



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

**DISERTACIÓN DE GRADO PREVIA A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN**

TEMA:

**DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE
ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE
LA FUNDACIÓN MARIANA DE JESÚS**

AUTOR:

PABLO ESTEBAN GRANJA BORJA

QUITO, 2015

DEDICATORIA

Quiero dedicar esta disertación de grado principalmente a mi madre, ya que ella ha sido un ejemplo de vida para mí en todo sentido. Siempre me ha apoyado en todas las circunstancias de mi vida y me ha sabido guiar por el buen camino para llegar a ser una persona de bien. Gracias a su amor, motivación y perseverancia me ha enseñado que los objetivos que uno se proponga en la vida deben ser cumplidos con éxito.

También la dedico a mi hermano, quien ha sabido apoyarme siempre en la vida y sobretodo en estas épocas de estudios, dándome consejos y preocupándose por mi superación tanto personal como profesional.

También la dedico a mi padre, quien con sus consejos me ha motivado siempre a realizar mis estudios y ser un profesional de calidad. Gracias a su apoyo he podido seguir adelante en mi carrera y lograr culminarla.

También la dedico a mis abuelitos Rafael y Lucy, quienes siempre han estado pendientes de mi bienestar y mis logros académicos. Ellos me han enseñado que con los estudios, uno puede abrir todas las puertas de las oportunidades en la vida.

También la dedico a mis amigos más cercanos quienes han sabido acompañarme en esta etapa de vida y han sabido motivar la superación profesional entre todos nosotros.

AGRADECIMIENTOS

Primeramente, quiero agradecer a Dios por siempre cuidarme, bendecirme y guiarme por el camino del bien.

También quiero agradecer a los docentes que fueron parte de mis estudios en la Pontificia Universidad Católica del Ecuador, por haber transmitido todas sus enseñanzas que me van a ser muy útiles en mi vida profesional. En especial a mi tribunal de grado, a mi director Ing. Oswaldo Luna por dirigir mi disertación de grado y haberme ayudado en esta etapa final de mi carrera. A mis correctores Ing. Beatriz Campos e Ing. Andrés Jiménez, por ayudarme revisando mi disertación de grado y apoyándome para que el trabajo sea de calidad.

También quiero agradecer a la Fundación Mariana de Jesús, en especial a Gabriela Parra, coordinadora del Programa de Voluntariado, por haber colaborado con todo lo necesario para poder culminar esta disertación de grado.

Por último, agradecer a mi familia, quienes fueron la fuente principal de motivación, apoyo e inspiración para poder cumplir este sueño de ser Ingeniero de Sistemas.

TABLA DE CONTENIDOS

RESUMEN	1
ABSTRACT	2
INTRODUCCIÓN	3
CAPÍTULO 1.- FUNDAMENTO TEÓRICO.....	4
1.1. Antecedentes de la Fundación Mariana de Jesús	4
1.2. Software libre: características	6
1.3. Programación Orientada a Objetos: características.....	7
1.4. Modelo en Cascada: características.....	10
1.5. Arquitectura de Programación en Tres Capas: características.....	12
1.6. Herramientas y Tecnologías	13
1.6.1 JAVA	13
1.6.2. MySQL.....	15
CAPÍTULO 2.- ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA	17
2.1. Análisis de situación actual.....	17
2.2. Levantamiento de requerimientos del sistema	18
2.2.1. Funcionalidades del sistema.....	18
2.2.1.1. Autenticación de usuario.....	18
2.2.1.2. Administración de usuarios	18
2.2.1.3. Administración de proyectos.....	18
2.2.1.4. Administración de subproyectos	18
2.2.1.5. Administración de voluntarios	19
2.2.1.6. Administración de estudiantes.....	19
2.2.1.7. Administración de profesiones	19
2.2.1.8. Administración de universidades	19
2.2.1.9. Administración de facultades	19
2.2.1.10. Administración de registros de asistencia.....	19
2.2.1.11. Generación de reportes	19
2.2.2. Requerimientos funcionales	20
2.2.2.1. Diagrama general.....	20
2.2.2.2. Diagramas de casos de uso.....	21
2.2.2.3. Diagrama de clases a nivel conceptual	72

2.2.3.	Requerimientos no funcionales	72
2.2.3.1.	Restricciones de diseño e implementación	72
2.2.3.2.	Requerimientos especiales	73
2.2.4.	Otros requerimientos.....	78
2.2.4.1.	Interfaces gráficas.....	78
2.2.4.2.	Requisitos de Hardware y Software	88
CAPITULO 3.- DISEÑO DEL SISTEMA		89
3.1.	Diagramas de paquetes	89
3.2.	Diagrama de clases.....	90
3.2.1.	Clases de Interfaz Gráfica del Usuario	90
3.2.2.	Clases del Dominio del Problema	94
3.2.3.	Clases del Manejo de Datos	97
3.3.	Diagramas de secuencia.....	101
3.4.	Diagramas de actividades	162
3.5.	Modelo Entidad Relación	165
3.5.1.	Modelo conceptual.....	165
3.5.2.	Modelo físico.....	168
3.6.	Definición de estándares de diseño y desarrollo	169
CAPÍTULO 4.- PRUEBAS DEL SISTEMA.....		173
4.1.	Plan de pruebas de integración	173
4.2.	Pruebas de los casos de uso	177
CAPÍTULO 5.- CONCLUSIONES Y RECOMENDACIONES		189
5.1.	Conclusiones	189
5.2.	Recomendaciones.....	191
TRABAJOS CITADOS		193
BIBLIOGRAFÍA		195
ANEXO 1.....		196
GLOSARIO DE TÉRMINOS TÉCNICOS Y SIGLAS		196
ANEXO 2.....		199
CARTA DE AUSPICIO Y ACEPTACIÓN		199

ÍNDICE DE FIGURAS

Figura 1: Gráfico en representación del modelo en cascada.....	11
Figura 2.1: Diagrama de Casos de Uso General	20
Figura 2.2: Diagrama de Casos de Uso Autenticación de usuario	21
Figura 2.3: Diagrama a Detalle Ingresar al sistema.....	21
Figura 2.4: Diagrama de Casos de Uso Administración de usuarios	22
Figura 2.5: Diagrama a Detalle Crear Usuario.....	23
Figura 2.6: Diagrama a Detalle Modificar Usuario	24
Figura 2.7: Diagrama a Detalle Consulta general Usuario	25
Figura 2.8: Diagrama a Detalle Consulta por parámetros Usuario	26
Figura 2.9: Diagrama a Detalle Eliminar Usuario	26
Figura 2.10: Diagrama de Casos de Uso Administración de proyectos	28
Figura 2.11: Diagrama a Detalle Crear proyecto	28
Figura 2.12: Diagrama a Detalle Modificar proyecto.....	29
Figura 2.13: Diagrama a Detalle Consulta general proyecto	30
Figura 2.14: Diagrama a Detalle Eliminar proyecto.....	31
Figura 2.15: Diagrama de Casos de Uso Administración de subproyectos	32
Figura 2.16: Diagrama a Detalle Crear subproyecto	32
Figura 2.17: Diagrama a Detalle Modificar subproyecto	33
Figura 2.18: Diagrama a Detalle Consulta general subproyecto.....	34
Figura 2.19: Diagrama a Detalle Eliminar subproyecto	35
Figura 2.20: Diagrama de Casos de Uso Administración de voluntarios	36
Figura 2.21: Diagrama a Detalle Crear voluntario	37
Figura 2.22: Diagrama a Detalle Modificar voluntario	38
Figura 2.23: Diagrama a Detalle Consulta general voluntario.....	39
Figura 2.24: Diagrama a Detalle Consulta por parámetros voluntario	40
Figura 2.25: Diagrama a Detalle Eliminar voluntario	41
Figura 2.26: Diagrama de Casos de Uso Administración de estudiantes	42
Figura 2.27: Diagrama a Detalle Crear estudiante	42
Figura 2.28: Diagrama a Detalle Modificar estudiante.....	43
Figura 2.29: Diagrama a Detalle Consulta general estudiante	45
Figura 2.30: Diagrama a Detalle Consultar por parámetros estudiante	46
Figura 2.31: Diagrama a Detalle Eliminar estudiante	47

Figura 2.32: Diagrama de Casos de Uso Administración de profesiones	48
Figura 2.33: Diagrama a Detalle Crear profesión	48
Figura 2.34: Diagrama a Detalle Modificar profesión.....	49
Figura 2.35: Diagrama a Detalle Consultar general profesión.....	50
Figura 2.36: Diagrama a Detalle Eliminar profesión	51
Figura 2.37: Diagrama de Casos de Uso Administración de universidades	52
Figura 2.38: Diagrama a Detalle Crear universidad	53
Figura 2.39: Diagrama a Detalle Modificar universidad	54
Figura 2.40: Diagrama a Detalle Consulta general universidad.....	55
Figura 2.41: Diagrama a Detalle Eliminar universidad	56
Figura 2.42: Diagrama de Casos de Uso Administración de facultades.....	57
Figura 2.43: Diagrama a Detalle Crear facultad.....	57
Figura 2.44: Diagrama a Detalle Modificar facultad.....	58
Figura 2.45: Diagrama a Detalle Consulta general facultad	60
Figura 2.46: Diagrama a Detalle Eliminar facultad.....	61
Figura 2.47: Diagrama de Casos de Uso Administración de registros de asistencia	62
Figura 2.48: Diagrama a Detalle Crear registro de asistencia.....	62
Figura 2.49: Diagrama a Detalle Modificar registro de asistencia.....	64
Figura 2.50: Diagrama a Detalle Consulta general registro de asistencia	65
Figura 2.51: Diagrama a Detalle Consulta por parámetros registro de asistencia	66
Figura 2.52: Diagrama a Detalle Eliminar registro de asistencia.....	67
Figura 2.53: Diagrama de Casos de Uso Generación de reportes	68
Figura 2.54: Diagrama a Detalle Reporte de participantes por fechas	69
Figura 2.55: Diagrama a Detalle Reporte de horas de asistencia.....	70
Figura 2.56: Diagrama de Clases a Nivel Conceptual.....	72
Figura 2.57: Cuadro de Requerimientos Especiales.....	73
Figura 2.58: Modelo de Interfaz Gráfica de Login.....	78
Figura 2.59: Modelo de Interfaz Gráfica de Pantalla Principal.....	78
Figura 2.60: Modelo de Interfaz Gráfica de Pantalla de Usuarios	79
Figura 2.61: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Usuarios	79
Figura 2.62: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Usuarios.....	79
Figura 2.63: Modelo de Interfaz Gráfica de Pantalla de Proyectos/Subproyectos.....	80

Figura 2.64: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Proyectos/Subproyectos	80
Figura 2.65: Modelo de Interfaz Gráfica de Pantalla de Voluntarios.....	81
Figura 2.66: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Voluntarios	81
Figura 2.67: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Voluntarios	82
Figura 2.68: Modelo de Interfaz Gráfica de Pantalla de Estudiantes	82
Figura 2.69: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Estudiantes.....	83
Figura 2.70: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Estudiantes.....	83
Figura 2.71: Modelo de Interfaz Gráfica de Pantalla de Profesiones	84
Figura 2.72: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Profesiones	84
Figura 2.73: Modelo de Interfaz Gráfica de Pantalla de Universidades/Facultades	85
Figura 2.74: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Universidades/Facultades.....	85
Figura 2.75: Modelo de Interfaz Gráfica de Pantalla de Registros de asistencia.....	86
Figura 2.76: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Registros de asistencia.....	86
Figura 2.77: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Registros de asistencia	87
Figura 2.78: Modelo de Interfaz Gráfica de Pantalla de Reporte de participantes por fechas	87
Figura 2.79: Modelo de Interfaz Gráfica de Pantalla de Reporte de horas de asistencia	88
Figura 3.1: Diagrama de paquetes GUI	89
Figura 3.2: Diagrama de paquetes DP	89
Figura 3.3: Diagrama de paquetes MD.....	90
Figura 3.4: Diagrama de paquetes Errores/Imágenes/Funciones	90
Figura 3.5: Diagrama de Clases de Interfaz Gráfica	93
Figura 3.6: Diagrama de Clases del Dominio del Problema.....	94
Figura 3.6: Diagrama de Clases del Dominio del Problema.....	96
Figura 3.7: Diagrama de Clases del Manejo de Datos.....	97
Figura 3.7: Diagrama de Clases del Manejo de Datos.....	98

Figura 3.7: Diagrama de Clases del Manejo de Datos.....	99
Figura 3.7: Diagrama de Clases del Manejo de Datos.....	100
Figura 3.8: Diagrama de Secuencia Ingresar al sistema	101
Figura 3.9: Diagrama de Secuencia Crear Usuario	102
Figura 3.10: Diagrama de Secuencia Modificar Usuario.....	103
Figura 3.11: Diagrama de Secuencia Consulta General Usuario	104
Figura 3.12: Diagrama de Secuencia Consulta por parámetros Usuario	104
Figura 3.13: Diagrama de Secuencia Eliminar Usuario	105
Figura 3.14: Diagrama de Secuencia Crear Proyecto.....	106
Figura 3.15: Diagrama de Secuencia Modificar Proyecto	107
Figura 3.16: Diagrama de Secuencia Consulta general Proyecto	108
Figura 3.17: Diagrama de Secuencia Eliminar Proyecto	108
Figura 3.18: Diagrama de Secuencia Crear subproyecto.....	109
Figura 3.19: Diagrama de Secuencia Modificar subproyecto.....	110
Figura 3.20: Diagrama de Secuencia Consulta general subproyecto	111
Figura 3.21: Diagrama de Secuencia Eliminar subproyecto.....	112
Figura 3.22: Diagrama de Secuencia Crear voluntario.....	114
Figura 3.23: Diagrama de Secuencia Modificar voluntario	116
Figura 3.24: Diagrama de Secuencia Consulta general voluntario	118
Figura 3.25: Diagrama de Secuencia Consulta por parámetros voluntario	120
Figura 3.26: Diagrama de Secuencia Eliminar voluntario.....	123
Figura 3.27: Diagrama de Secuencia Crear estudiante.....	126
Figura 3.28: Diagrama de Secuencia Modificar estudiante	129
Figura 3.29: Diagrama de Secuencia Consulta general estudiante.....	131
Figura 3.30: Diagrama de Secuencia Consulta por parámetros estudiante.....	133
Figura 3.31: Diagrama de Secuencia Eliminar estudiante	136
Figura 3.32: Diagrama de Secuencia Crear profesión	137
Figura 3.33: Diagrama de Secuencia Modificar profesión	138
Figura 3.34: Diagrama de Secuencia Consulta general profesión.....	139
Figura 3.35: Diagrama de Secuencia Eliminar profesión	140
Figura 3.36: Diagrama de Secuencia Crear universidad.....	141
Figura 3.37: Diagrama de Secuencia Modificar universidad.....	142
Figura 3.38: Diagrama de Secuencia Consulta general universidad	143

Figura 3.39: Diagrama de Secuencia Eliminar universidad.....	144
Figura 3.40: Diagrama de Secuencia Crear facultad	145
Figura 3.41: Diagrama de Secuencia Modificar facultad	146
Figura 3.42: Diagrama de Secuencia Consulta general facultad.....	147
Figura 3.43: Diagrama de Secuencia Eliminar facultad	148
Figura 3.44: Diagrama de Secuencia Crear registro de asistencia	150
Figura 3.45: Diagrama de Secuencia Modificar registro de asistencia	152
Figura 3.46: Diagrama de Secuencia Consulta general registro de asistencia.....	153
Figura 3.47: Diagrama de Secuencia Consulta por parámetros registro de asistencia	155
Figura 3.48: Diagrama de Secuencia Eliminar registro de asistencia	157
Figura 3.49: Diagrama de Secuencia Reporte de participantes por fecha	159
Figura 3.50: Diagrama de Secuencia Reporte de horas de asistencia	161
Figura 3.51: Diagrama de Actividades Ingreso de Voluntario	162
Figura 3.52: Diagrama de Actividades Ingreso de Estudiante	163
Figura 3.53: Diagrama de Actividades Evaluación de las horas de asistencia	164
Figura 3.54: Diagrama de Modelo Conceptual.....	165
Figura 3.54: Diagrama de Modelo Conceptual.....	167
Figura 3.55: Diagrama de Modelo Físico	168
Figura 4.1: Diagrama de Integración por dependencias	174
Figura 4.1: Diagrama de Integración por dependencias	175
Figura 4.1: Diagrama de Integración por dependencias	176

RESUMEN

Esta disertación de grado muestra el desarrollo de un sistema de administración para el Programa de Voluntariado de la Fundación Mariana de Jesús.

Para el desarrollo del sistema se utilizó el enfoque metodológico del Modelo en Cascada, en donde se sigue un orden riguroso de las etapas en el proceso del desarrollo del software, y ninguna etapa puede comenzar sin antes terminar la anterior. De esta forma, los requerimientos que necesite la empresa, se los analizará minuciosamente, para así definirlos antes de diseñar el sistema y proceder con la codificación.

Para la codificación del sistema se utilizó el paradigma de Programación Orientada a Objetos, para realizar un sistema en donde los objetos tengan interacción y se relacionen más con el mundo real, y de esta forma poder hacerlo más visual para el usuario final.

También se utilizó la Arquitectura de Programación en Tres Capas, ya que de esta forma se pudo distribuir el desarrollo del sistema en niveles, y con esto permitir un mantenimiento más sencillo, además de volverlo flexible a cambios o nuevas funcionalidades que el usuario pueda necesitar.

Se utilizaron herramientas y tecnologías de software libre como el lenguaje de programación Java, el IDE de desarrollo Netbeans, y como base de datos MySQL.

Este trabajo está dividido en cinco capítulos, en los cuales el primero nos indica los antecedentes de la Fundación Mariana de Jesús y el fundamento teórico de la parte del desarrollo del sistema. En los tres siguientes capítulos, se desarrolla el sistema de acuerdo a las fases del ciclo de vida del Modelo en Cascada. Y el último capítulo está destinado para las conclusiones y recomendaciones que se obtuvieron al finalizar el desarrollo del sistema.

