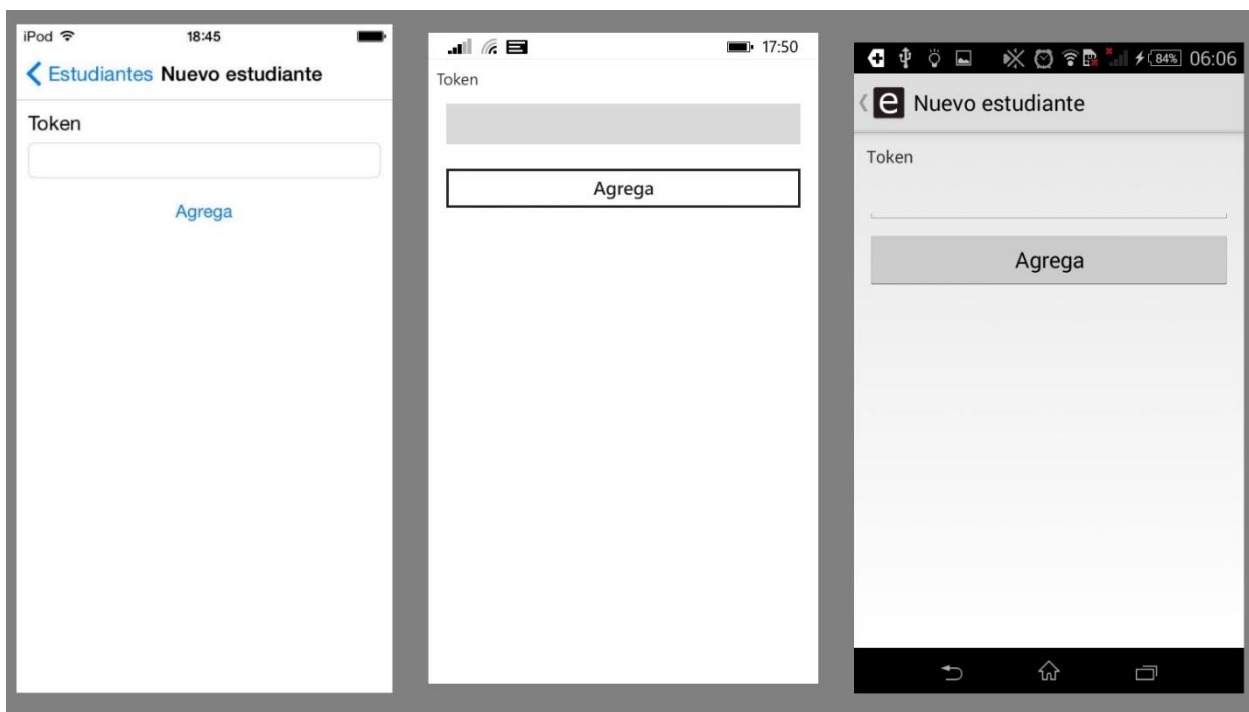


Ilustración 139 Pantalla de ingreso de token

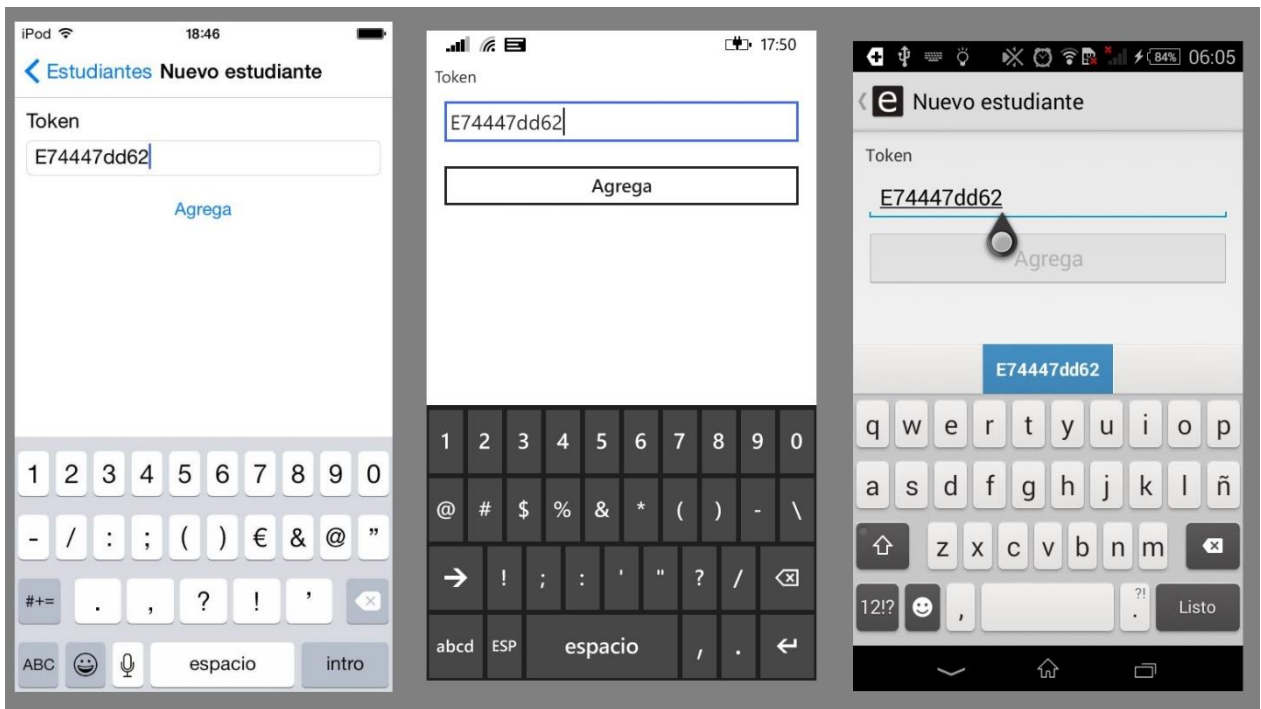


Ilustración 140 Pantalla de ingreso de token, con vista del teclado nativo

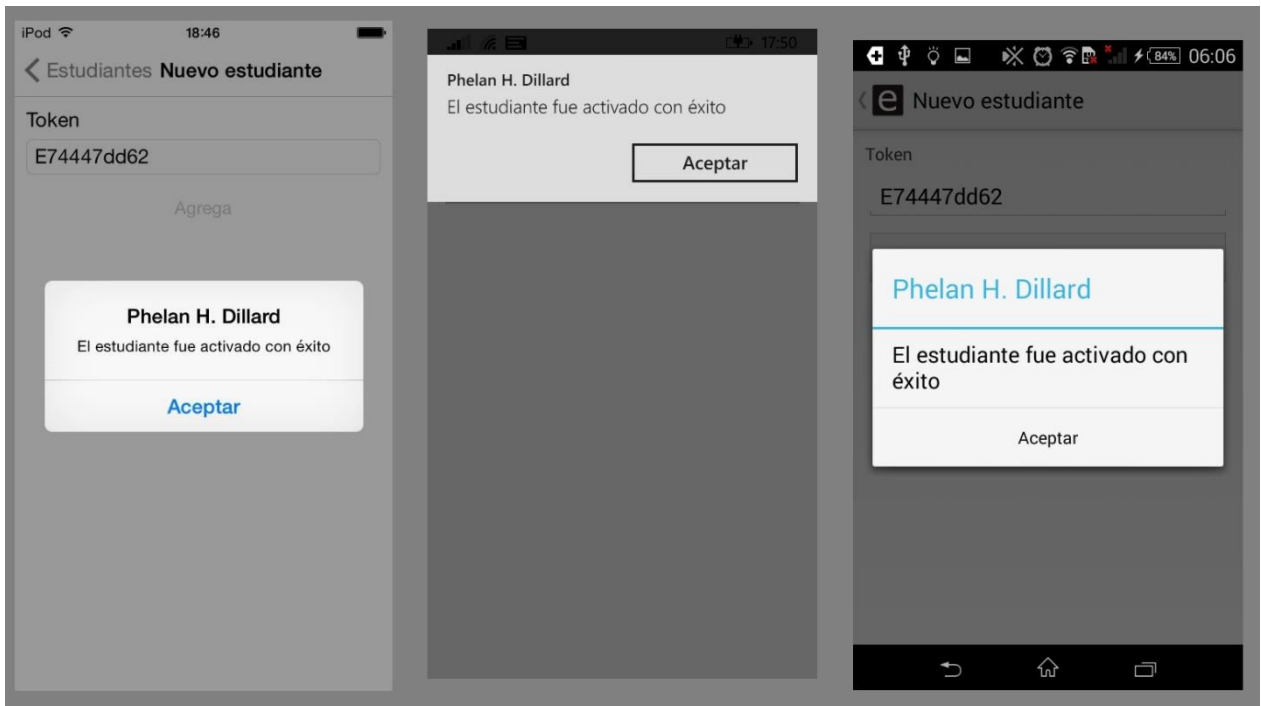


Ilustración 141 Confirmación de estudiante añadido

Una vez ingresado un *token* válido de estudiante, se envía un mensaje de confirmación informando que la acción se realizó con éxito.