ABSTRACT

This degree dissertation shows the development of the system of Administration for the Volunteer Program of the Fundacion Mariana de Jesus.

For the development of the system was used the methodological approach called Waterfall Model, where a strict order of steps is followed in the process of software development, and any phase of development can begin without first completing the above one.

Thus, the requirements needed by the company, will be analyze carefully and be able to define them before designing the system and proceed with the coding.

For coding the system, it was used the Paradigm OOP, where objects have more interaction with the real world, and make it more visual for the end user.

Also, the Multi-tier Architecture of Programming was used, because with this architecture, it was possible to distribute the development of the system into tiers, and thereby allow easier maintenance, and make it flexible for changes or new features that the user may need.

Tools and open source technologies such as the Java programming language, the Netbeans IDE development, and the MySQL database were used.

This degree dissertation is divided into five chapters, in which the first chapter tells us the history of the Fundacion Mariana de Jesus and also the theoretical foundation of the development of the system. The next three chapters, shows the development of the system according to the phases of the life cycle of the Waterfall Model. And the last chapter is intended for the conclusions and recommendations that were obtained to finalize the development of the system.

INTRODUCCIÓN

La Fundación Mariana de Jesús posee su Programa de Voluntariado, donde maneja diferentes proyectos de ayuda social, en los cuales participan voluntarios y estudiantes de universidades. Cada proyecto comprende diferentes actividades, las cuales son realizadas por voluntarios que desean participar, o por estudiantes de universidades que eligen estos proyectos para ayudar a la comunidad y a la vez cumplir con horas de acción social que son requisitos en ciertas universidades.

El proceso de control que la fundación ha realizado de cada proyecto con los voluntarios y estudiantes que participan en ellos, hasta el momento ha sido administrado de forma manual. Esto ha generado una desorganización en el manejo de la información de cada voluntario o de cada estudiante, produciendo confusión a la hora de realizar consultas o reportes de las actividades que se realizan en cada proyecto.

Por lo cual, esta disertación pretende crear un sistema de administración que automatice todos los procesos que se han llevado de forma manual en la Fundación Mariana de Jesús, y así agilizar el manejo de la información de cada voluntario o de cada estudiante, y conocer en qué proyectos están participando y cuáles son las actividades que se están realizando.

Este proyecto agilizará las consultas del estado de cada uno de los voluntarios o estudiantes que estén realizando sus proyectos de acción social en la fundación, y permitirá llevar un control más ordenado de las actividades realizadas en cada uno de los proyectos de la fundación.

CAPÍTULO 1.- FUNDAMENTO TEÓRICO

1.1. Antecedentes de la Fundación Mariana de Jesús

La Fundación Mariana de Jesús es una institución de derecho privado sin fines de lucro, con finalidad social o pública. Constituida legalmente el 5 de diciembre de 1939 y aprobada mediante Acuerdo Ministerial N° 1734 del 20 de junio de 1950 por el Ministerio de Previsión Social y Trabajo, luego modificado mediante Acuerdo Ministerial N° 1009 del 23 de julio de 1998 por el Ministerio de Bienestar Social, que se halla vigente a la fecha. **(1)**

Su objetivo general es mejorar la calidad de vida de la población de menores recursos económicos facilitando su acceso a servicios que contribuyan a dignificar a la persona, la familia y la comunidad.

Las directrices para la gestión institucional social que manejan son las siguientes:

- Identificar oportuna e integralmente a los grupos sociales que se hallan en situación de exclusión. **(1)**
- Promover la participación en el diseño y gestión de los programas sociales.
- Establecer alianzas con diversos grupos para constituir redes locales de solidaridad y acción social. **(1)**
- Ejecutar estrategias de inclusión social conjuntamente con redes, servicios públicos e iniciativas privadas. **(1)**
- Integrar formación y cultivo de valores en capacitación, organización y autogestión con grupos y comunidades de intervención. **(1)**
- Generar incidencia política pública local en los diversos niveles del gobierno con los que se coordine durante la ejecución de los programas. **(1)**
- Generar alianzas estratégicas con redes jesuitas tanto nacionales como internacionales. **(1)**
- Sistematizar los aprendizajes generados en cada programa social y así poder mejorar los modelos de gestión. **(1)**

La Fundación Mariana de Jesús como parte de su gestión institucional social promueve Programas de Desarrollo Humano y Social, los cuales inician de la persona y son destinados para impulsar su integración familiar, comunitaria y social utilizando, como medios: la participación, la organización y el trabajo conjunto.

Cada Programa está enfocado en un horizonte común: “Queremos un Ecuador con sujetos activos en el desarrollo integral del país que incidan en la transformación de la sociedad desde los valores del Evangelio, sujetos compasivos y críticos, capaces de asumir su dignidad de seres humanos y de responder creativamente a los

vertiginosos cambios del mundo”. Plan Apostólico de la Fundación Mariana de Jesús 2013-2017. (2)

Actualmente, estos son los Programas que se están desarrollando:

1. INCLUSIÓN SOCIAL DE ADULTOS MAYORES (2)

Programas para incentivar una calidad de vida digna para los adultos mayores, realizando actividades participativas con el apoyo de los colaboradores del Programa de Voluntariado, invitando a las familias y comunidades con el fin de promover la integración social.

2. EDUCACIÓN DE NIÑOS Y JOVENES (2)

Programas para dotar de recursos necesarios a los niños y jóvenes de los sectores rurales, apoyando los esfuerzos de sus familias y comunidades para que permanezcan dentro del sistema educativo y terminen la educación básica o media.

También aportando un ambiente de experiencias positivas de servicio al prójimo con el Programa de Voluntariado de los colaboradores.

3. VIVIENDA SOCIAL DIGNA (2)

La prioridad de la Fundación Mariana de Jesús es atender en todos los territorios donde realizan sus programas y con más empeño a las personas que carecen de vivienda o no pueden acceder a una.

4. ATENCIÓN PRIMARIA DE SALUD (2)

La Fundación Mariana de Jesús impulsará la creación de unidades de salud primaria en sectores que han sido excluidos por diferentes razones como distancia o pobreza.

5. ASESORÍA Y ASISTENCIA TÉCNICA (2)

Este servicio es dirigido a gobiernos autónomos descentralizados tanto parroquiales como municipales y a organizaciones comunitarias tanto urbanas como rurales, que requieren asesoría y/o acompañamiento en procesos de negociación técnica, política o jurídica con instituciones públicas y privadas.

Toda esta labor no se la podría realizar si no fuera por la ayuda de los colaboradores de la Fundación Mariana de Jesús que pertenecen al Programa de Voluntariado.

Su objetivo es promover la participación de la sociedad civil, nacional o extranjera en proyectos sociales que vinculen la experiencia vivida con el sentir, evidenciando la marginación, exclusión y pobreza que se manifiesta en el día a día.

La Fundación Mariana de Jesús al ver que su Programa de Voluntariado ha recibido una gran acogida y cada vez son más los colaboradores que se unen a esta noble

causa, ha querido automatizar su proceso de administración de este programa, para agilizar el proceso de inscripción y organización de sus colaboradores.

1.2. Software libre: características

Software libre es aquel que respeta la libertad de los usuarios y la comunidad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el software libre es una cuestión de libertad, no de precio. **(3)**

Por lo tanto, se podría decir que el Software libre no es lo mismo que software gratis.

La definición de software libre no contempla un asunto de precio; un eslogan muy popular usado es "libre como en libertad, no como en cerveza gratis" o en inglés "Free as in freedom, not as in free beer" **(4)**

Para evitar confusiones, ciertas personas utilizan los términos "libre" y "gratis" para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia todo el resto del mundo. **(4)**

Las libertades que el Software Libre promueve deben ser para todos los usuarios tanto individualmente como en forma colectiva, y así permitirles controlar el programa y lo que este hace. Si los usuarios se encuentran con programas que impiden su control, se dice que dicho programa no es libre sino privativo, dando control total solamente al programador.

Características (3)

Un programa es software libre solo si los usuarios tienen estas cuatro libertades esenciales:

- La libertad de ejecutar el programa como se desea, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente¹ es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto permite ofrecer a toda la comunidad, la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si los usuarios tienen todas estas libertades de una manera adecuada. De lo contrario no es libre. Existen diversos esquemas de

¹ Código fuente, también llamado código base, es un texto que se ha escrito...

distribución que no son libres, y si bien podemos distinguirlos en base a cuánto les falta para llegar a ser libres, nosotros los consideramos contrarios a la ética a todos por igual.

Ventajas (5)

- El usuario no comete delito por tenerlo o usarlo.
- Amplísima gama y variedad de herramientas libres.
- Actualizaciones periódicas con alta frecuencia.
- 100% libre de virus.
- Altísimo nivel de estabilidad comprobada.
- Protege y defiende la Soberanía.
- Tiene una gran comunidad de apoyo y soporte.
- Diversidad de soluciones informáticas.
- Costo.
- Flexibilidad de las soluciones informáticas.
- Independencia tecnológica.

Desventajas (5)

- El hardware debe ser de calidad y estándares abiertos.
- Carece de una estructura ampliada de mercadeo (marketing).
- Algunas aplicaciones específicas no están en el mercado.
- Requiere profesionales debidamente calificados para la administración del sistema (es un sistema administrado).
- Dificultad en el intercambio de archivos.
- Algunas aplicaciones (bajo Linux) pueden llegar a ser algo complicadas de instalar.
- Inexistencia de garantía por parte del autor.
- Interfaces gráficas menos amigables.
- Poca estabilidad y flexibilidad en el campo de multimedia y juegos.
- Menor compatibilidad con el hardware.

1.3. Programación Orientada a Objetos: características

La programación² orientada a objetos o POO³ (OOP según sus siglas en inglés) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe

² Programación es el proceso de diseñar...

³ POO (Programación Orientada a Objetos)

una gran variedad de lenguajes de programación que soportan la orientación a objetos. (6)

Ventajas (7)

- Fomentar la reutilización y extensión del código.
- Permitir crear sistemas mucho más complejos.
- Relacionar de mejor forma el sistema al mundo real.
- Facilitar la creación de programas más visuales.
- Permitir la construcción de prototipos.
- Agilizar el desarrollo de software.
- Facilitar el trabajo en equipo.
- Facilitar el mantenimiento del software.

Desventajas (8)

- No todos los programas pueden ser modelados con exactitud por el modelo de objetos.
- Al forzar el lenguaje en el concepto de POO, se pierden algunas de las características de lenguajes útiles, como los "lenguajes funcionales".
- El concepto que un programador tiene de lo que constituye un objeto abstracto puede no coincidir con la visión de otro programador.
- Los objetos a menudo requieren una extensa documentación.

Lo interesante de la POO es que proporciona los conceptos y herramientas, con las cuales se puede modelar y representar el mundo real tan fielmente como sea posible.

Características (6)

Existe un acuerdo acerca de qué características contempla la "orientación a objetos". Las características que se detallan a continuación, son las más importantes:

Abstracción

Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción. La abstracción permite seleccionar las características más relevantes dentro de un conjunto e identificar los comportamientos más comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que nos permitan modelar la realidad o el problema que se quiere atacar.

Encapsulamiento

Significa reunir todos los elementos que puedan considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Lo que permitirá aumentar la cohesión de los componentes del sistema.

Modularidad

Es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales deberá ser tan independiente como sea posible de la aplicación en sí, y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan la modularidad de diversas formas.

Principio de ocultación

Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege las propiedades de un objeto impidiendo su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no puedan cambiar el estado interno de otro objeto de manera inesperada, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.

Polimorfismo

Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. Es decir, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

Herencia

Las clases no se encuentran aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente, agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más

de una clase se dice que hay una herencia múltiple; siendo de alta complejidad técnica por lo cual suele recurrirse a la herencia virtual para evitar la duplicación de datos.

Recolección de basura

Es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente los objetos que hayan quedado sin ninguna referencia, y por lo tanto desvincula la memoria asociada a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando.

1.4. Modelo en Cascada: características

En Ingeniería de software⁴ el desarrollo en cascada, también llamado modelo en cascada (denominado así por la posición de las fases en el desarrollo de esta, que parecen caer en cascada “por gravedad” hacia las siguientes fases), es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software⁵, de tal forma que para iniciar cada etapa se debe esperar la finalización de la etapa anterior. Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida. **(9)**

El más conocido, está basado en el ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

⁴ Ingeniería de software es la aplicación de un enfoque sistemático...

⁵ Desarrollo de software es una estructura aplicada al desarrollo...

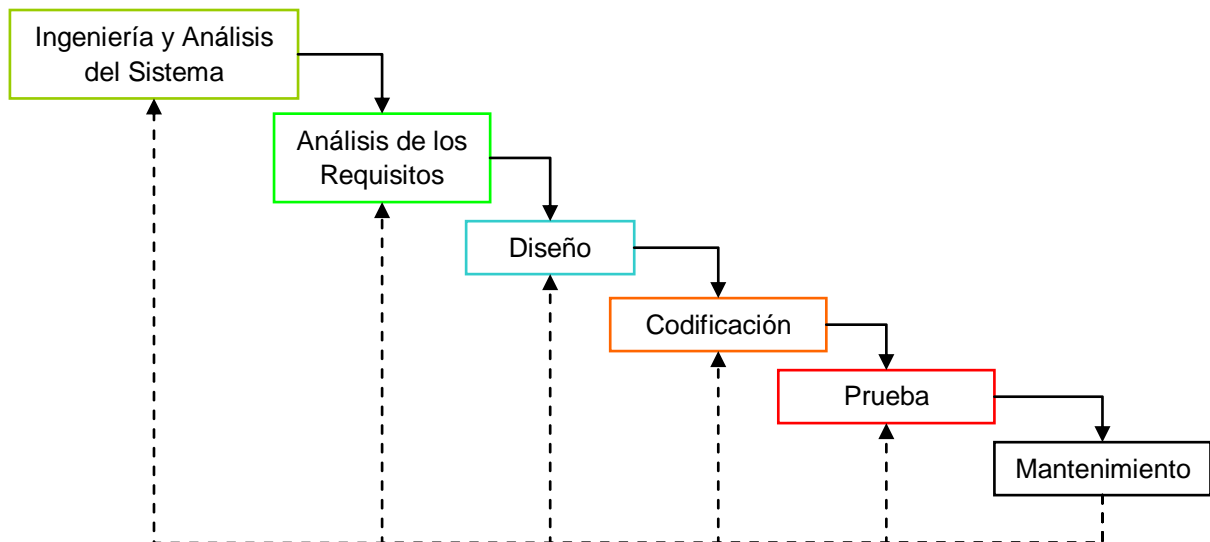


Figura 1: Gráfico en representación del modelo en cascada

Fuente: <http://portal-ingenieriadesoftware.wikispaces.com/Modelo+de+proceso+de+Software>

Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

Codificación: el diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

Prueba: una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debido a que se encuentren errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento. (11)

Ventajas (10)

- Fácil entendimiento e implementación.
- Ampliamente utilizado y conocido.
- Refuerza buenos hábitos: definir antes que diseñar, diseñar antes que codificar.
- Identifica entregables e hitos.
- Orientado a documentos.
- Funciona bien en productos maduros y equipos débiles.

Desventajas (10)

- No aprovecha la iteración, ni el desarrollo exploratorio.
- No espera requerimientos definidos completamente al inicio del proyecto.
- Rara vez, un proyecto sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso.
- El software es entregado tarde en el proyecto y esto hace que se detecten errores graves muy tarde.
- Hacer cambios es difícil y costoso.

1.5. Arquitectura de Programación en Tres Capas: características

La arquitectura 3 capas o programación 3 capas consiste literalmente en separar un proyecto en Capa de Presentación, Capa de Negocio y Capa de Datos. Esto permite distribuir el trabajo de creación de una aplicación por niveles; y cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API⁶ que existe entre niveles. (12)

Ventajas (12)

- El desarrollo se puede llevar a cabo en varios niveles.
- Aplicaciones más robustas debido al encapsulamiento.
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).

⁶ API (Application Programming Interface) es un conjunto...

Desventajas (13)

- Complejidad.
- Realización de trabajo innecesario o redundante entre capas.
- Dificultad al corregir la granularidad de las capas.

Capas Y Niveles (12)

Capa de Presentación: Esta es la parte que ve el usuario, las pantallas que se le muestra para que interactúe con el programa (también se la conoce como “capa de usuario”), comunicándole la información y recolectando la información suministrada por el usuario en un mínimo de proceso (realiza validaciones para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio llevando y trayendo los datos o registros necesarios, es la interfaz gráfica del programa y debe ser lo más amena posible para una mejor comunicación con el usuario.

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todos los procesos que deben realizarse.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

1.6. Herramientas y Tecnologías

1.6.1 JAVA (14)

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención, es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run

anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Características (15)

Independencia de la plataforma

Java funciona con las principales plataformas de hardware y sistemas operativos, o bien con el software JVM (Java Virtual Machine) directamente desde Oracle, a través de uno de los muchos partners del ecosistema de Java, o como parte de la comunidad OpenJDK.

Alto rendimiento

HotSpot y JRockit son ejemplos de tecnologías de equipos virtuales de interpretación dinámica (JIT) y de eficacia probada que hacen de Java, uno de los entornos de programación más rápidos. Las optimizaciones integradas para entornos multiproceso lo hacen aún más rápido.

Fácil de aprender

Java es el lenguaje de programación preferido por las universidades y las instituciones de enseñanza de todo el mundo.

El modelo de Java para la gestión de la memoria, los procesos múltiples y la gestión de excepciones lo convierte en un lenguaje muy eficaz para los desarrolladores nuevos y para los más experimentados.

Basado en estándares

El lenguaje Java y la tecnología relacionada evolucionan a través de Java Community Process, un mecanismo que permite el desarrollo de especificaciones técnicas para la tecnología Java.

Prevalencia mundial

Java es la plataforma de aplicaciones más popular del planeta, proporcionando un interesante ecosistema de desarrolladores que son impulsados por herramientas eficaces, libros, bibliotecas, muestras de código y mucho más.

Entornos de ejecución coherentes

Java permite realizar despliegues confiables, con entornos de tiempo de ejecución que van de Java SE en equipos de sobremesa a Java SE for Embedded Devices y Oracle Java Micro Edition Embedded Client

Optimizado para los dispositivos integrados

Java SE for Embedded Devices incluye compatibilidad con requisitos clave, como la compatibilidad con procesadores integrados, la gestión de potencia, los despliegues con huella pequeña y mucho más.

Oracle Java ME Embedded Client se basa en Connected Device Configuration (CDC), un subconjunto de la plataforma Java SE, y proporciona rendimiento Java para los dispositivos con recursos restringidos.

Aplicaciones portátiles con alto rendimiento

Java alcanza un rendimiento nativo y proporciona portabilidad en una amplia gama de procesadores y sistemas operativos integrados.

Modelo con seguridad probada

Java ofrece un entorno de aplicaciones avanzado con un alto nivel de seguridad que es idóneo para las aplicaciones de red.

1.6.2. MySQL (16)

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos, deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

Principales características (17)

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soportar gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovechar la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

Ventajas (17)

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad en la configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

Desventajas (17)

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

CAPÍTULO 2.- ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA

2.1. Análisis de situación actual

En la actualidad, la FMDJ⁷ administra los perfiles de todos los voluntarios y estudiantes que participan en los diferentes proyectos del Programa Social de Voluntariado, de forma manual mediante plantillas en Excel. Estas plantillas son llenadas a partir de información recolectada personalmente en entrevistas, o de documentos físicos que contienen la información específica de la participación de cada estudiante o voluntario en los diferentes proyectos.

El proceso que se maneja en la actualidad, comienza con el ingreso del voluntario o estudiante en el proyecto que se haya escogido. Para esto se llena un formulario, a mano, con todos los datos del participante, que luego será ingresado en una plantilla de Excel.

Luego se firma una Carta de Compromiso, indicando las fechas cuando se inicia el proyecto y hasta cuándo será la participación. Estas fechas también se las incluirá en la plantilla de Excel.

Para el registro de asistencia de cada participante a los diferentes proyectos, se llenan a mano, unos formularios donde se especifica el proyecto en el que se está envuelto y las fechas con las horas de entrada y de salida diarias. Cada día se llena las horas respectivas y se debe firmar.

Cuando las horas de participación en un proyecto se culminan, se realiza una evaluación del participante. Para esto se debe recolectar todos los formularios de registro de asistencia que el participante haya llenado, y luego las horas son contabilizadas a mano para generar un informe del desempeño del participante.

Al tener tantos participantes en los proyectos de la FMDJ, se generan bastantes documentos físicos, y el manejo de la información se vuelve complejo. Pasar toda esa información a mano, puede resultar una labor que tarde mucho tiempo y a la vez agotadora, por lo que puede haber errores al momento de ingresarla y perderse ciertos datos que son importantes.

También el problema que se presenta, al tener una gran cantidad de documentos y tratar de archivarlos, genera molestias cuando es necesario buscar información de algún participante en particular y tener que buscar en todos los documentos. Incluso con tantos papeles, se corre el riesgo de perder algún documento que tenga información importante.

⁷ FMDJ (Fundación Mariana de Jesús)

Todo esto ha provocado que la FMDJ opte por utilizar una aplicación que automatice todo el proceso de administración del programa de voluntariado.

2.2. Levantamiento de requerimientos del sistema

Para realizar el proceso de levantamiento de requerimientos del sistema, se llevaron a cabo varias visitas a la FMDJ, en donde se mantuvo una serie de entrevistas con la coordinadora del Programa de Voluntariado de la FMDJ, y después de discutir las diferentes actividades que se realizan y la forma como se administran, se logró establecer un conjunto de requerimientos necesarios para la implementación del sistema.

2.2.1. Funcionalidades del sistema

- F1: El sistema contará con una autenticación para ingresar a la aplicación.
- F2: El sistema será capaz de administrar usuarios.
- F3: El sistema será capaz de administrar proyectos.
- F4: El sistema será capaz de administrar subproyectos.
- F5: El sistema será capaz de administrar voluntarios.
- F6: El sistema será capaz de administrar estudiantes.
- F7: El sistema será capaz de administrar profesiones.
- F8: El sistema será capaz de administrar universidades.
- F9: El sistema será capaz de administrar facultades.
- F10: El sistema será capaz de administrar registros de asistencia.
- F11: El sistema será capaz de generar reportes.

2.2.1.1. Autenticación de usuario

La Autenticación de usuario consiste en la verificación del usuario que ingresa al sistema por medio de un código de usuario y una contraseña únicos.

2.2.1.2. Administración de usuarios

La Administración de usuarios servirá para: crear, modificar, consultar y eliminar usuarios que puedan ingresar al sistema.

2.2.1.3. Administración de proyectos

La Administración de proyectos servirá para: crear, modificar, consultar y eliminar proyectos que posea la FMDJ en su Programa de Voluntariado.

2.2.1.4. Administración de subproyectos

La Administración de subproyectos servirá para: crear, modificar, consultar y eliminar subproyectos que pertenezcan a los proyectos principales de la FMDJ en su Programa de Voluntariado.

2.2.1.5. Administración de voluntarios

La Administración de voluntarios servirá para: crear, modificar, consultar y eliminar voluntarios que posea la FMDJ en su Programa de Voluntariado.

2.2.1.6. Administración de estudiantes

La Administración de estudiantes servirá para: crear, modificar, consultar y eliminar estudiantes que posea la FMDJ en su Programa de Voluntariado.

2.2.1.7. Administración de profesiones

La Administración de profesiones servirá para: crear, modificar, consultar y eliminar las profesiones que posean los voluntarios que estén participando para la FMDJ en su Programa de Voluntariado.

2.2.1.8. Administración de universidades

La Administración de universidades servirá para: crear, modificar, consultar y eliminar las universidades de los estudiantes que estén participando para la FMDJ en su Programa de Voluntariado.

2.2.1.9. Administración de facultades

La Administración de facultades servirá para poder crear, modificar, consultar y eliminar las facultades pertenecientes a las universidades de los estudiantes que estén participando para la FMDJ en su Programa de Voluntariado.

2.2.1.10. Administración de registros de asistencia

La Administración de registros de asistencia servirá para: crear, modificar, consultar y eliminar los registros de asistencia de los voluntarios y estudiantes que estén participando en los diferentes proyectos para la FMDJ en su Programa de Voluntariado.

2.2.1.11. Generación de reportes

La generación de reportes servirá para: crear reportes de la información de los voluntarios o estudiantes y saber en qué proyectos están envueltos y como es su participación en ellos.

2.2.2. Requerimientos funcionales

2.2.2.1. Diagrama general

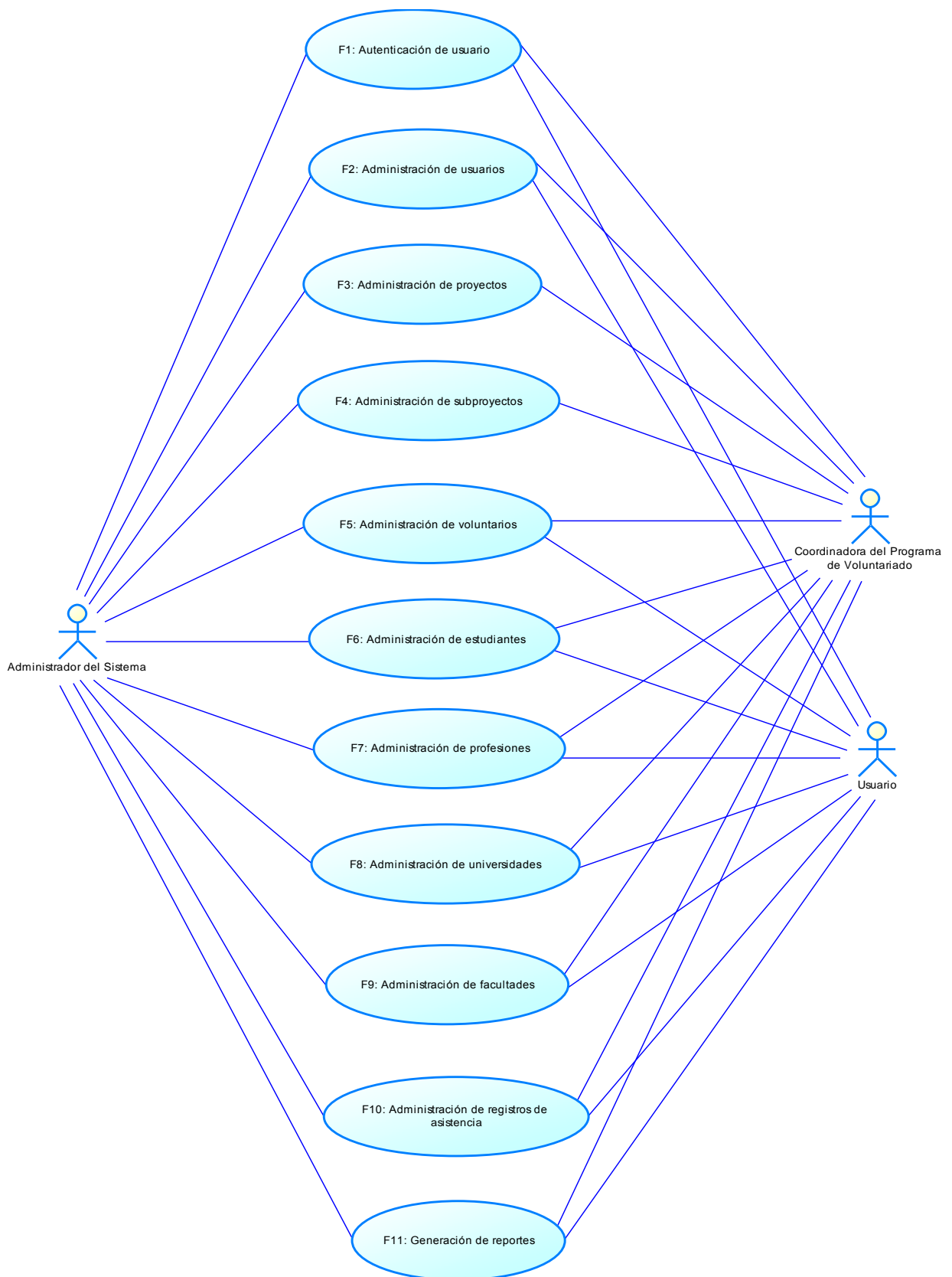


Figura 2.1: Diagrama de Casos de Uso General

2.2.2.2. Diagramas de casos de uso

F1: Autenticación de usuario

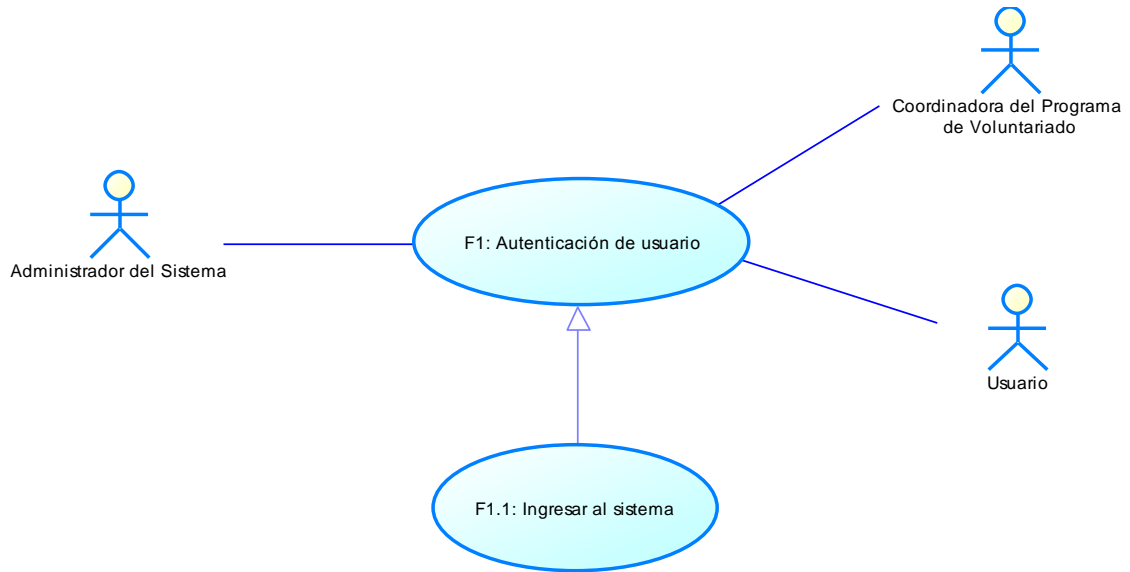


Figura 2.2: Diagrama de Casos de Uso Autenticación de usuario

DETALLE:

F1.1: Ingresar al sistema

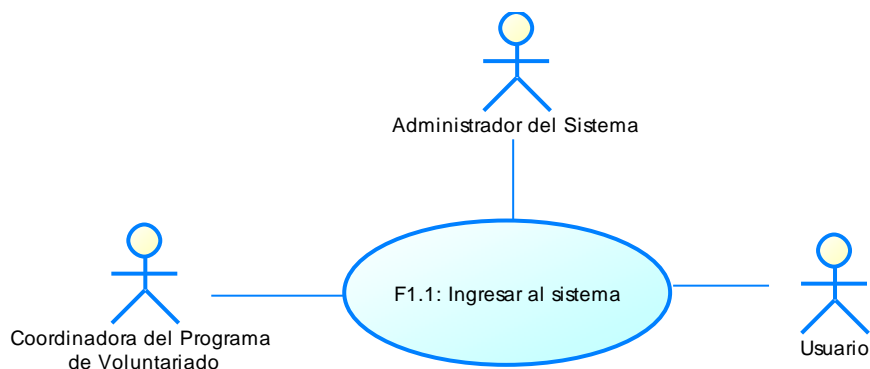


Figura 2.3: Diagrama a Detalle Ingresar al sistema

Descripción: Se podrá ingresar al sistema por medio de un código de usuario y una contraseña únicos.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Inicia la aplicación	2	Presenta pantalla de registro de usuario y contraseña	

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

3	Ingresar código del usuario			
4	Ingresar contraseña del usuario			
5	Pulsar botón Ingresar	6	Validar los datos ingresados	E1, E2, E3
		7	Presentar pantalla inicial de la aplicación	

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F2: Administración de usuarios

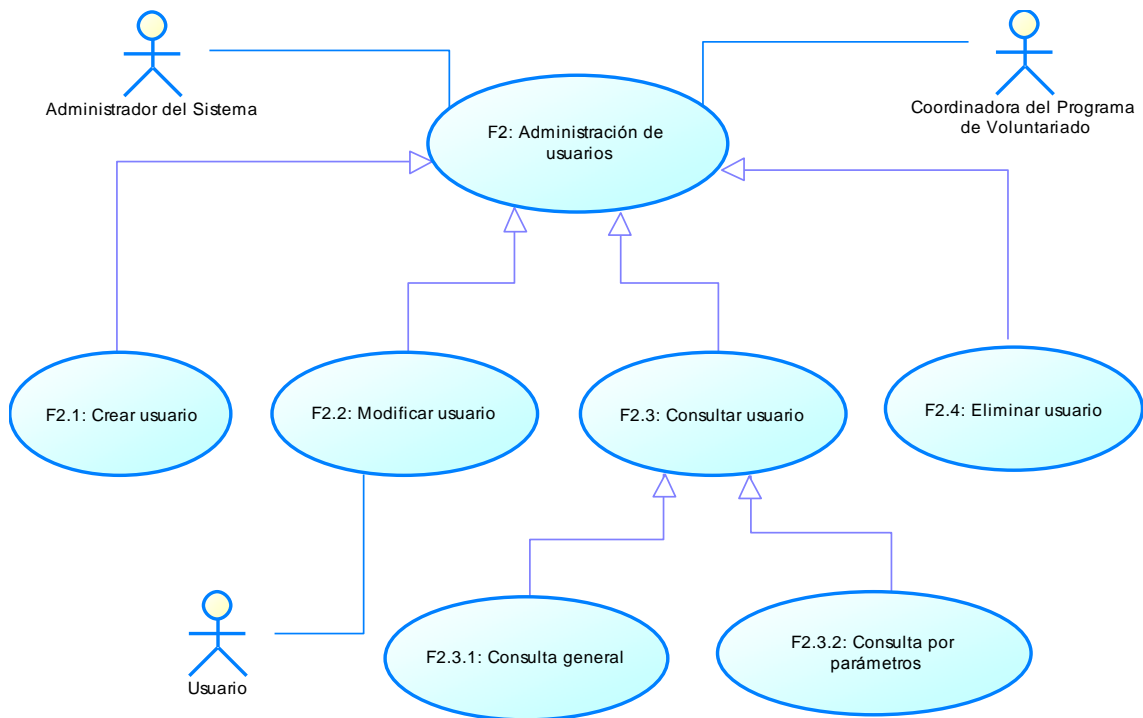


Figura 2.4: Diagrama de Casos de Uso Administración de usuarios

DETALLE:

F2.1: Crear usuario

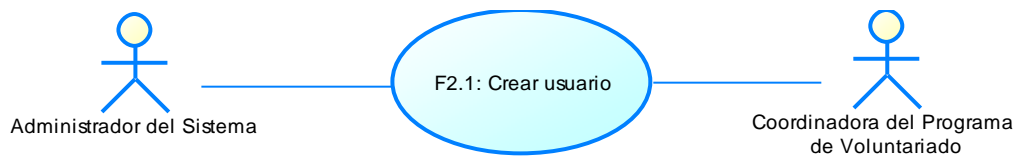


Figura 2.5: Diagrama a Detalle Crear Usuario

Descripción: Se podrán crear los usuarios que tendrán acceso al sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Usuarios	4	Presenta pantalla de ingreso de usuarios	
5	Ingresar cédula del usuario que va a ser creado	6	Verifica si no existe esa cédula	E3
		7	Desbloquea los demás campos	
		8	Desbloquea el botón de Guardar	
9	Ingresar los datos del usuario			
10	Pulsar el botón Guardar	11	Almacena los datos ingresados	E1, E2, E3
		12	Presenta un mensaje indicando que el usuario se ha creado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
6	F2.2 o F2.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F2.2: Modificar usuario