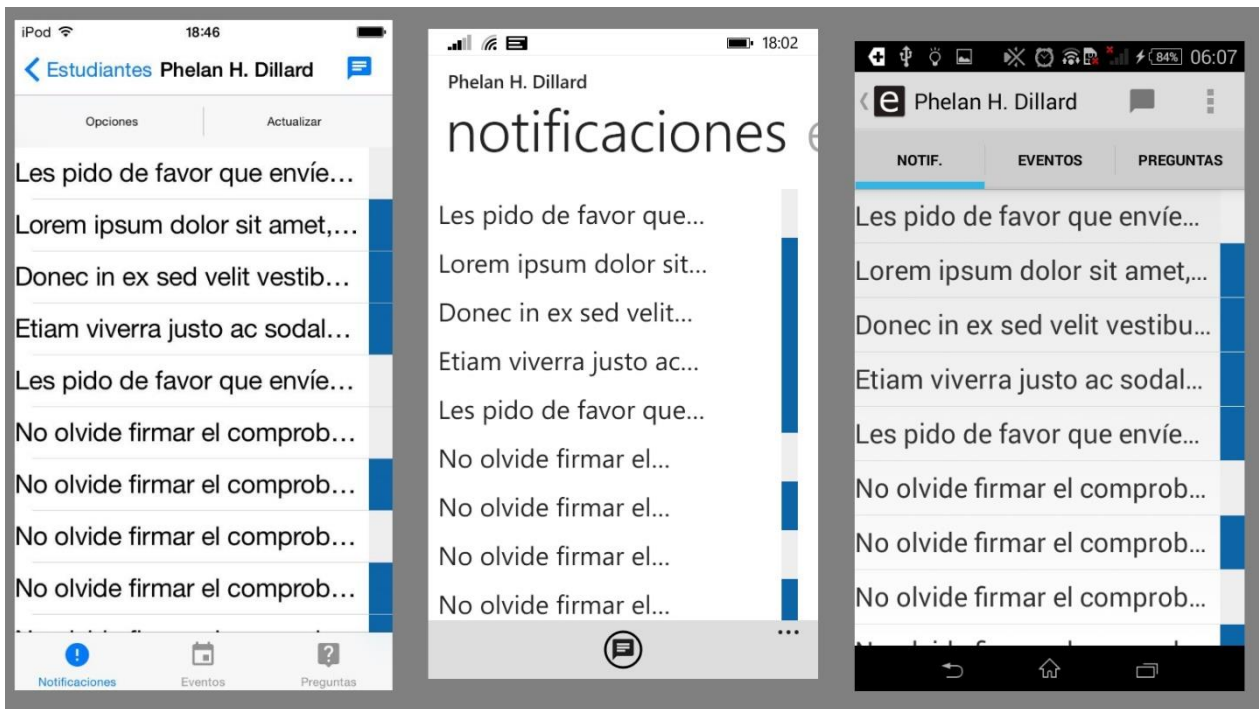


Ilustración 142 Lista de notificaciones

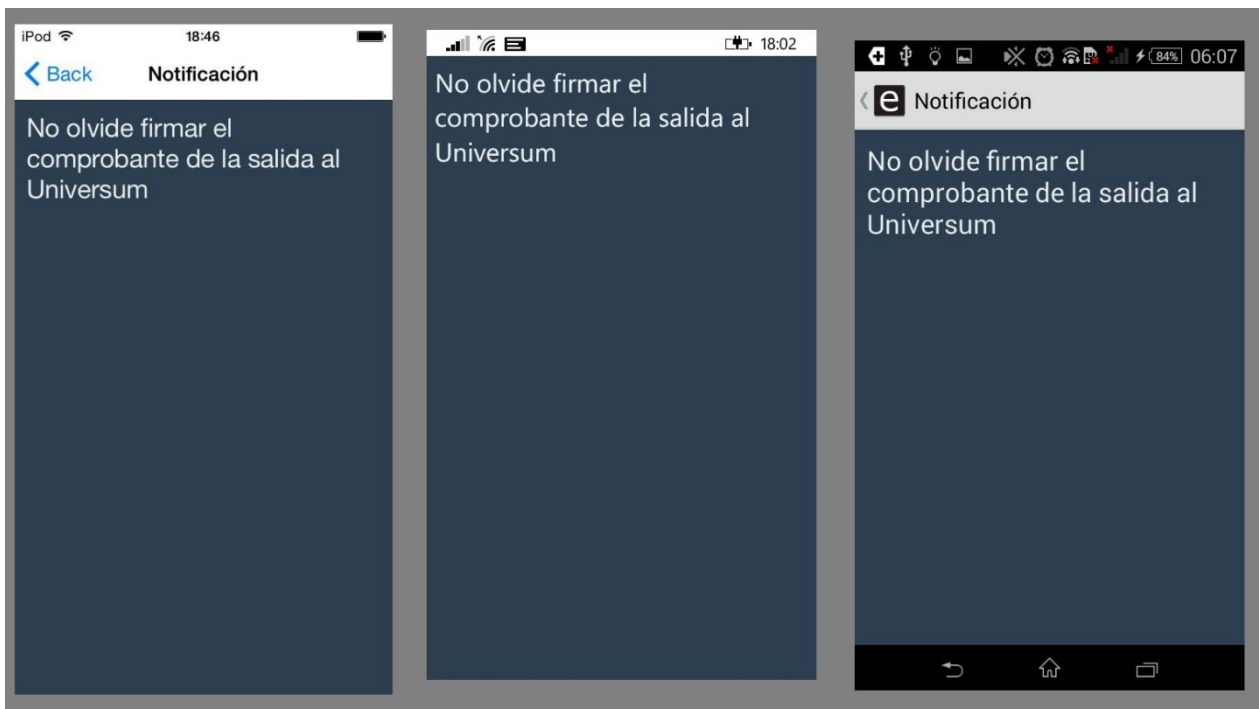


Ilustración 143 Pantalla de notificación

También cuenta con una pantalla de notificaciones, desde la que el tutor puede consultar las notificaciones más recientes, así como verlas por completo, acción que marca como vista la notificación.

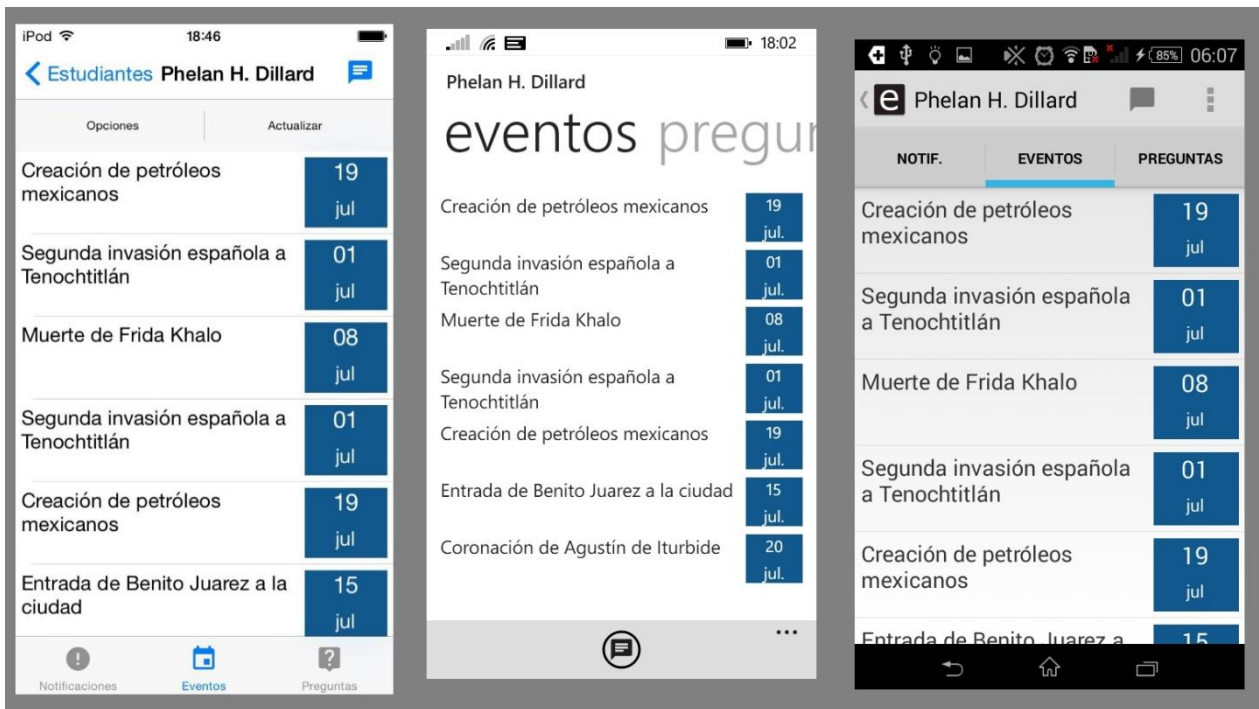


Ilustración 144 Lista de eventos

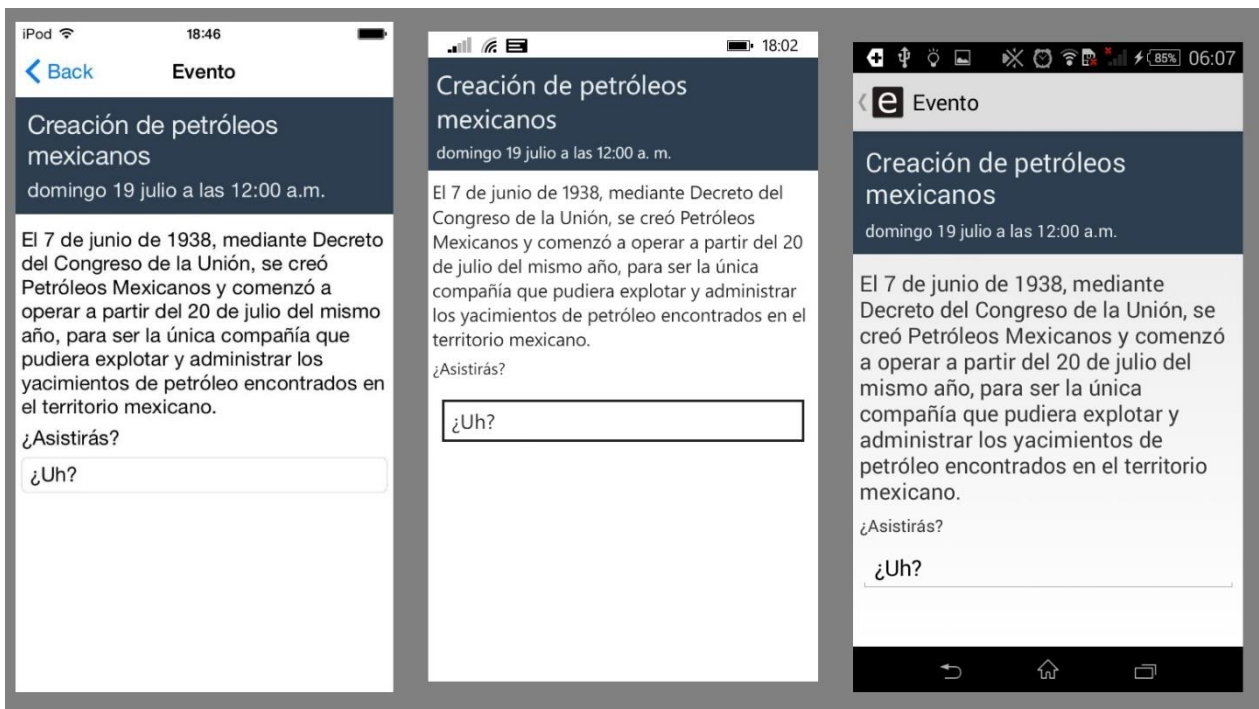


Ilustración 145 Pantalla de evento

A través de la *app* el tutor también tiene la posibilidad de ver los eventos a los que ha sido invitado, y una vez que seleccione uno de ellos, puede confirmar su asistencia dentro de la misma aplicación.

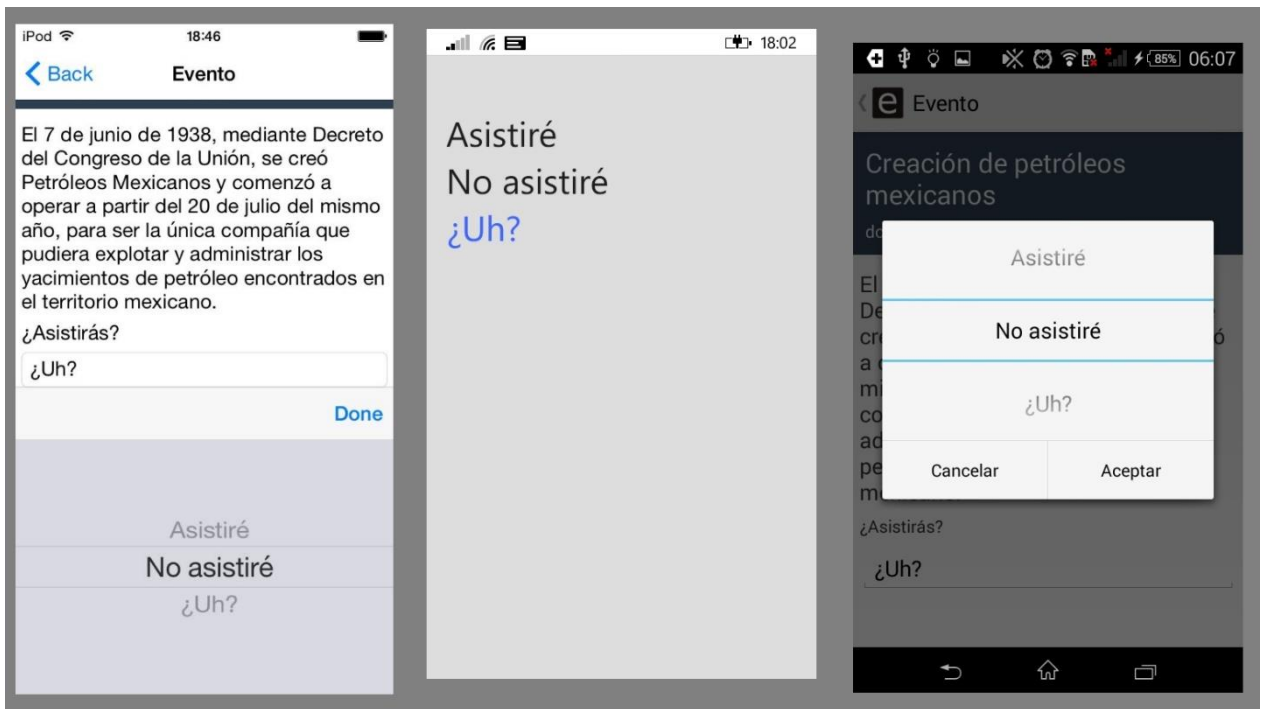


Ilustración 146 Control de responder a evento

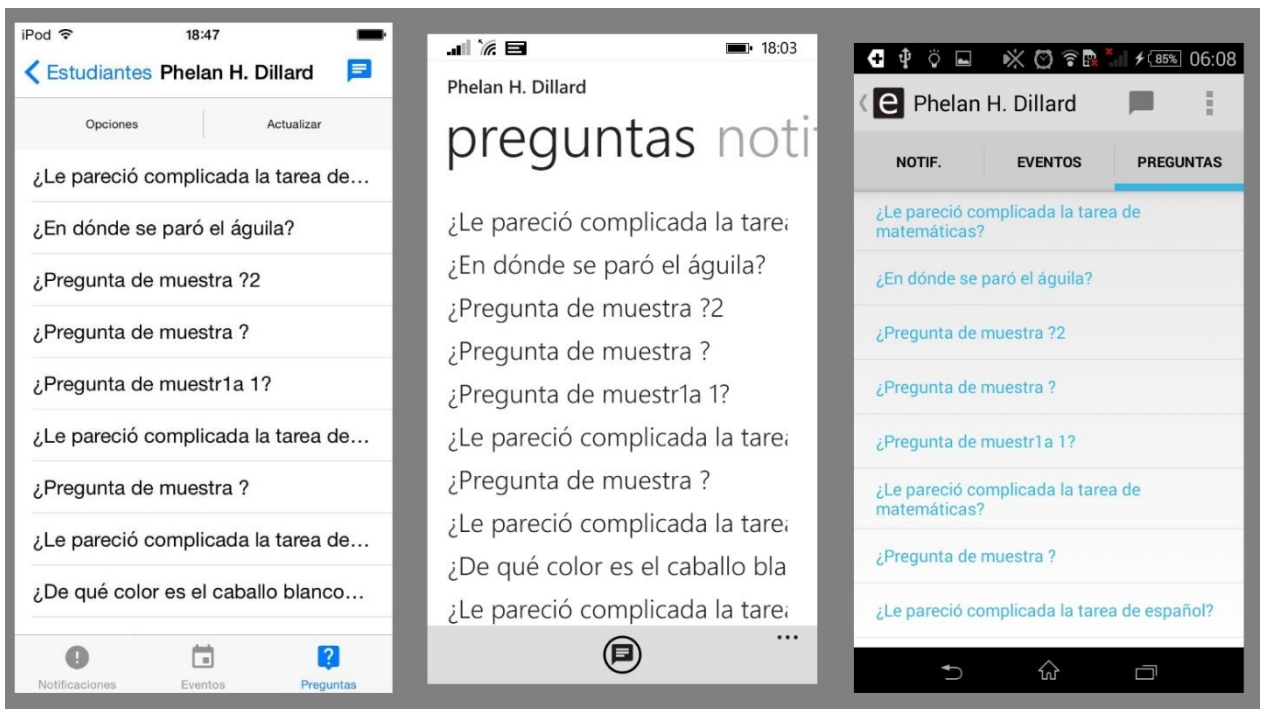


Ilustración 147 Lista de preguntas

La app también tiene una forma de mostrarle las preguntas que puede contestar el tutor, esto a través de una lista, una vez seleccionada una, puede responderlas usando la lista de respuestas predeterminadas que el profesor le asignó a dicha pregunta.