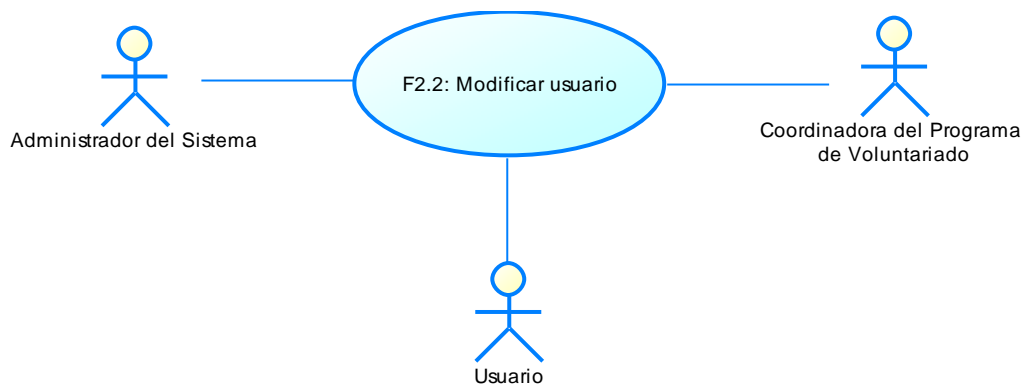


Figura 2.6: Diagrama a Detalle Modificar Usuario

Descripción: Se podrán modificar ciertos campos de la información o toda la información de los usuarios existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Usuarios	4	Presenta pantalla de ingreso de usuarios	
5	Ingresa cédula del usuario que va a ser modificado	6	Verifica si existe esa cédula	E3
		7	Desbloquea los demás campos	
		8	Carga todos los datos del usuario	E3
		9	Asigna la etiqueta Modificar en el botón	
10	Modifica los datos del usuario que se desea cambiar			
11	Pulsa el botón Modificar	12	Almacena los datos ingresados	E1, E2, E3
		13	Presenta un mensaje indicando que el usuario se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
6	F2.1 o F2.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F2.3: Consultar usuario

F2.3.1: Consulta general

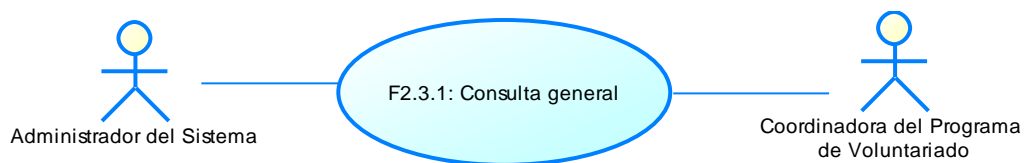


Figura 2.7: Diagrama a Detalle Consulta general Usuario

Descripción: Se podrán consultar todos los usuarios existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Usuarios	6	Carga todos los usuarios que existen en el sistema	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F2.3.2: Consulta por parámetros

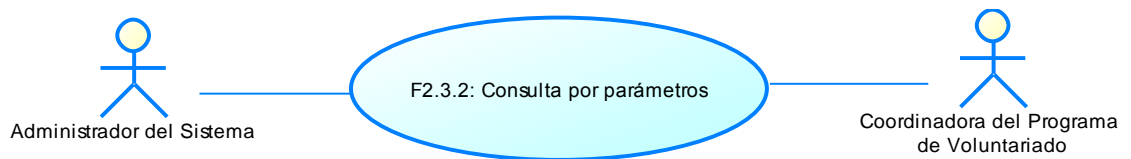


Figura 2.8: Diagrama a Detalle Consulta por parámetros Usuario

Descripción: Se podrán consultar los usuarios que existen en el sistema por medio de parámetros que se ingresen.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona Parámetros	4	Presenta pantalla de consulta por parámetros	
5	Selecciona la pestaña de Usuarios			
6	Ingresa los parámetros de búsqueda			
7	Pulsa el botón Consultar	8	Carga los usuarios que contengan los parámetros ingresados	E1, E2

Excepciones:

Código	Mensaje	Alternativa
E1	Parámetros vacíos	Llenar al menos un parámetro
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F2.4: Eliminar usuario

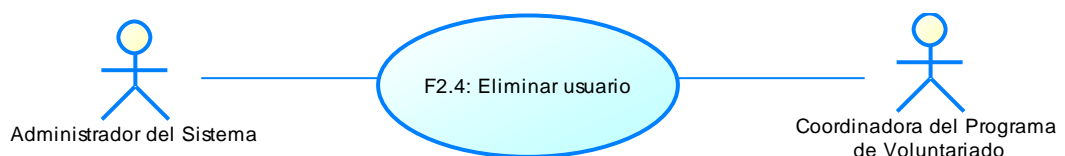


Figura 2.9: Diagrama a Detalle Eliminar Usuario

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

Descripción: Se podrán eliminar los usuarios existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Usuarios	4	Presenta pantalla de ingreso de usuarios	
5	Ingresa cédula del usuario que va a ser eliminado	6	Verifica si existe esa cédula	E1
		7	Carga todos los datos del usuario	E1
		8	Aparece otro botón con la etiqueta Eliminar	
9	Pulsa el botón Eliminar	10	Presenta un mensaje para confirmar la eliminación del usuario	
11	Pulsa Sí en el mensaje	12	Presenta un mensaje indicando que el usuario se ha eliminado exitosamente	E1

Flujo Alterno:

Paso	Caso de uso
6	F2.1 o F2.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F3: Administración de proyectos

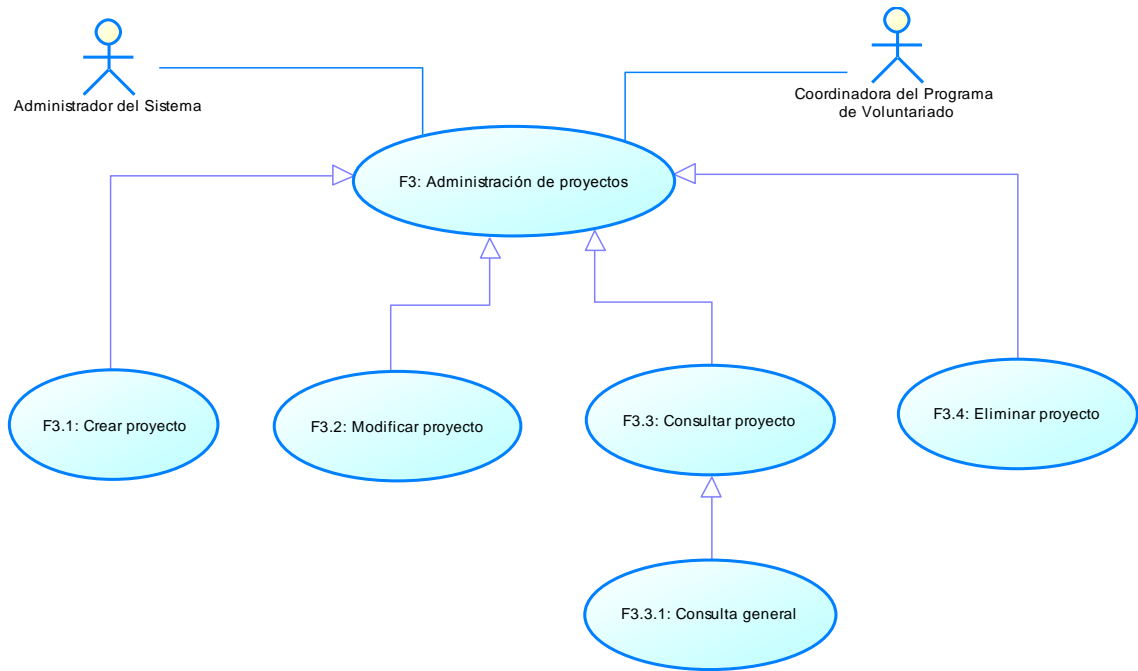


Figura 2.10: Diagrama de Casos de Uso Administración de proyectos

DETALLE:

F3.1: Crear proyecto

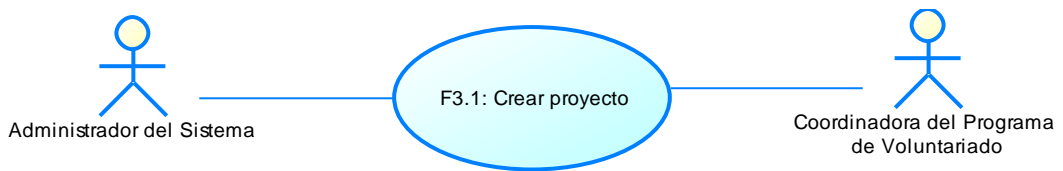


Figura 2.11: Diagrama a Detalle Crear proyecto

Descripción: Se podrán crear los diferentes proyectos del Programa de Voluntariado.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Proyectos	4	Presenta pantalla de ingreso de proyectos	
		5	Carga todos los proyectos existentes en el sistema	E2

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

		6	Desbloquea los demás campos	
		8	Desbloquea el botón de Guardar	
9	Ingresar el nombre del proyecto			
10	Pulsa el botón Guardar	11	Almacena los datos ingresados	E1, E2
		12	Presenta un mensaje indicando que el proyecto se ha creado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
5	F3.2 o F3.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F3.2: Modificar proyecto

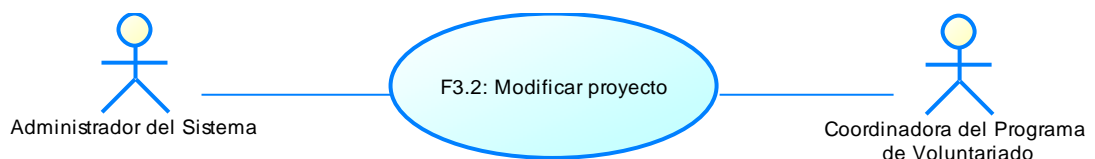


Figura 2.12: Diagrama a Detalle Modificar proyecto

Descripción: Se podrán modificar ciertos campos de la información o toda la información de los proyectos existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Proyectos	4	Presenta pantalla de ingreso de proyectos	
		5	Carga todos los proyectos existentes en el sistema	E2
		6	Desbloquea los demás campos	
7	Selecciona el proyecto que	8	Carga el nombre del	

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	va a ser modificado		proyecto	
		9	Asigna la etiqueta Modificar en el botón	
10	Modifica el nombre del proyecto que se desea cambiar			
11	Pulsa el botón Modificar	12	Almacena los datos ingresados	E1, E2
		13	Presenta un mensaje indicando que el proyecto se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
5	F3.1 o F3.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F3.3: Consultar proyecto

F3.3.1: Consulta general

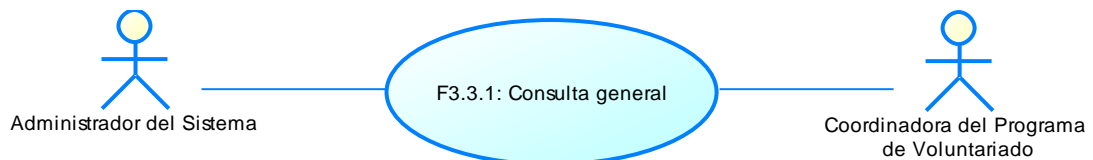


Figura 2.13: Diagrama a Detalle Consulta general proyecto

Descripción: Se podrán consultar todos los proyectos existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Proyectos/Subproyectos	6	Carga todos los proyectos que existen en el sistema	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F3.4: Eliminar proyecto

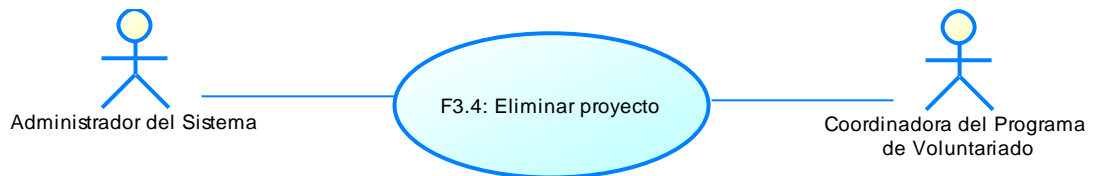


Figura 2.14: Diagrama a Detalle Eliminar proyecto

Descripción: Se podrán eliminar los proyectos existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Proyectos	4	Presenta pantalla de ingreso de proyectos	
		5	Carga todos los proyectos existentes en el sistema	E2
		6	Desbloquea los demás campos	
7	Selecciona el proyecto que va a ser eliminado	8	Carga el nombre del proyecto	
		9	Aparece otro botón con la etiqueta Eliminar	
10	Pulsa el botón Eliminar	11	Presenta un mensaje para confirmar la eliminación del proyecto	
12	Pulsa Sí en el mensaje	13	Presenta un mensaje indicando que el proyecto se ha eliminado exitosamente	E1

Flujo Alterno:

Paso	Caso de uso
5	F3.1 o F3.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F4: Administración de subproyectos

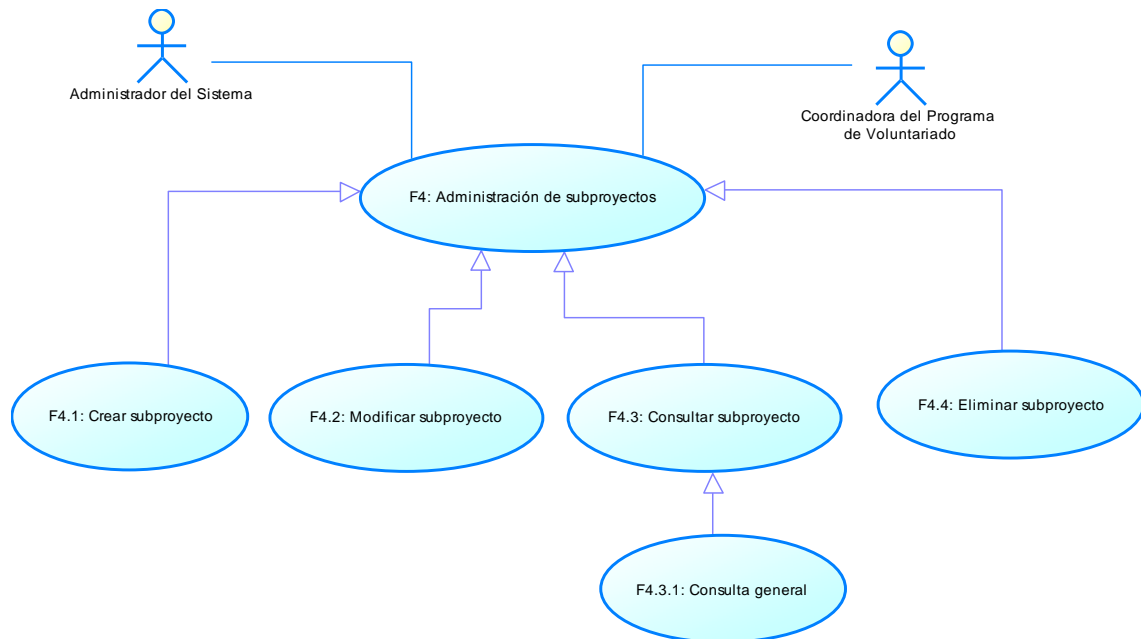


Figura 2.15: Diagrama de Casos de Uso Administración de subproyectos

DETALLE:

F4.1: Crear subproyecto

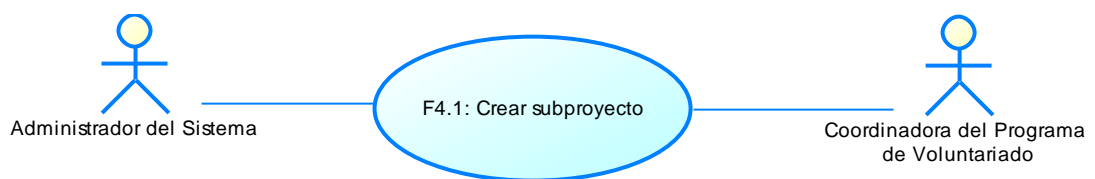


Figura 2.16: Diagrama a Detalle Crear subproyecto

Descripción: Se podrán crear los diferentes subproyectos que pertenezcan a los proyectos principales del Programa de Voluntariado.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción	2	Presenta lista con todas las	

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	Administración del menú principal		administraciones	
3	Selecciona Subproyectos	4	Presenta pantalla de ingreso de subproyectos	
		5	Carga todos los proyectos que existen en el sistema	E2
6	Selecciona el proyecto al cual quiere agregar los subproyectos	7	Carga los subproyectos asociados al proyecto seleccionado	E2
		8	Desbloquea los demás campos	
		9	Desbloquea el botón de Guardar	
10	Ingresar el nombre del subproyecto			
11	Pulsa el botón Guardar	12	Almacena los datos ingresados	E1, E2
		13	Presenta un mensaje indicando que el subproyecto se ha creado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
7	F4.2 o F4.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F4.2: Modificar subproyecto

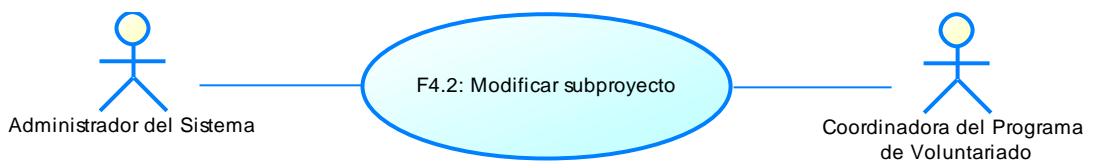


Figura 2.17: Diagrama a Detalle Modificar subproyecto

Descripción: Se podrán modificar ciertos campos de la información o toda la información de los subproyectos existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Subproyectos	4	Presenta pantalla de ingreso de subproyectos	
		5	Carga todos los proyectos que existen en el sistema	E2
6	Selecciona el proyecto del cual quiere modificar los subproyectos	7	Carga los subproyectos asociados al proyecto seleccionado	E2
8	Selecciona el subproyecto que va a ser modificado	9	Carga el nombre del subproyecto	
		10	Asigna la etiqueta Modificar en el botón	
11	Modifica los datos del subproyecto que se desea cambiar			
12	Pulsa el botón Modificar	13	Almacena los datos ingresados	E1, E2
		14	Presenta un mensaje indicando que el subproyecto se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
7	F4.1 o F4.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F4.3: Consultar usuario

F4.3.1: Consulta general

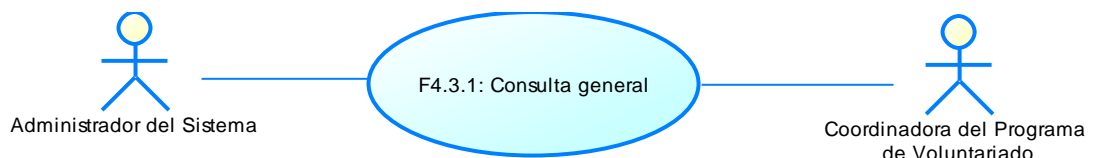


Figura 2.18: Diagrama a Detalle Consulta general subproyecto

Descripción: Se podrán consultar todos los subproyectos existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Proyectos/Subproyectos	6	Carga todos los proyectos que existen en el sistema	E1
7	Selecciona el proyecto del cual quiere consultar los subproyectos	8	Carga los subproyectos asociados al proyecto seleccionado	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F4.4: Eliminar subproyecto

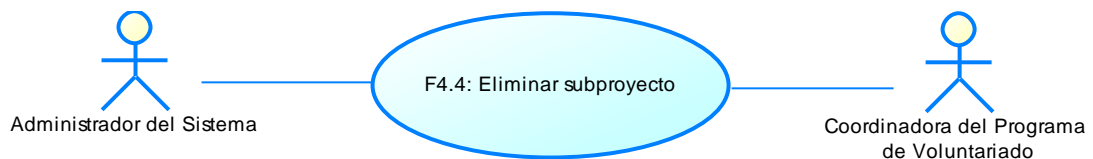


Figura 2.19: Diagrama a Detalle Eliminar subproyecto

Descripción: Se podrán eliminar los subproyectos existentes en el sistema.