Ilustración 148 Pantalla de pregunta

Las conversaciones representan gran parte del intercambio de información que se busca que exista entre padres y tutores, y a través de esta aplicación es posible enviar y recibir mensajes.

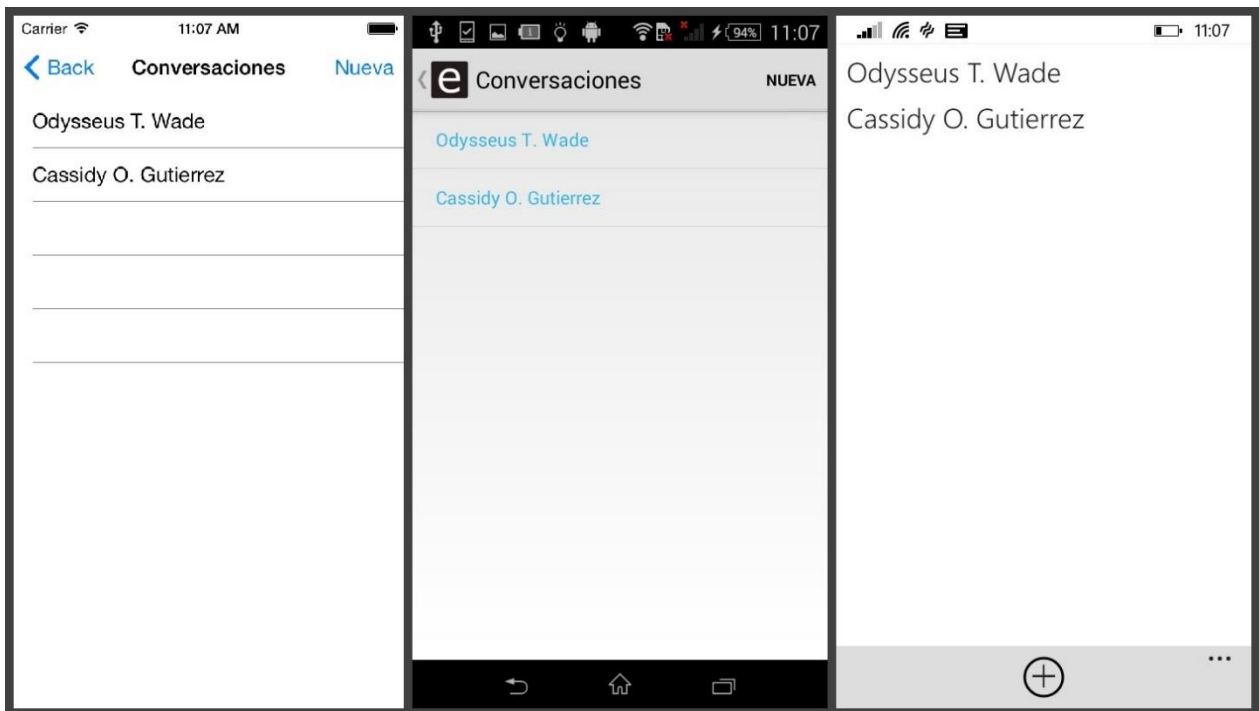


Ilustración 149 Conversaciones del tutor

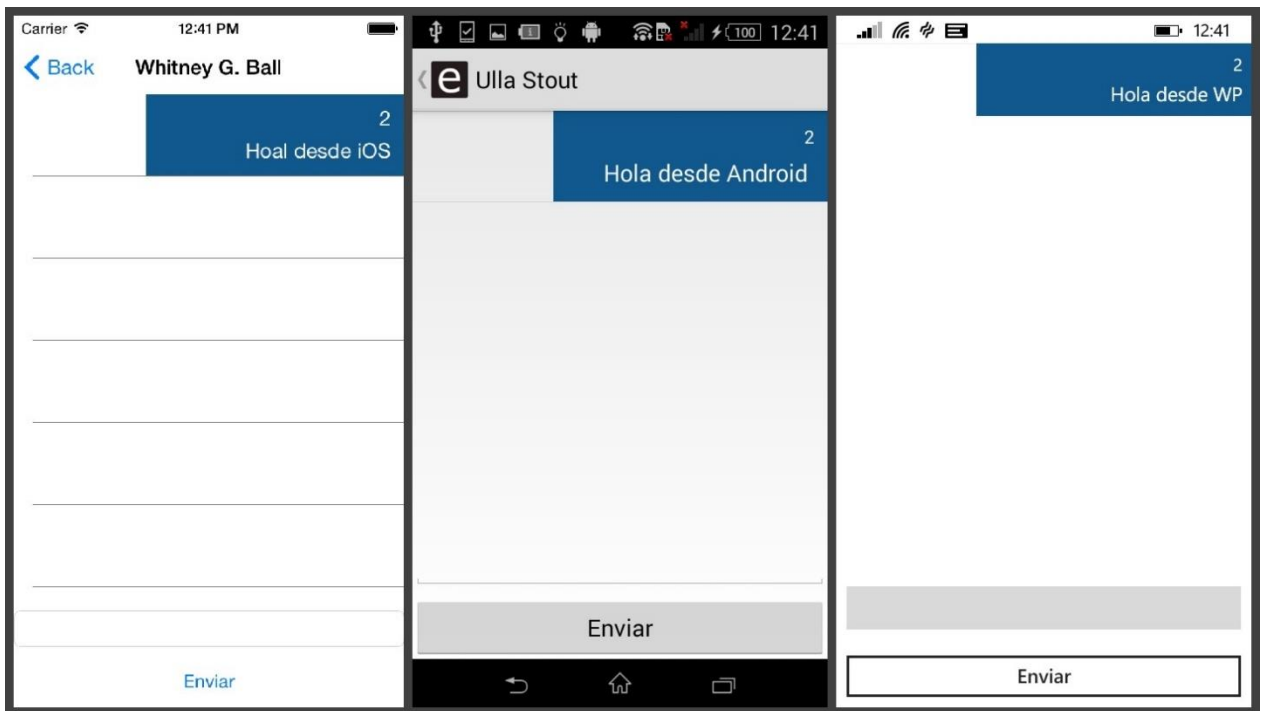


Ilustración 150 Mensajes del tutor

Aplicación para profesores

Esta aplicación está hecha para que los profesores interactúen con los tutores, por tanto, al contrario de la de tutores, esta solamente permite iniciar sesión con un solo número de usuario y contraseña, en esta las interacciones se rigen por los grupos. Al usuario se le presenta una lista de todos sus grupos una vez que ha iniciado sesión.

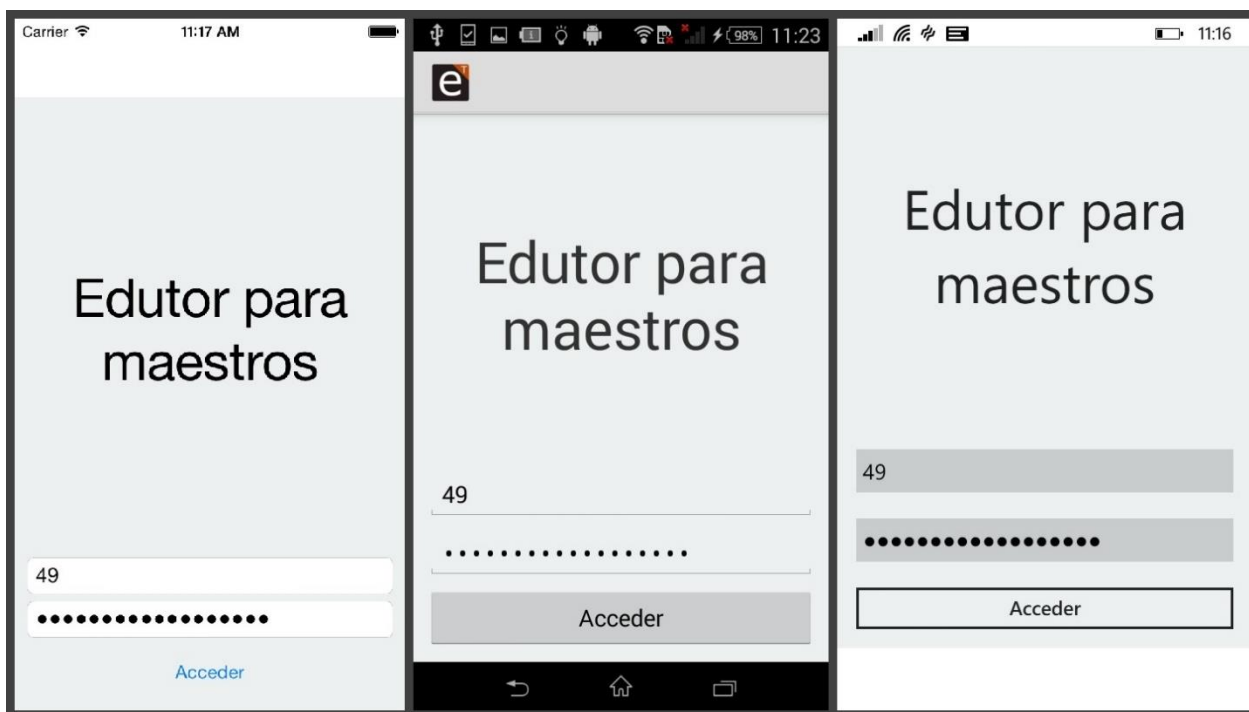


Ilustración 151 Pantalla de inicio de sesión para profesores

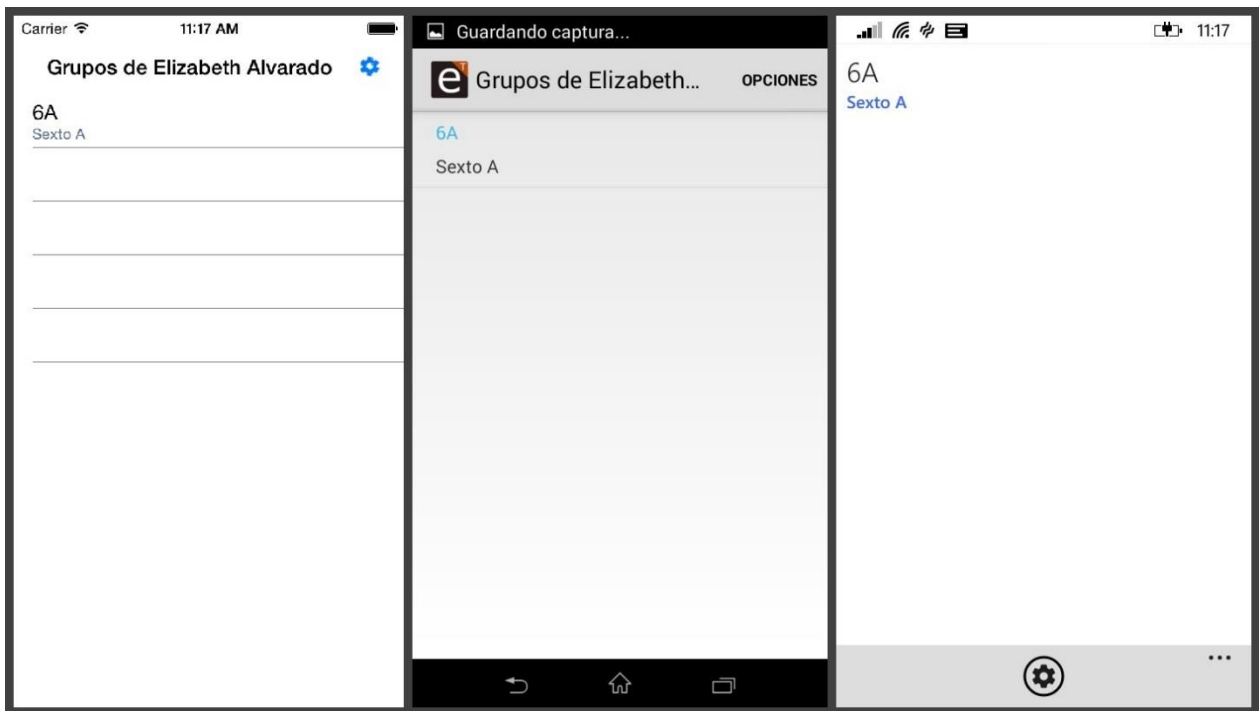


Ilustración 152 Pantalla de listado de grupos

Para cada grupo el usuario puede revisar el listado de interacciones que ha creado para él.