Actores: Administrador del Sistema y Coordinadora del Programa de Voluntariado

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Subproyectos	4	Presenta pantalla de ingreso de subproyectos	
		5	Carga todos los proyectos que existen en el sistema	E1
6	Selecciona el proyecto del cual quiere eliminar los subproyectos	7	Carga los subproyectos asociados al proyecto seleccionado	E1
8	Selecciona el subproyecto que va a ser eliminado	9	Carga el nombre del subproyecto	

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

		10	Aparece otro botón con la etiqueta Eliminar	
11	Pulsa el botón Eliminar	12	Presenta un mensaje para confirmar la eliminación del proyecto	
13	Pulsa Sí en el mensaje	14	Presenta un mensaje indicando que el proyecto se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
7	F4.1 o F4.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F5: Administración de voluntarios

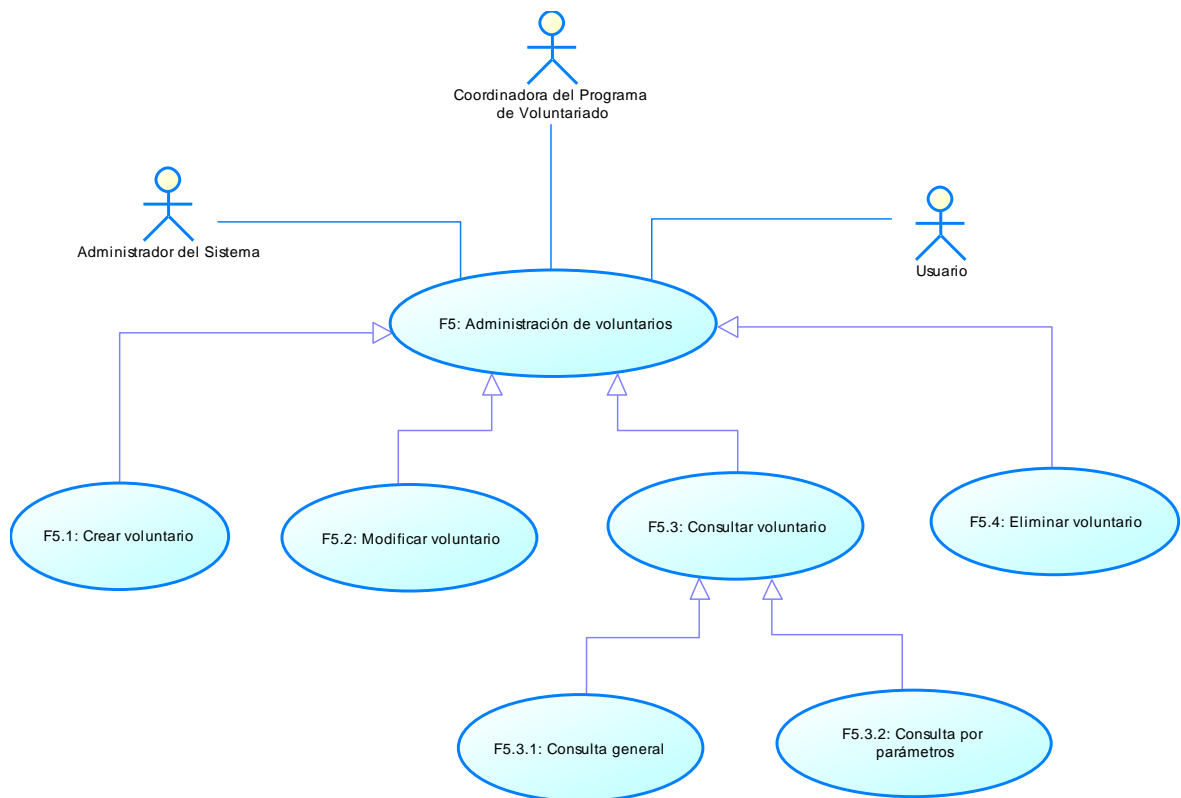


Figura 2.20: Diagrama de Casos de Uso Administración de voluntarios

DETALLE:

F5.1: Crear voluntario

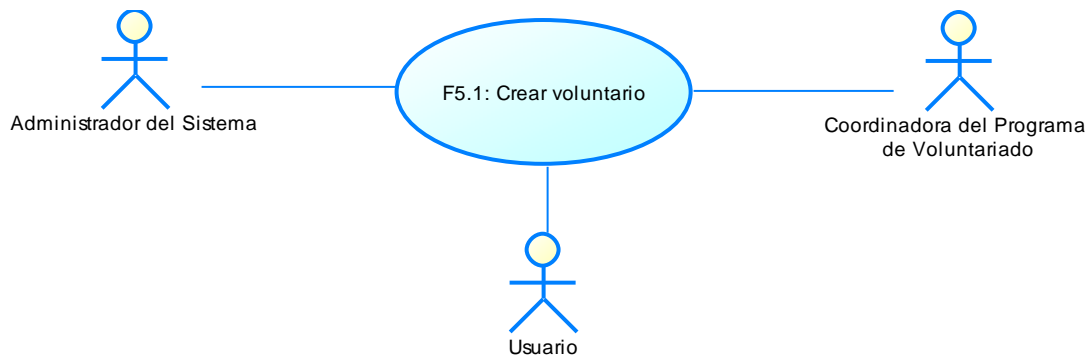


Figura 2.21: Diagrama a Detalle Crear voluntario

Descripción: Se podrán crear los voluntarios que participen en el Programa de Voluntariado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Voluntarios	4	Presenta pantalla de ingreso de voluntarios	
5	Ingresar cédula del voluntario que va a ser creado	6	Verifica si no existe esa cédula	E3
		7	Desbloquea los demás campos	
		8	Desbloquea el botón de Guardar	
9	Ingresar los datos del voluntario			
10	Pulsar el botón Guardar	11	Almacena los datos ingresados	E1, E2, E3
		12	Presenta un mensaje indicando que el voluntario se ha creado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
6	F5.2 o F5.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes

E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F5.2: Modificar voluntario

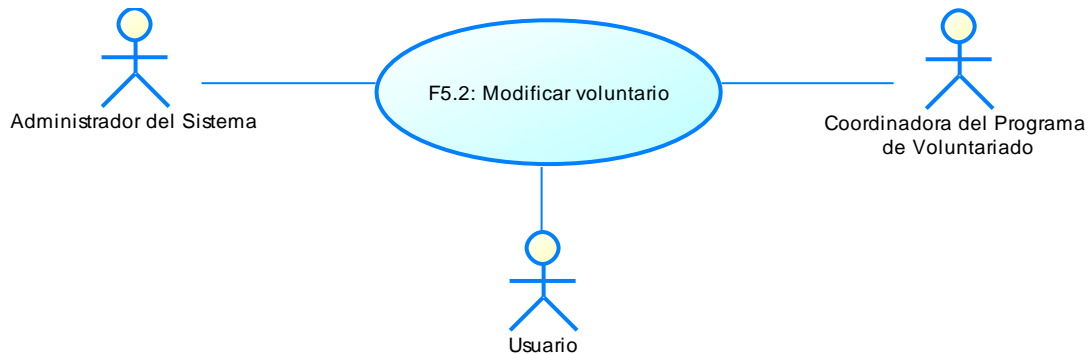


Figura 2.22: Diagrama a Detalle Modificar voluntario

Descripción: Se podrán modificar ciertos campos de la información o toda la información de los voluntarios existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Voluntarios	4	Presenta pantalla de ingreso de voluntarios	
5	Ingresar cédula del voluntario que va a ser modificado	6	Verifica si existe esa cédula	E3
		7	Desbloquea los demás campos	
		8	Carga todos los datos del voluntario	E3
		9	Asigna la etiqueta Modificar en el botón	
10	Modifica los datos del voluntario que se desea cambiar			
11	Pulsa el botón Modificar	12	Almacena los datos ingresados	E1, E2, E3
		13	Presenta un mensaje indicando que el voluntario se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
6	F5.1 o F5.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F5.3: Consultar voluntario

F5.3.1: Consulta general

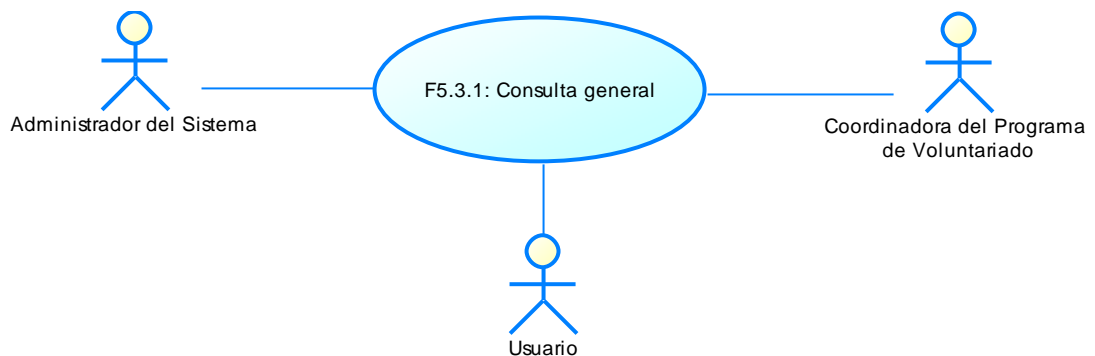


Figura 2.23: Diagrama a Detalle Consulta general voluntario

Descripción: Se podrán consultar todos los voluntarios existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Voluntarios	6	Carga todos los voluntarios que existen en el sistema	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F5.3.2: Consulta por parámetros

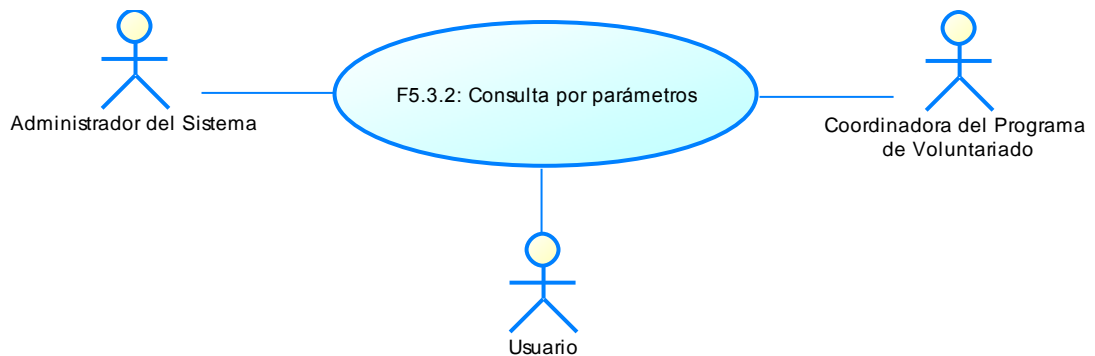


Figura 2.24: Diagrama a Detalle Consulta por parámetros voluntario

Descripción: Se podrán consultar los voluntarios que existen en el sistema por medio de parámetros que se ingresen.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona Parámetros	4	Presenta pantalla de consulta por parámetros	
5	Selecciona la pestaña de Voluntarios			
6	Ingresar los parámetros de búsqueda			
7	Pulsa el botón Consultar	8	Carga los voluntarios que contengan los parámetros ingresados	E1, E2

Excepciones:

Código	Mensaje	Alternativa
E1	Parámetros vacíos	Llenar al menos un parámetro
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F5.4: Eliminar voluntario

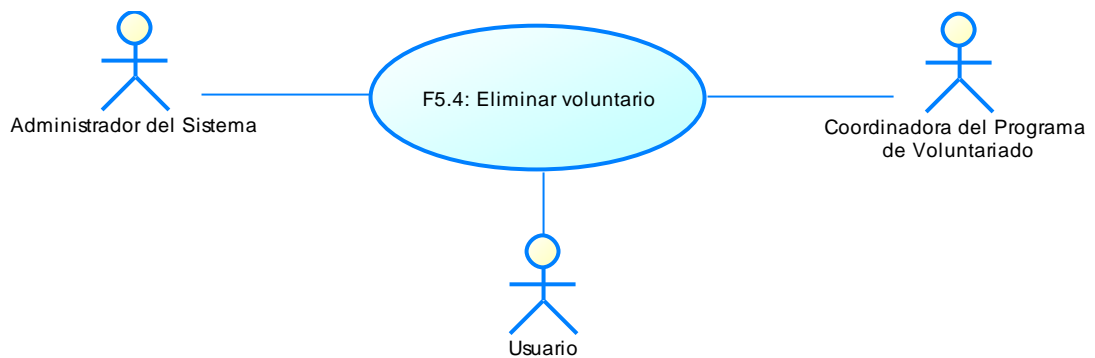


Figura 2.25: Diagrama a Detalle Eliminar voluntario

Descripción: Se podrán eliminar los voluntarios existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Voluntarios	4	Presenta pantalla de ingreso de voluntarios	
5	Ingresar cédula del voluntario que va a ser eliminado	6	Verifica si existe esa cédula	E1
		7	Carga todos los datos del voluntario	E1
		8	Aparece otro botón con la etiqueta Eliminar	
9	Pulsa el botón Eliminar	10	Presenta un mensaje para confirmar la eliminación del voluntario	
11	Pulsa Sí en el mensaje	12	Presenta un mensaje indicando que el voluntario se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
6	F5.1 o F5.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F6: Administración de estudiantes

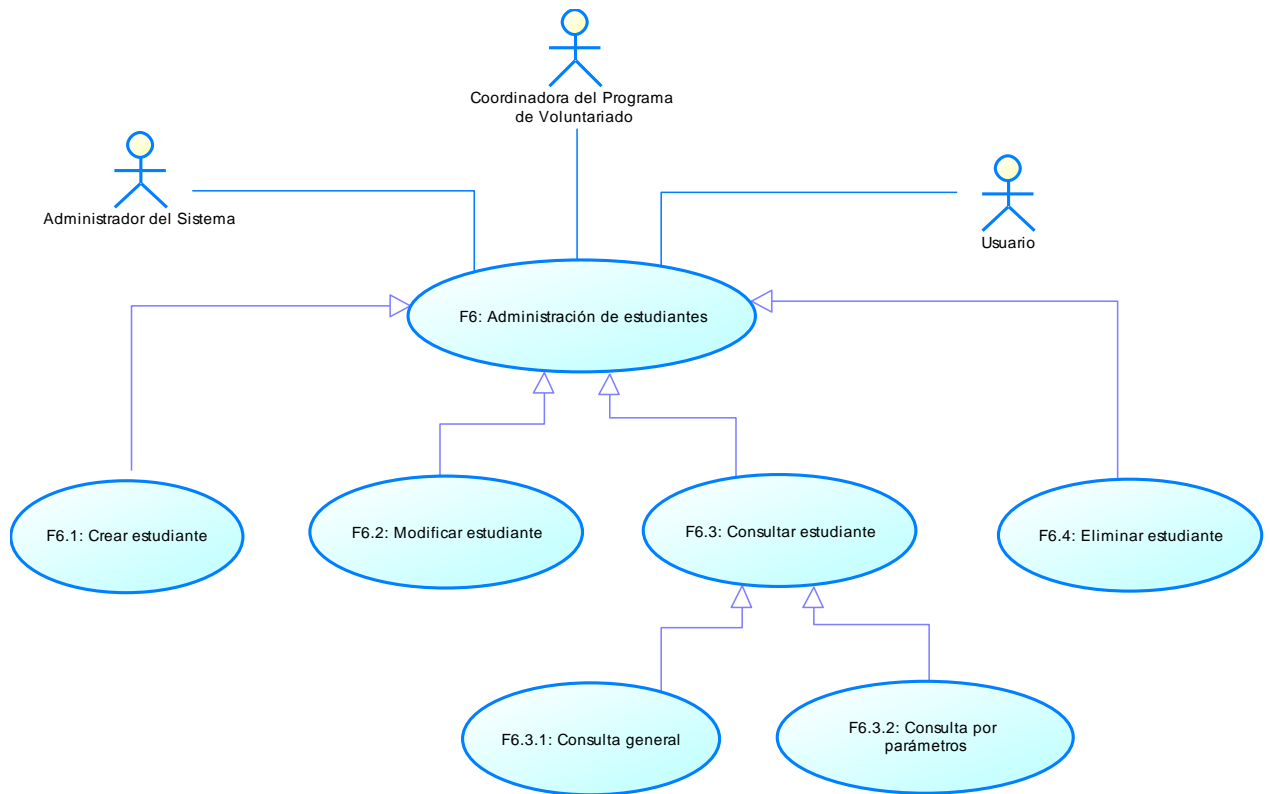


Figura 2.26: Diagrama de Casos de Uso Administración de estudiantes

DETALLE:

F6.1: Crear estudiante

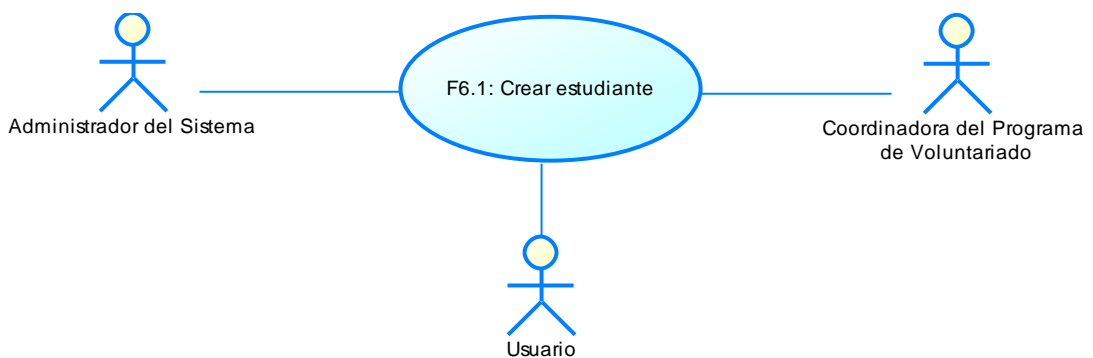


Figura 2.27: Diagrama a Detalle Crear estudiante

Descripción: Se podrán crear los estudiantes que participen en el Programa de Voluntariado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Estudiantes	4	Presenta pantalla de ingreso de estudiantes	
5	Ingresa cédula del estudiante que va a ser creado	6	Verifica si no existe esa cédula	E3
		7	Desbloquea los demás campos	
		8	Desbloquea el botón de Guardar	
9	Ingresa los datos del estudiante			
10	Pulsa el botón Guardar	11	Almacena los datos ingresados	E1, E2, E3
		12	Presenta un mensaje indicando que el estudiante se ha creado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
6	F6.2 o F6.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F6.2: Modificar estudiante

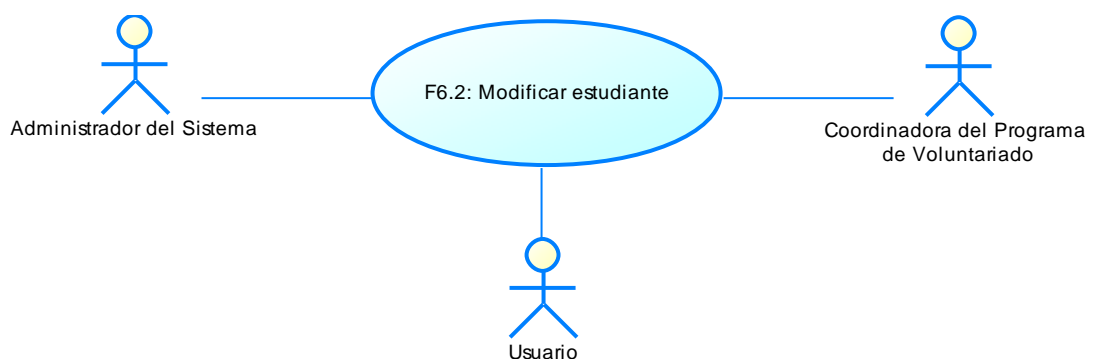


Figura 2.28: Diagrama a Detalle Modificar estudiante

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

Descripción: Se podrán modificar ciertos campos de la información o toda la información de los estudiantes existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Estudiantes	4	Presenta pantalla de ingreso de estudiantes	
5	Ingresa cédula del estudiante que va a ser modificado	6	Verifica si existe esa cédula	E3
		7	Desbloquea los demás campos	
		8	Carga todos los datos del estudiante	E3
		9	Asigna la etiqueta Modificar en el botón	
10	Modifica los datos del estudiante que se desea cambiar			
11	Pulsa el botón Modificar	12	Almacena los datos ingresados	E1, E2, E3
		13	Presenta un mensaje indicando que el estudiante se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
6	F6.1 o F6.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F6.3: Consultar estudiante

F6.3.1: Consulta general

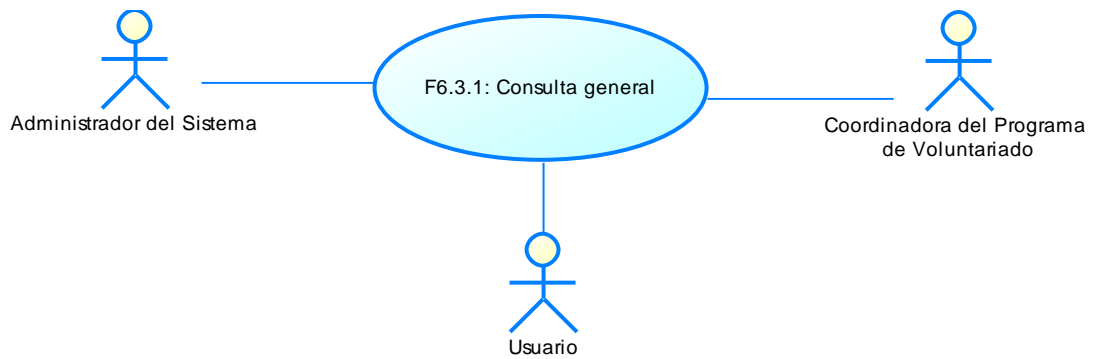


Figura 2.29: Diagrama a Detalle Consulta general estudiante

Descripción: Se podrán consultar todos los estudiantes existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Estudiantes	6	Carga todos los estudiantes que existen en el sistema	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F6.3.2: Consulta por parámetros

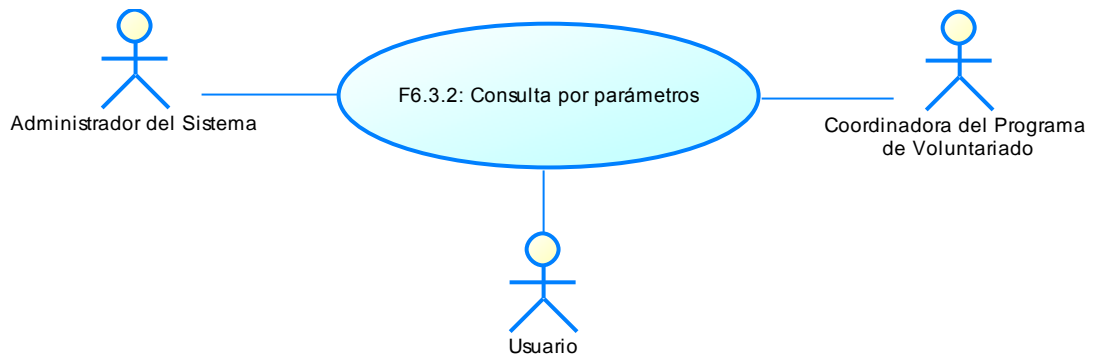


Figura 2.30: Diagrama a Detalle Consultar por parámetros estudiante

Descripción: Se podrán consultar los estudiantes que existen en el sistema por medio de parámetros que se ingrese.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona Parámetros	4	Presenta pantalla de consulta por parámetros	
5	Selecciona la pestaña de Estudiantes			
6	Ingresa los parámetros de búsqueda			
7	Pulsa el botón Consultar	8	Carga los estudiantes que contengan los parámetros ingresados	E1, E2

Excepciones:

Código	Mensaje	Alternativa
E1	Parámetros vacíos	Llenar al menos un parámetro
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F6.4: Eliminar estudiante

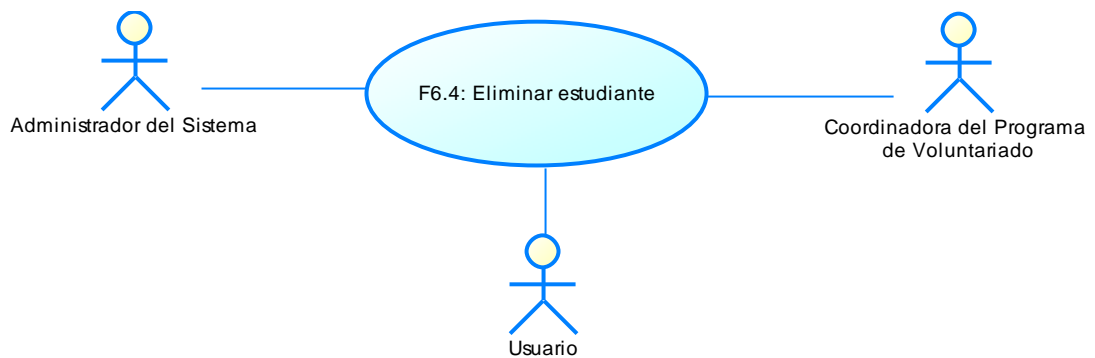


Figura 2.31: Diagrama a Detalle Eliminar estudiante

Descripción: Se podrán eliminar los estudiantes existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Estudiantes	4	Presenta pantalla de ingreso de estudiantes	
5	Ingresa cédula del estudiante que va a ser eliminado	6	Verifica si existe esa cédula	E1
		7	Carga todos los datos del estudiante	E1
		8	Aparece otro botón con la etiqueta Eliminar	
9	Pulsa el botón Eliminar	10	Presenta un mensaje para confirmar la eliminación del estudiante	
11	Pulsa Sí en el mensaje	12	Presenta un mensaje indicando que el estudiante se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
6	F6.1 o F6.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F7: Administración de profesiones

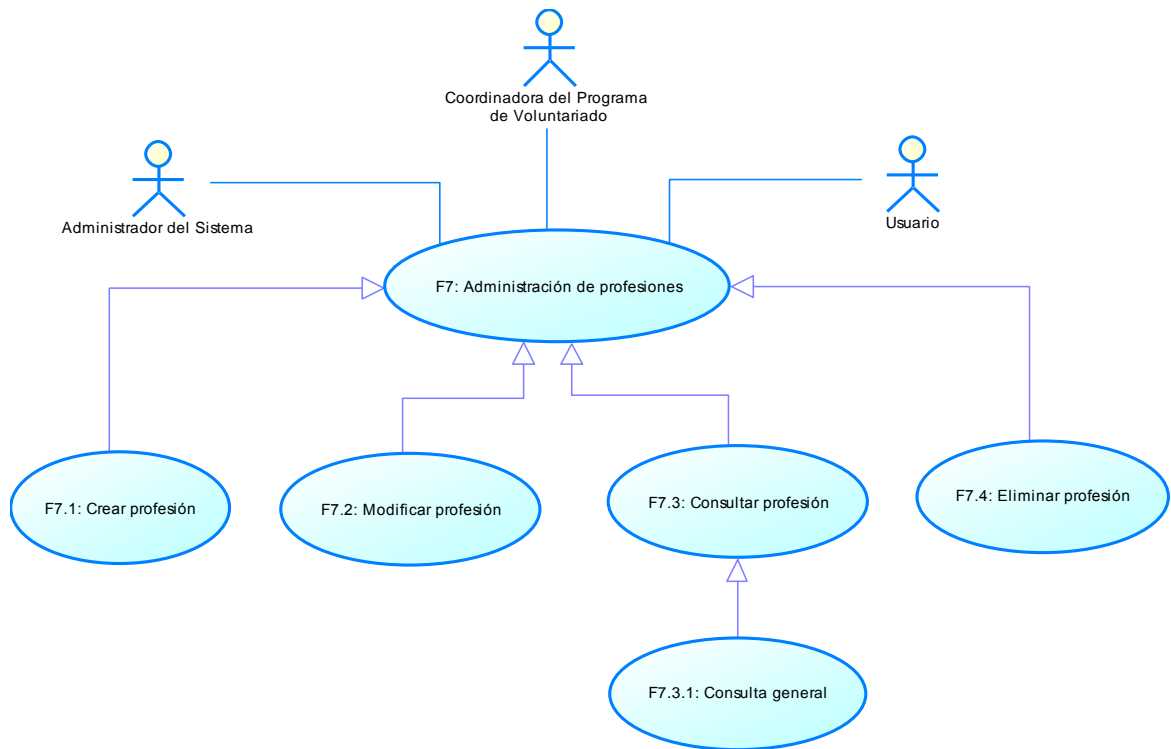


Figura 2.32: Diagrama de Casos de Uso Administración de profesiones

DETALLE:

F7.1: Crear profesión

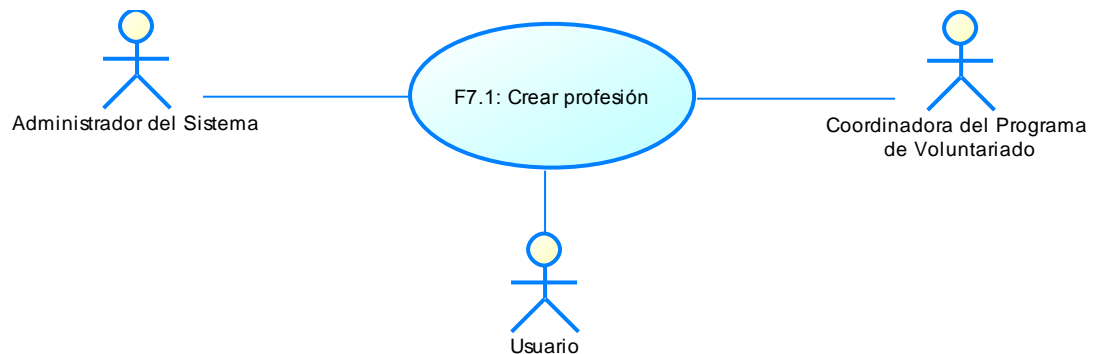


Figura 2.33: Diagrama a Detalle Crear profesión

Descripción: Se podrán crear las profesiones que posean los voluntarios del Programa de Voluntariado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Profesionales	4	Presenta pantalla de ingreso de profesiones	
		5	Carga todos las profesiones existentes en el sistema	E2
		6	Desbloquea los demás campos	
		7	Desbloquea el botón de Guardar	
8	Ingresa el nombre de la profesión			
9	Pulsa el botón Guardar	10	Almacena los datos ingresados	E1, E2
		11	Presenta un mensaje indicando que la profesión se ha creado exitosamente	

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F7.2: Modificar profesión

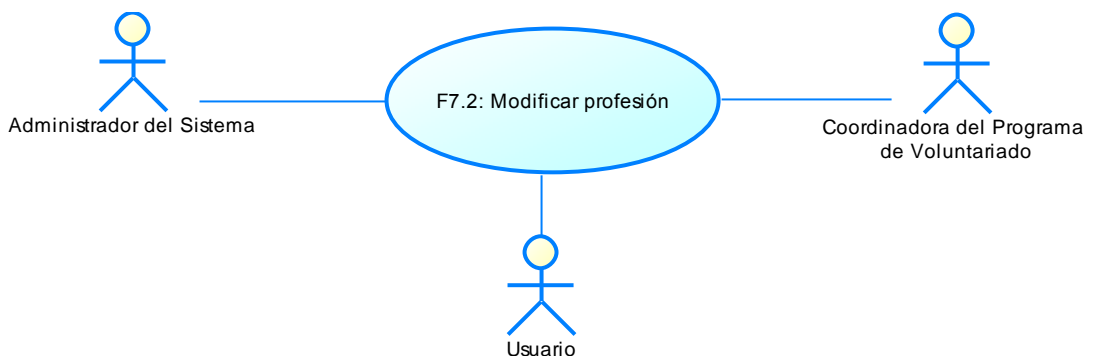


Figura 2.34: Diagrama a Detalle Modificar profesión

Descripción: Se podrán modificar el nombre de las profesiones existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Profesionales	4	Presenta pantalla de ingreso de profesiones	
		5	Carga todas las profesiones existentes en el sistema	E2
6	Selecciona la profesión que va a ser modificada	7	Carga el nombre de la profesión	E2
		8	Asigna la etiqueta Modificar en el botón	
9	Modifica el nombre de la profesión que se seleccionó			
10	Pulsa el botón Modificar	11	Almacena los datos ingresados	E1, E2
		12	Presenta un mensaje indicando que la profesión se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
5	F7.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F7.3: Consultar profesión