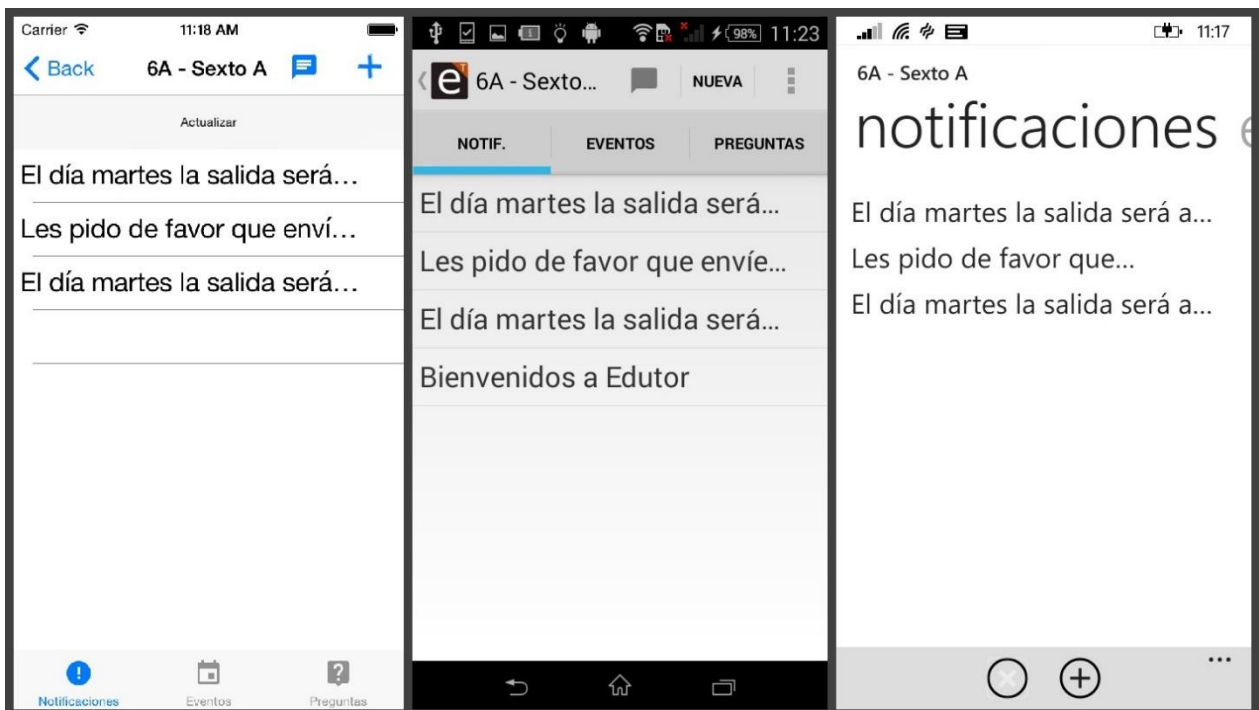


Ilustración 153 Listado de notificaciones

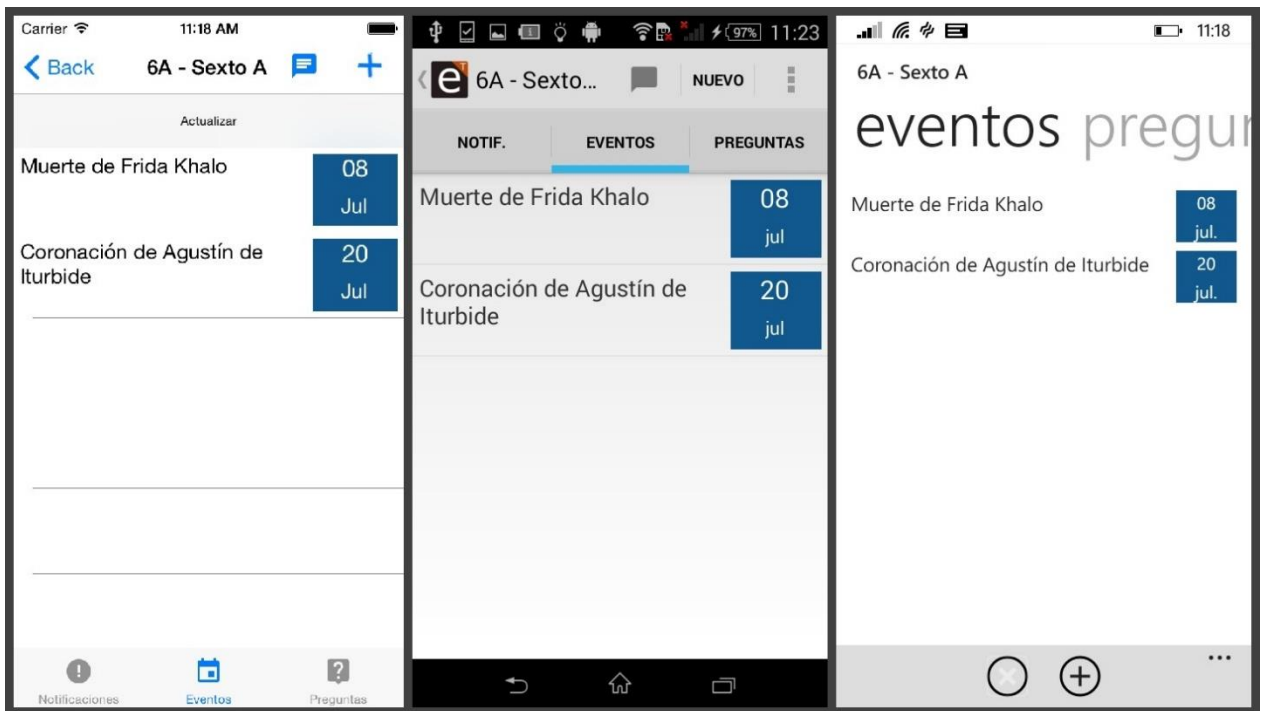


Ilustración 154 Listado de notificaciones

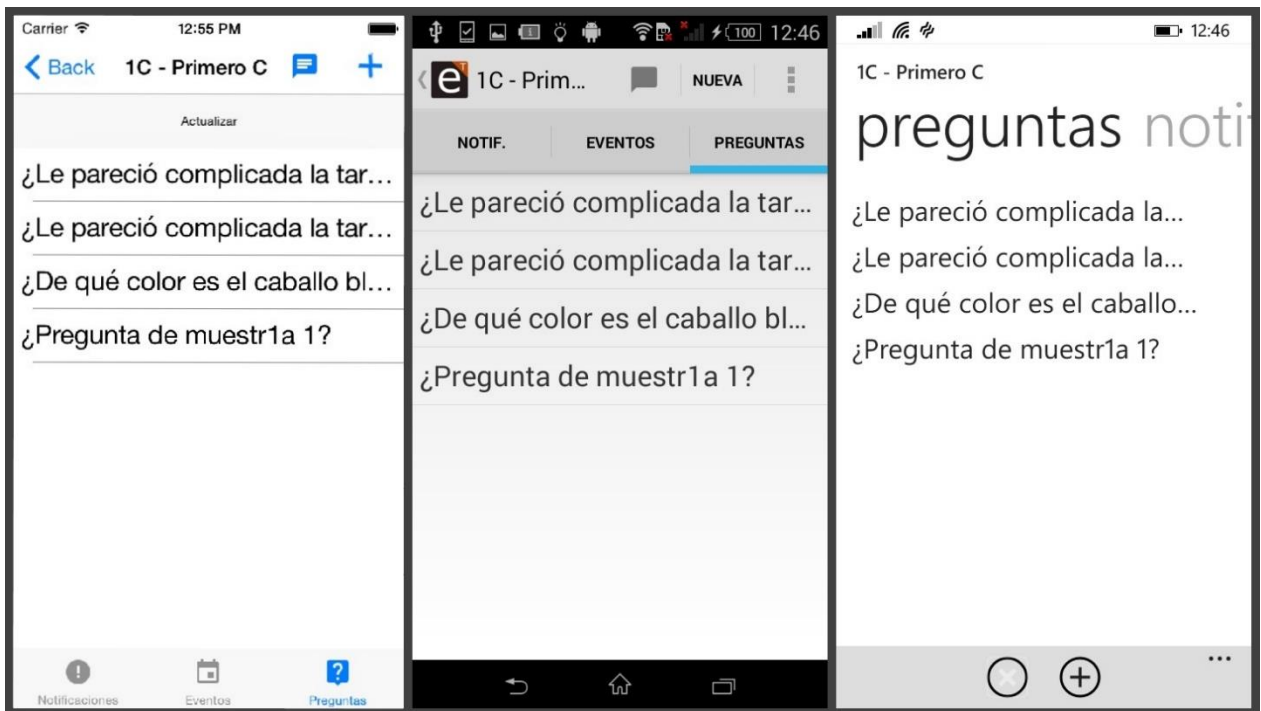


Ilustración 155 Listado de preguntas

Así mismo, desde esta aplicación se pueden crear nuevas interacciones de manera sencilla empleando los controles de cada plataforma:

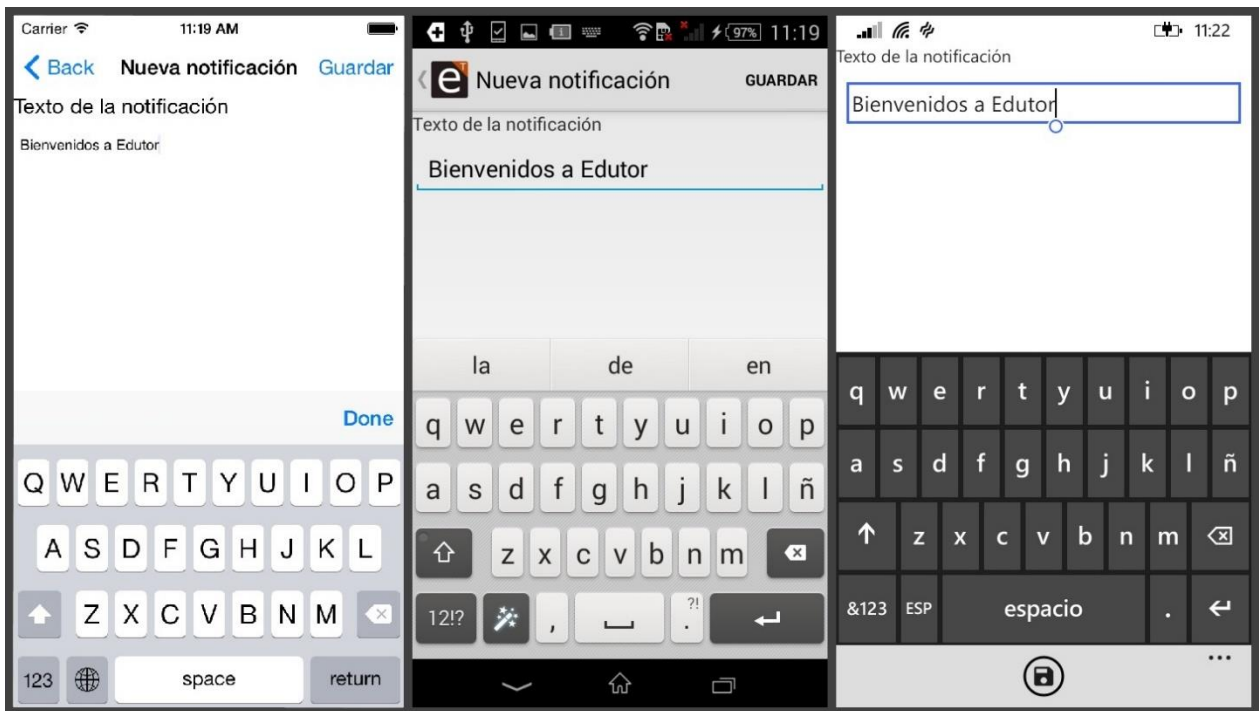


Ilustración 156 Creación de notificaciones

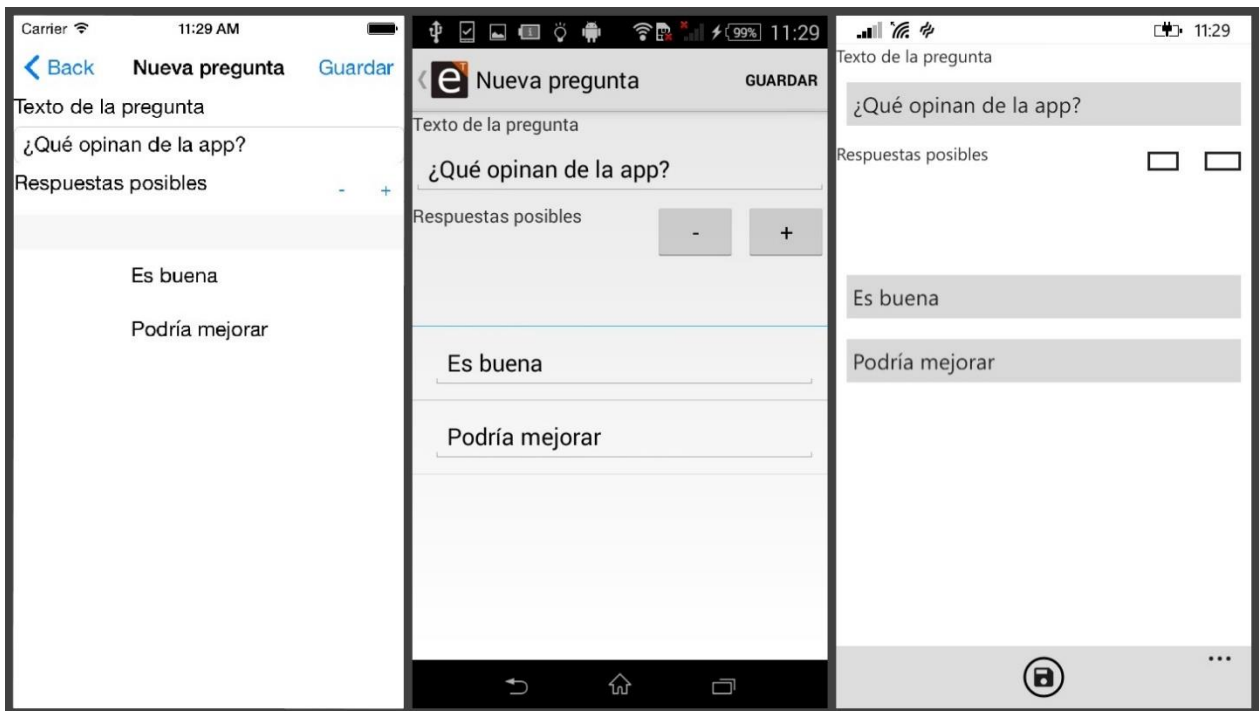


Ilustración 157 Creación de preguntas

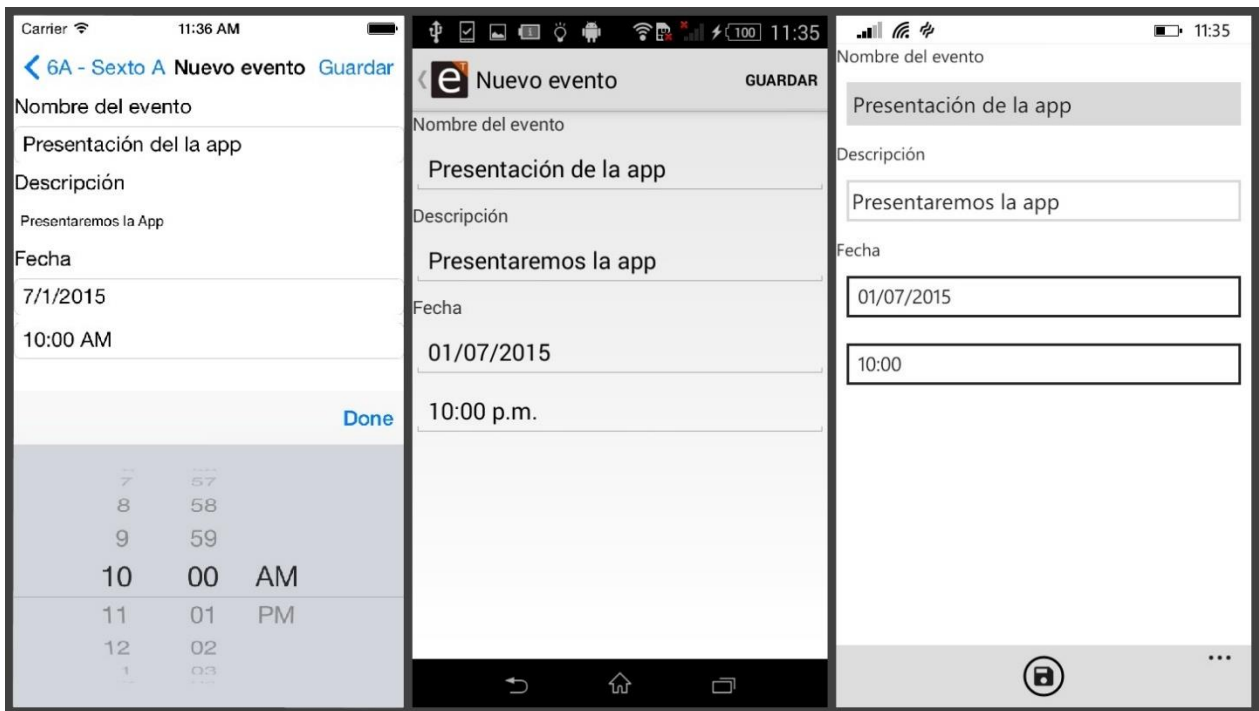


Ilustración 158 Creación de eventos

Es importante también recuperar las interacciones, por lo cual la aplicación muestra la cantidad de respuestas a preguntas y asistentes a los eventos que ha creado el profesor.



Ilustración 159 Visualización de preguntas



Ilustración 160 Visualización de eventos

Así mismo, a través de la aplicación se pueden acceder a las conversaciones que ha creado el profesor, o que un tutor creó para conversar con el profesor.

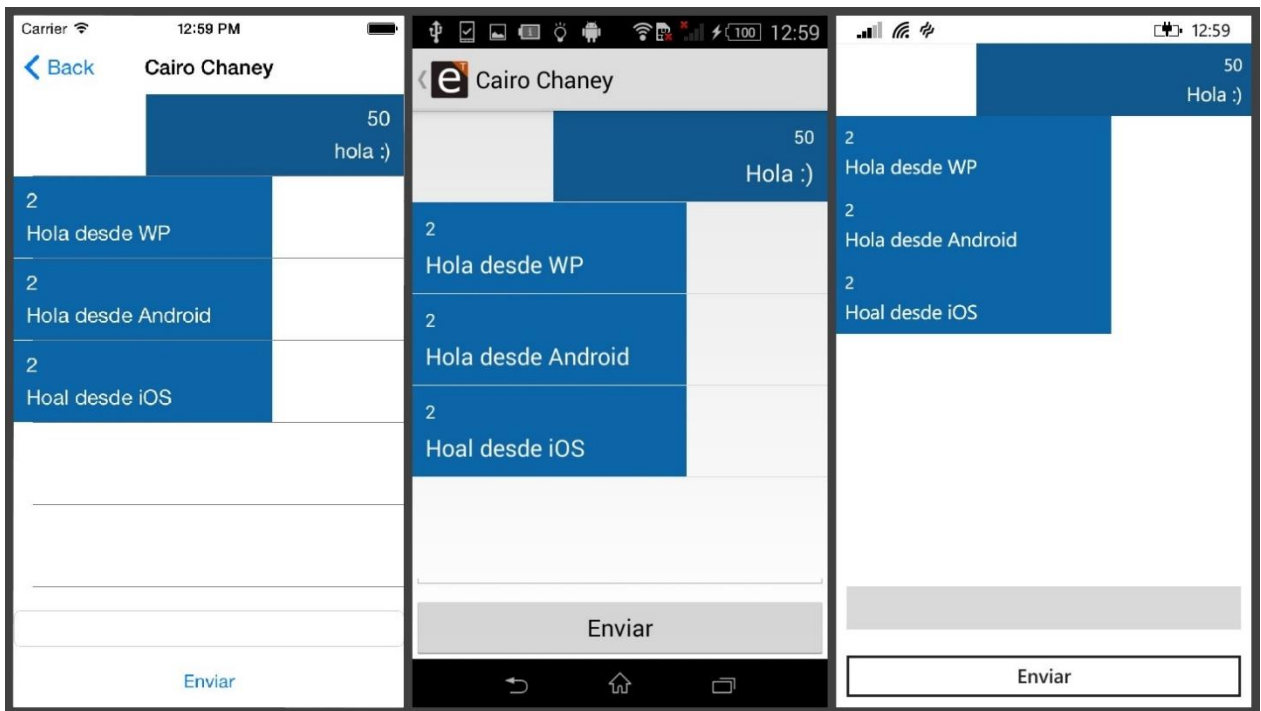


Ilustración 161 Página de conversaciones

Como se mencionó, la aplicación es solo un demo para fin de pruebas del equipo de desarrollo y para un grupo reducido de personas que accedieron a usarla en un entorno productivo para obtener retroalimentación y observar el comportamiento del servicio desarrollado en este trabajo terminal.

Conclusiones

Se puede concluir que la elección de las tecnologías de desarrollo fue acertada, ya que debido a los conocimientos previos sobre ellas, estas otorgaron la posibilidad de desarrollar la *API*. En particular se puede destacar el uso de *ASP.NET Web API* para elaborar el componente web del sistema, ya que la forma de definir las rutas, los tipos de petición y de respuesta se realizan de manera sencilla usando el mismo lenguaje de programación, es decir, sin incluir archivos de configuración extra.

Hubieron dificultades con el desarrollo de la conexión con la base de datos, dado que el equipo de desarrollo no había trabajado con *NHibernate* para definir el mapeo de objetos con el gestor de la base, sin embargo, dichas dificultades fueron superadas a tiempo con los conocimientos suficientes para continuar el desarrollo del sistema.

También se tuvieron dificultades para elegir la mejor forma de documentar el servicio web, de inicio se había planteado emplear *Swagger* para realizar dicha tarea, pero esta otorgaba más características que las requeridas para este trabajo terminal, es por eso que se decidió emplear el componente *Help Pages* del framework *ASP.NET*.

El presente trabajo fue consecuencia de una necesidad de la vida real, ya que los tutores por sus diferentes actividades diarias no prestan la atención necesaria a sus hijos lo cual tiene ciertas repercusiones como el rezago y abandono escolar de los alumnos.

Es por ello que se desarrolló la *API Edutor* la cual cuenta con los siguientes módulos:

1. Mensajería, el cual permite al profesor y tutor tener una comunicación directa.
2. Notificaciones, las cuales pueden realizarse de dos maneras, la primera general para toda la comunidad esto a través del administrador. La segunda por medio del profesor para el grupo de su elección.
3. Planeación de Eventos, el cual tiene la finalidad de agendar un evento e invitar a los tutores de toda la escuela o de un grupo en específico según sea el caso.
4. Encuestas, esta sección permite saber la opinión de los tutores respecto a diferentes temas que la escuela o el profesor deseen saber.

Así mismo se realizó el manual técnico de la API Edutor para que en trabajos futuros puedan hacer uso de dicha API y escalarla. Por último se realizó la aplicación cliente demo que consume los recursos de la API Edutor, esto con la finalidad de probar la correcta funcionabilidad y generar la evidencia de pruebas y resultados.

Con lo antes mencionado se cumplió con los objetivos planteados tanto generales como particulares y finalmente se concluye que el sistema resultante de este trabajo satisface las necesidades de los usuarios para quien fue planteada la solución.

Trabajo a futuro

Si bien el sistema desarrollado en durante el periodo correspondiente a trabajo terminal cubre las necesidades propuestas y los objetivos tanto generales como particulares, cabe destacar que es propenso a ser escalado, es decir, ser dotado de mayores funciones para robustecerlo añadiendo módulos que provean mayores y mejores formas de interacción entre tutores,-profesores y escuela. Razón por la cual se proponen las siguientes sugerencias de mejora:

1. Integrar la capacidad de que el servicio web entregue notificaciones push a los dispositivos registrados en él.
2. Permitir que las notificaciones vayan acompañadas de archivos, esto con la finalidad de que el profesor además de informar sobre tareas o trabajos dejados a sus alumnos, también pueda compartir algún documento donde se explique a mayor detalle el trabajo encargado.
3. Permitir que el alumno tenga más de un tutor, esto con la finalidad de otorgar mayor flexibilidad a los encargados de la supervisión del niño y asegurar que más personas se involucren en su desarrollo escolar.
4. Permitir que las encuestas consten de más de una pregunta, permitiendo así poder recabar más información de una manera más sencilla e intuitiva y con menos pasos al momento de crearlas.
5. Añadir un módulo de control de asistencia, para el caso de que exista una tutoría compartida poder detectar inasistencias del alumno y notificar a uno o varios de los tutores de las faltas que tiene el tutorado durante el ciclo escolar.
6. Añadir un mecanismo de notificaciones para detectar a los tutores que no hayan consultado sus eventos, preguntas o notificaciones nuevas durante algún periodo de tiempo y avisar al administrador o administradores de ese suceso.

Además de que al ser una API montada en un servicio web, es posible crear a su alrededor una variedad de aplicaciones cliente para crear un ecosistema más completo que permita su uso en diversos sitios. Por ello como trabajo a futuro también se proponen algunas aplicaciones cliente:

1. Sitio web, para que el administrador realice sus tareas de administración.
2. Aplicación para tablet, para que el profesor pueda generar la comunicación a través de lo ofrecido por Edutor.
3. Aplicación para smartphones, para que el tutor tenga siempre a la mano la información que genera el profesor.

4. Aplicación para smartwatch, al igual que la de smartphone esta servirá para ofrecer al tutor una mejor experiencia al usar la API.

Referencias

- [1] C. Alfonso, «El papel de la familia en la motivación escolar del alumnado,» de *La participación de los padres y madres en las escuelas*, Barcelona, Grao, 2010, pp. 55-57.
- [2] D. P. Caamaño, «Familia y escuela como contextos educativos,» de *Familia, Educacion e inmigración. Un programa de intervención pedagógica*, Santiago de Compostela, Uinversidad de Santiago de Compostela, 2008, pp. 93-146.
- [3] C. Alfonso, «Un espacio de comunicación y crecimiento múltiple: familias y centros educativos,» de *La participación de los padres y madres en la escuela*, Barcelona, Grao, 2010, pp. 18-20.
- [4] M. G. G. Alcaraz, «Biblioweb UNAM,» [En línea]. Available: http://biblioweb.tic.unam.mx/diccionario/htm/articulos/sec_9.htm. [Último acceso: 1 Septiembre 2014].
- [5] R. V. Neira, «El maestro como miembro del equipo educativo y su relación con la familia,» de *Educación infantil, padres y maestros*, Lulu.com, 2011, pp. 62-63.
- [6] J. Ramírez Cabañas y L. Gago Matías, *Guía práctica del profesor-tutor en educación primaria y secundaria*, España: Narcea Ediciones, 1995.
- [7] H. Duque Yepes, «Aproximaciones a un programa de escuela de padres,» de *La vida en familia: escuela de padres*, Editorial San Pablo, 1991, pp. 9-29.
- [8] Escolapp, «Escolapp,» Escolap, [En línea]. Available: <http://www.escolapp.es/>. [Último acceso: 14 septiembre 2014].
- [9] Clickart, «Clickedu,» Clickart, [En línea]. Available: <http://www.clickartedu.com/index-es.html>. [Último acceso: 14 septiembre 2014].
- [10] Apps4C, «KinderClose,» Apps4C, [En línea]. Available: <http://www.kinderclose.com/>. [Último acceso: 14 septiembre 2014].

- [11] Secretaría de Educación Pública, «Educación Preescolar,» Secretaría de Educación Pública, [En línea]. Available: http://www2.sepdf.gob.mx/que_hacemos/preescolar.jsp. [Último acceso: 12 septiembre 2014].
- [12] Secretaría de Educación Pública, «Educación Primaria,» Secretaría de Educación Pública, [En línea]. Available: http://www2.sepdf.gob.mx/que_hacemos/primaria.jsp. [Último acceso: 2014 septiembre 12].
- [13] Secretaría de Educación Pública, «Educación Secundaria,» Secretaría de Educación Pública, [En línea]. Available: http://www2.sepdf.gob.mx/que_hacemos/secundaria.jsp. [Último acceso: 2014 septiembre 12].
- [14] Microsoft Corporation, La Arquitectura Orientada a Servicios (SOA) de Microsoft, Diciembre 2006.
- [15] S. M. Solís, La Web Semántica, 2007.
- [16] D. K. Barry, «Web Services Explained,» de *Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager's Guide*, Newnes, 2012, pp. 22-33.
- [17] D. Orenstein, «Application Programming Interface,» *Computer World*, vol. 34, n° 2, pp. 66-67, 2000.
- [18] Varios, «Definition of Mobile App,» WhatIs.com, [En línea]. Available: <http://whatis.techtarget.com/definition/mobile-app>. [Último acceso: 17 septiembre 2014].
- [19] R. Englander, «The SOAP Message,» de *Java and SOAP*, O'Reilly Media, 2002, pp. 9-10.
- [20] C. A. M. Flores, «SOAP vs REST,» 8 Diciembre 2012. [En línea]. Available: <http://carlosmayta.blogspot.mx/>. [Último acceso: 7 Enero 2015].
- [21] Recess, [En línea]. Available: <http://www.recessframework.org/>. [Último acceso: 2014 octubre 13].
- [22] Play Framework, «Play Framework,» [En línea]. Available: <https://www.playframework.com/>. [Último acceso: 13 octubre 2014].

- [23] A. Håkansson y S. Robbins, «NANCY,» [En línea]. Available: <http://nancyfx.org/>. [Último acceso: 14 octubre 2014].
- [24] Facebook Inc., «Tornado Web Server,» [En línea]. Available: <http://www.tornadoweb.org>. [Último acceso: 14 october 2014].
- [25] Microsoft, «ASP.NET Web API,» [En línea]. Available: <http://www.asp.net/web-api>. [Último acceso: 2014 octubre 2014].
- [26] Oracle Corporation, «MySQL Community Server Edition,» Oracle Corporation, [En línea]. Available: <http://dev.mysql.com/downloads/mysql/>. [Último acceso: 20 noviembre 2014].
- [27] D. R. Márquez Ríos, G. Moreno Coatzozón y C. E. Rodríguez García, «Análisis comparativo entre SMBD,» [En línea]. Available: <http://es.slideshare.net/cinthiaerendida/analisis-comparativo-my-sql-vs-oracle-2716929>. [Último acceso: 20 noviembre 2014].
- [28] Microsoft Corporation, «Explorar SQL Server 2014,» [En línea]. Available: <http://www.microsoft.com/es-es/server-cloud/products/sql-server/>. [Último acceso: 21 Noviembre 2014].
- [29] Microsoft Corporation, «¿Qué es Azure?,» [En línea]. Available: <http://azure.microsoft.com/es-es/overview/what-is-azure/>. [Último acceso: 23 noviembre 2014].
- [30] Amazon Web Services, Inc., «Productos y servicios de Amazon Web Services,» [En línea]. Available: <http://aws.amazon.com/es/products/>. [Último acceso: 23 noviembre 2014].
- [31] Rackspace Inc., «About Rackspace,» [En línea]. Available: <http://www.rackspace.com/about>. [Último acceso: 24 noviembre 2014].
- [32] Quantitative Software Management. Inc., «QSM. The Intelligence behind Successful Software Projects,» [En línea]. Available: <http://www.qsm.com/resources/function-point-languages-table>. [Último acceso: Enero 2015].

[33] P. Galván, «Software Guru,» Diciembre 2014. [En línea]. Available: <http://sg.com.mx/revista/46/estudio-salarios-2014#.VM7IfO10zDc>. [Último acceso: 20145 Febrero 28].

Glosario

- **API:** (interfaz de programación de aplicaciones), es la interfaz a través de la cual un programa de aplicación accede a los recursos de un servicio u otro programa.
- **Edutor:** Nombre del sistema propuesto.
- **JSON:** acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.
- **Modelo-Vista-Controlador:** es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **REST:** (Transferencia de Estado Representacional) Arquitectura para el diseño de sistemas distribuidos en internet, centrado en el uso del protocolo HTTP.
- **Servicio Web:** Es cualquier componente de software que está disponible a través del protocolo HTTP, implementa un lenguaje estandarizado para intercambiar información.
- **URL:** localizador de recursos uniforme, es un identificador de recursos uniforme (URI) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que designa recursos en una red, como Internet.
- **XML:** lenguaje de marcas extensible, es un lenguaje de marcas desarrollado por el World Wide Web Consortium utilizado para almacenar datos en forma legible.