F7.3.1: Consulta general

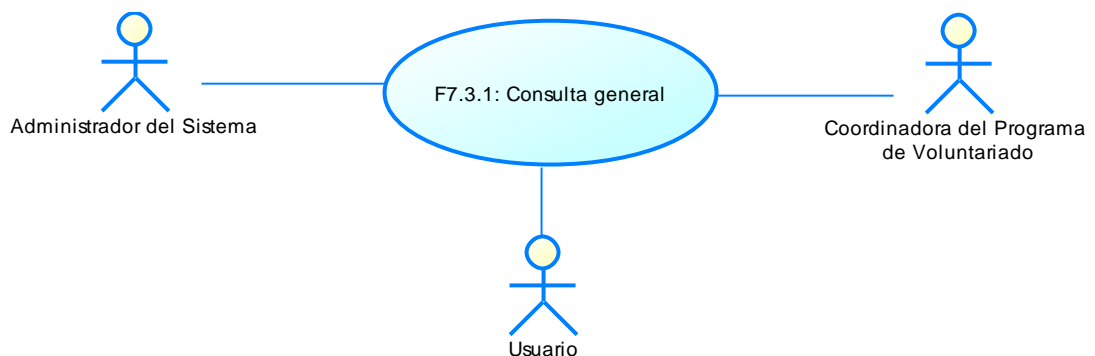


Figura 2.35: Diagrama a Detalle Consultar general profesión

Descripción: Se podrán consultar todas las profesiones existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Profesiones	6	Carga todos las profesiones que existen en el sistema	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F7.4: Eliminar profesión

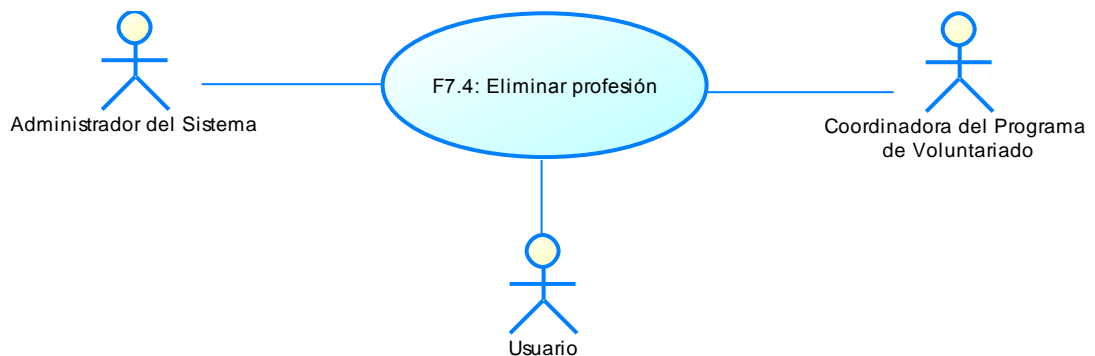


Figura 2.36: Diagrama a Detalle Eliminar profesión

Descripción: Se podrán eliminar las profesiones existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Profesiones	4	Presenta pantalla de ingreso de profesiones	
		5	Carga todas las profesiones existentes en el sistema	E1

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

6	Selecciona la profesión que va a ser eliminada	7	Carga el nombre de la profesión	E1
		8	Aparece otro botón con la etiqueta Eliminar	
9	Pulsa el botón Eliminar	10	Presenta un mensaje para confirmar la eliminación de la profesión	
11	Pulsa Sí en el mensaje	12	Presenta un mensaje indicando que la profesión se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
5	F7.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F8: Administración de universidades

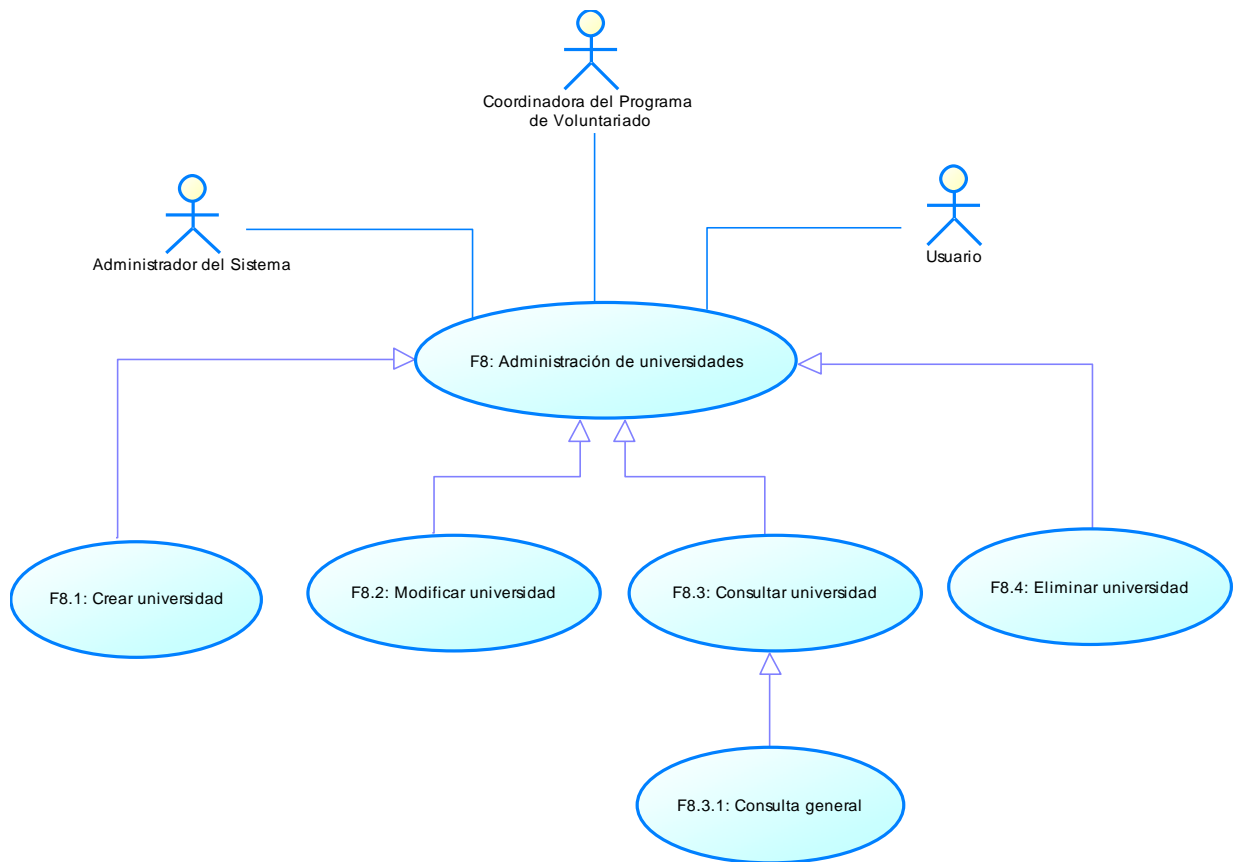


Figura 2.37: Diagrama de Casos de Uso Administración de universidades

DETALLE:

F8.1: Crear universidad

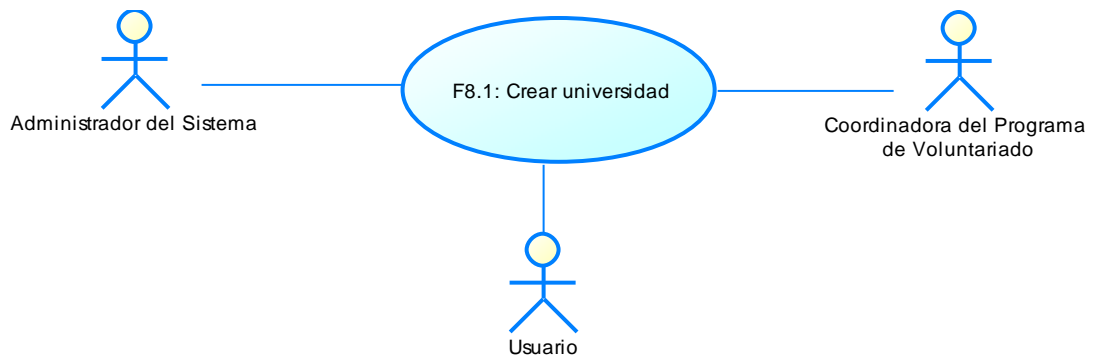


Figura 2.38: Diagrama a Detalle Crear universidad

Descripción: Se podrán crear las diferentes universidades de los estudiantes del Programa de Voluntariado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Universidades	4	Presenta pantalla de ingreso de universidades	
		5	Carga todas las universidades existentes en el sistema	E2
6	Ingresa el nombre de la universidad que va a ser creada	7	Desbloquea el botón de Guardar	E2
8	Pulsa el botón Guardar	9	Almacena los datos ingresados	E1, E2
		10	Presenta un mensaje indicando que la universidad se ha creado exitosamente	

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F8.2: Modificar universidad

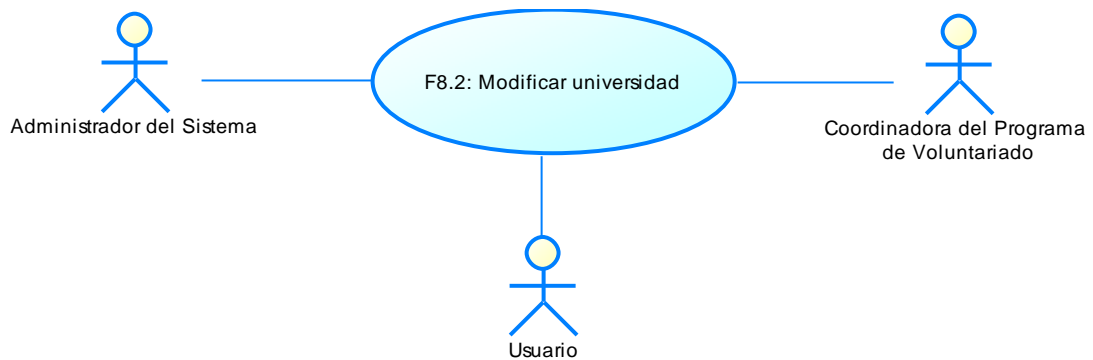


Figura 2.39: Diagrama a Detalle Modificar universidad

Descripción: Se podrán modificar el nombre de las universidades existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Universidades	4	Presenta pantalla de ingreso de universidades	
		5	Carga todas las universidades existentes en el sistema	E2
6	Selecciona la universidad que va a ser modificada	7	Carga el nombre de la universidad	E2
		8	Asigna la etiqueta Modificar en el botón	
9	Modifica el nombre de la universidad que se seleccionó			
10	Pulsa el botón Modificar	11	Almacena los datos ingresados	E1, E2
		12	Presenta un mensaje indicando que la universidad se ha modificado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
5	F8.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F8.3: Consultar universidad

F8.3.1: Consulta general

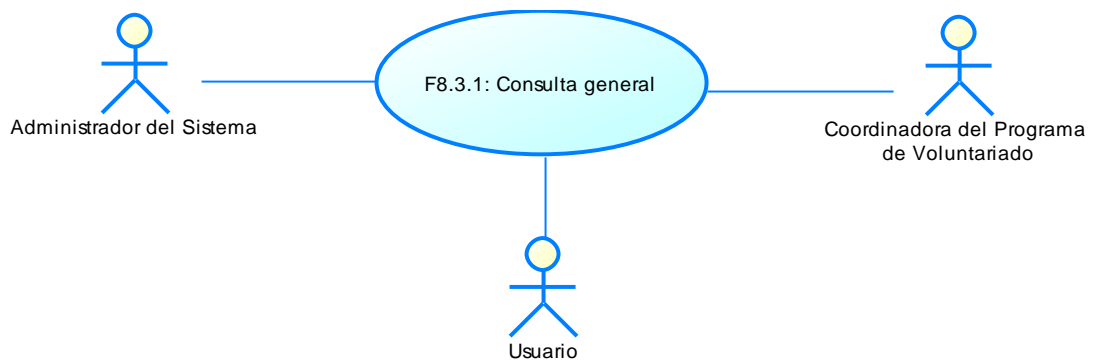


Figura 2.40: Diagrama a Detalle Consulta general universidad

Descripción: Se podrán consultar todas las universidades existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Universidades/Facultades	6	Carga todos las universidades que existen en el sistema	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F8.4: Eliminar universidad

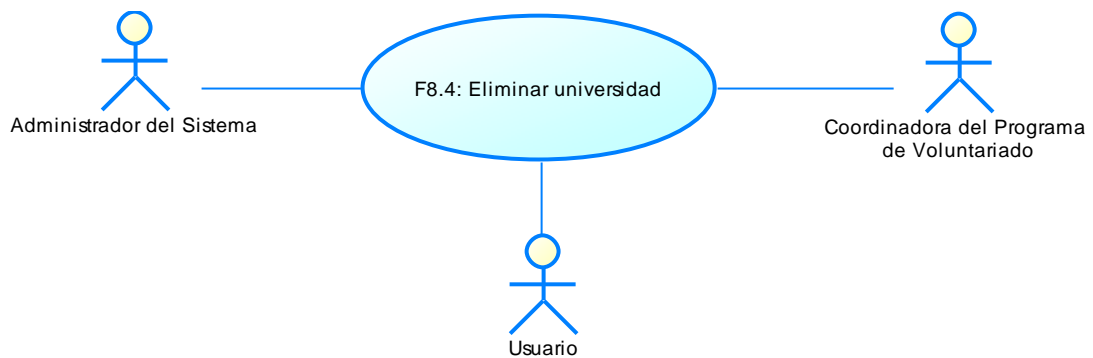


Figura 2.41: Diagrama a Detalle Eliminar universidad

Descripción: Se podrán eliminar las universidades existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Universidades	4	Presenta pantalla de ingreso de universidades	
		5	Carga todas las universidades existentes en el sistema	E1
6	Selecciona la universidad que va a ser eliminada	7	Carga el nombre de la universidad	E1
		8	Aparece otro botón con la etiqueta Eliminar	
9	Pulsa el botón Eliminar	10	Presenta un mensaje para confirmar la eliminación de la universidad	
11	Pulsa Sí en el mensaje	12	Presenta un mensaje indicando que la universidad se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
5	F8.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F9: Administración de facultades

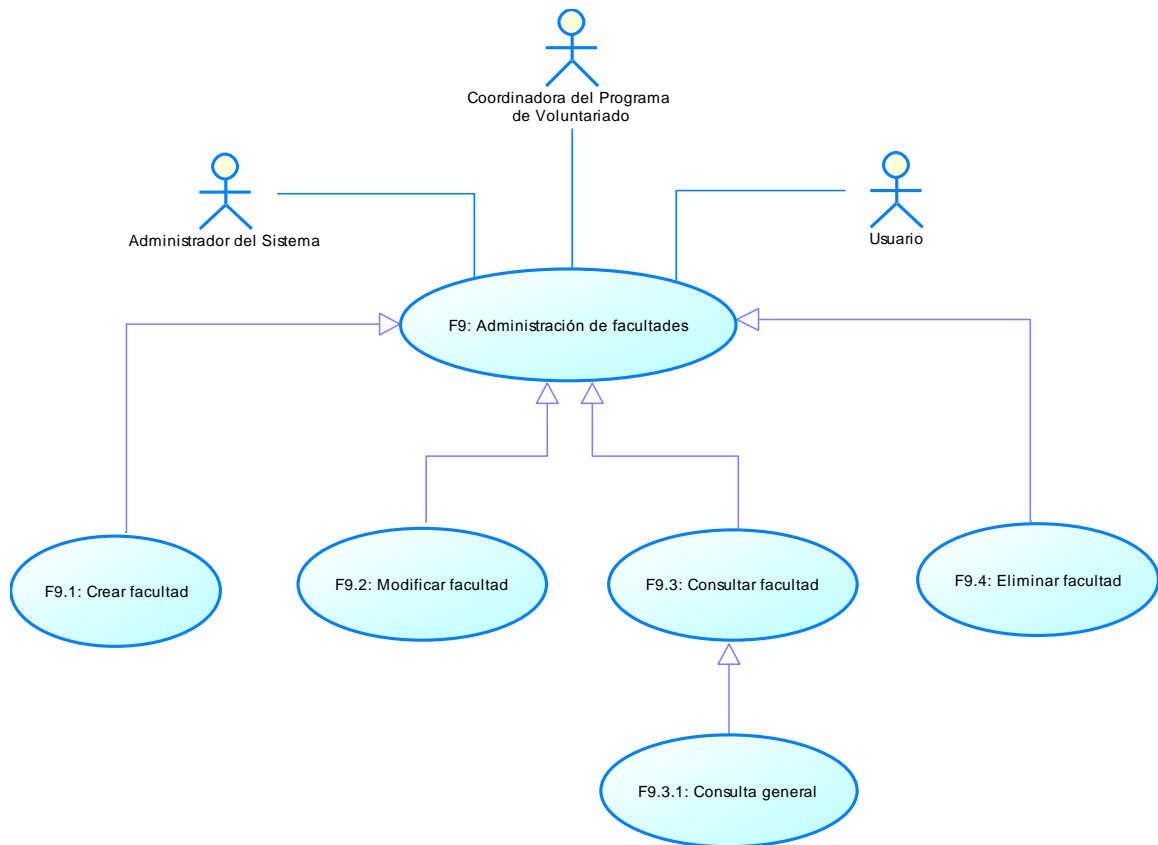


Figura 2.42: Diagrama de Casos de Uso Administración de facultades

DETALLE:

F9.1: Crear facultad

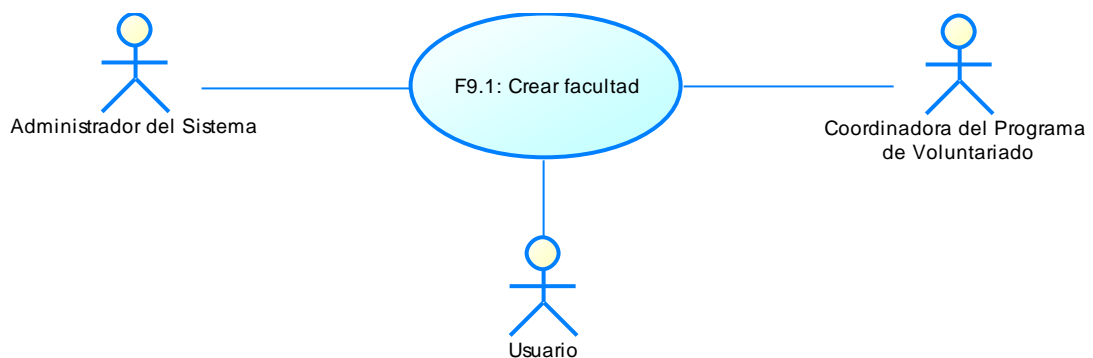


Figura 2.43: Diagrama a Detalle Crear facultad

Descripción: Se podrán crear las diferentes facultades pertenecientes a las universidades de los estudiantes del Programa de Voluntariado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Facultades	4	Presenta pantalla de ingreso de facultades	
		5	Carga todas las universidades que existen en el sistema	E2
6	Selecciona la universidad a la cual quiere agregar las facultades	7	Carga las facultades asociadas a la universidad seleccionada	E2
8	Ingresa el nombre de la facultad que va a ser creada	9	Verifica si no existe ese nombre	E2
		10	Desbloquea el botón de Guardar	
11	Pulsa el botón Guardar	12	Almacena los datos ingresados	E1, E2
		13	Presenta un mensaje indicando que la facultad se ha creado exitosamente	

Flujo Alternativo:

Paso	Caso de uso
7	F9.2 o F9.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F9.2: Modificar facultad

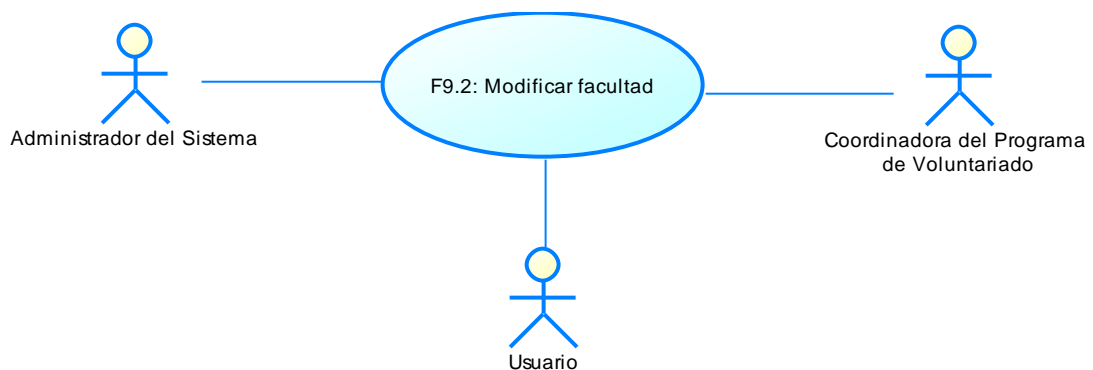


Figura 2.44: Diagrama a Detalle Modificar facultad

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

Descripción: Se podrán modificar el nombre de las facultades existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Facultades	4	Presenta pantalla de ingreso de facultades	
		5	Carga todas las universidades que existen en el sistema	E2
6	Selecciona la universidad de la cual quiere modificar las facultades	7	Carga las facultades asociadas a la universidad seleccionada	E2
8	Selecciona la facultad que va a ser modificada	9	Carga el nombre de la facultad seleccionada	E2
		10	Asigna la etiqueta Modificar en el botón	
11	Modifica el nombre de la facultad que seleccionó			
12	Pulsa el botón Modificar	13	Almacena los datos ingresados	E1, E2
		14	Presenta un mensaje indicando que la facultad se ha modificado exitosamente	

Flujo Alterno:

Paso	Caso de uso
7	F9.1 o F9.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F9.3: Consultar facultad

F9.3.1: Consulta general

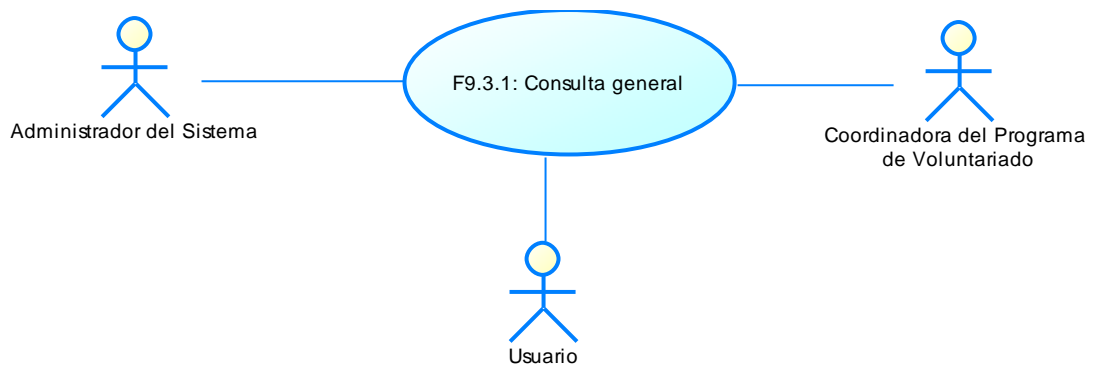


Figura 2.45: Diagrama a Detalle Consulta general facultad

Descripción: Se podrán consultar todas las facultades existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Universidades/Facultades	6	Carga todas las universidades que existen en el sistema	E1
7	Selecciona la universidad de la cual quiere consultar las facultades	8	Carga las facultades asociadas a la universidad seleccionada	E1

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F9.4: Eliminar facultad

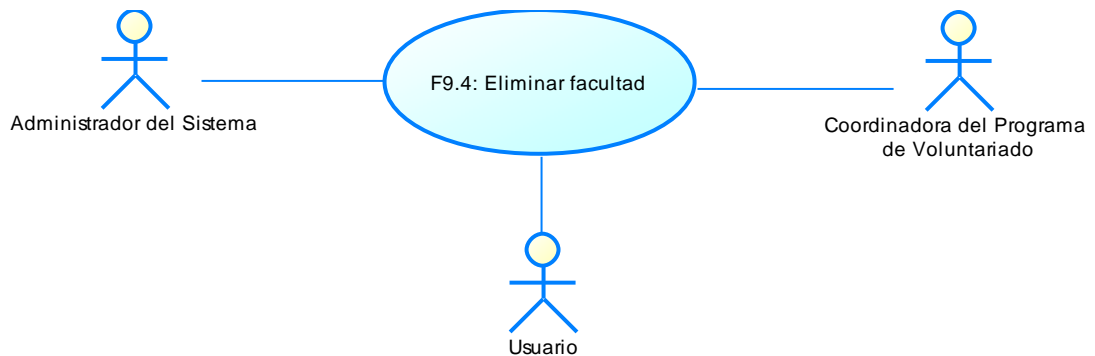


Figura 2.46: Diagrama a Detalle Eliminar facultad

Descripción: Se podrán eliminar las facultades existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Facultades	4	Presenta pantalla de ingreso de facultades	
		5	Carga todas las universidades existentes en el sistema	E1
6	Selecciona la universidad de la cual quiere eliminar las facultades	7	Carga las facultades asociadas a la universidad seleccionada	E1
8	Selecciona la facultad que va a ser eliminada	9	Carga el nombre de la facultad	E1
		10	Aparece otro botón con la etiqueta Eliminar	
11	Pulsa el botón Eliminar	12	Presenta un mensaje para confirmar la eliminación de la facultad	
13	Pulsa Sí en el mensaje	14	Presenta un mensaje indicando que la facultad se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
7	F9.1 o F9.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F10: Administración de registros de asistencia

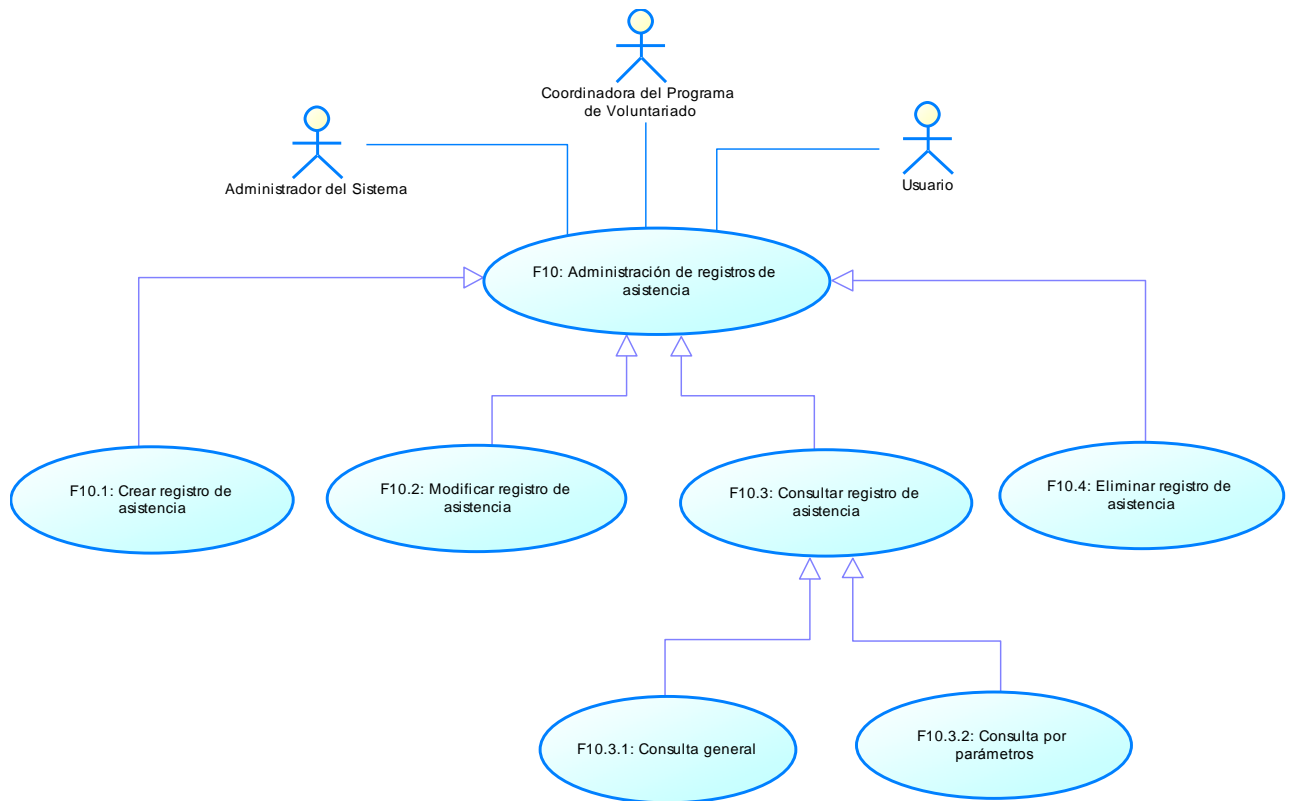


Figura 2.47: Diagrama de Casos de Uso Administración de registros de asistencia

DETALLE:

F10.1: Crear registro de asistencia

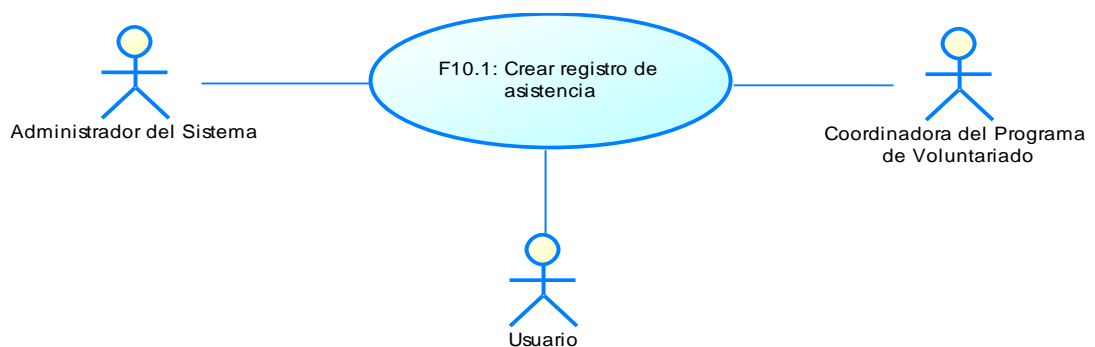


Figura 2.48: Diagrama a Detalle Crear registro de asistencia

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

Descripción: Se podrán crear los registros de asistencia de los voluntarios y estudiantes que participen en el Programa de Voluntariado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Registros de asistencia	4	Presenta pantalla de registros de asistencia	
		5	Carga todos los proyectos que existen en el sistema	E3
6	Selecciona el proyecto del cual quiere seleccionar los subproyectos	7	Carga los subproyectos asociados al proyecto seleccionado	E3
8	Selecciona el subproyecto que quiere consultar	9	Carga los voluntarios y estudiantes que están en el subproyecto seleccionado	E3
10	Selecciona el participante al cual se quiere crear un registro de asistencia	11	Carga los registros de asistencias del participante seleccionado	E3
12	Ingresa los datos del registro de asistencia	13	Verifica si no existe ese registro	
		14	Desbloquea el botón de Guardar	
15	Pulsa el botón Guardar	16	Almacena los datos ingresados	E1, E2, E3
		17	Presenta un mensaje indicando que el registro de asistencia se ha creado exitosamente	

Flujo Alterno:

Paso	Caso de uso
11	F10.2 o F10.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F10.2: Modificar registro de asistencia

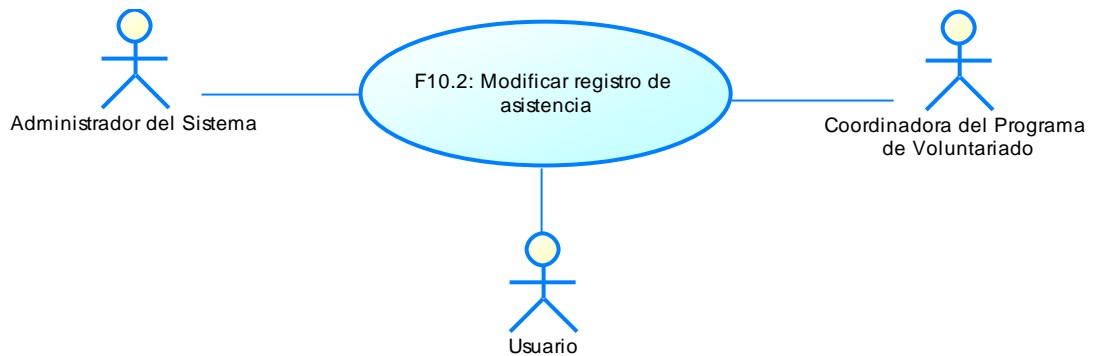


Figura 2.49: Diagrama a Detalle Modificar registro de asistencia

Descripción: Se podrán modificar ciertos campos de la información o toda la información de los registros de asistencia existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Registros de asistencia	4	Presenta pantalla de registros de asistencia	
		5	Carga todos los proyectos que existen en el sistema	E3
6	Selecciona el proyecto del cual quiere seleccionar los subproyectos	7	Carga los subproyectos asociados al proyecto seleccionado	E3
8	Selecciona el subproyecto que quiere consultar	9	Carga los voluntarios y estudiantes que están en el subproyecto seleccionado	E3
10	Selecciona el participante del cual se quiere modificar un registro de asistencia	11	Carga los registros de asistencias del participante seleccionado	E3
12	Selecciona el registro de asistencia que va a ser modificado	13	Carga todos los datos del registro de asistencia seleccionado	E3
		14	Asigna la etiqueta Modificar en el botón	
15	Modifica los datos del registro de asistencia que se desea cambiar			
16	Pulsa el botón Modificar	17	Almacena los datos ingresados	E1, E2, E3
		18	Presenta un mensaje indicando que el registro de asistencia se ha modificado	

			exitosamente	
--	--	--	--------------	--

Flujo Alternativo:

Paso	Caso de uso
11	F10.1 o F10.4

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Datos ingresados incorrectos	Corregir los datos incorrectos y volver a intentar
E3	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F10.3: Consultar registro de asistencia

F10.3.1: Consulta general

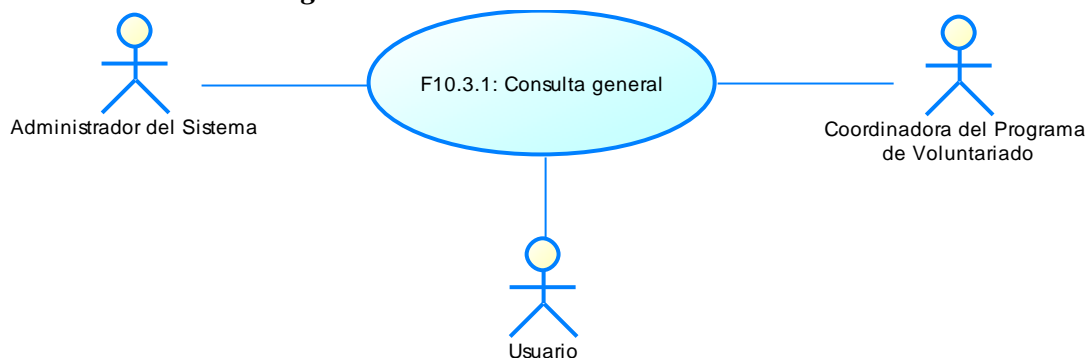


Figura 2.50: Diagrama a Detalle Consulta general registro de asistencia

Descripción: Se podrán consultar todos los registros de asistencia existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona General	4	Presenta pantalla de consulta general	
5	Selecciona la pestaña de Registros de asistencia	6	Carga todos los proyectos que existen en el sistema	E1
7	Selecciona el proyecto del cual quiere seleccionar los subproyectos	8	Carga los subproyectos asociados al proyecto seleccionado	E1
9	Selecciona el subproyecto que quiere consultar			

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

10	Selecciona Voluntarios o Estudiantes	11	Carga los registros de asistencias de Voluntarios o Estudiantes	E1
----	--------------------------------------	----	---	----

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F10.3.2: Consulta por parámetros

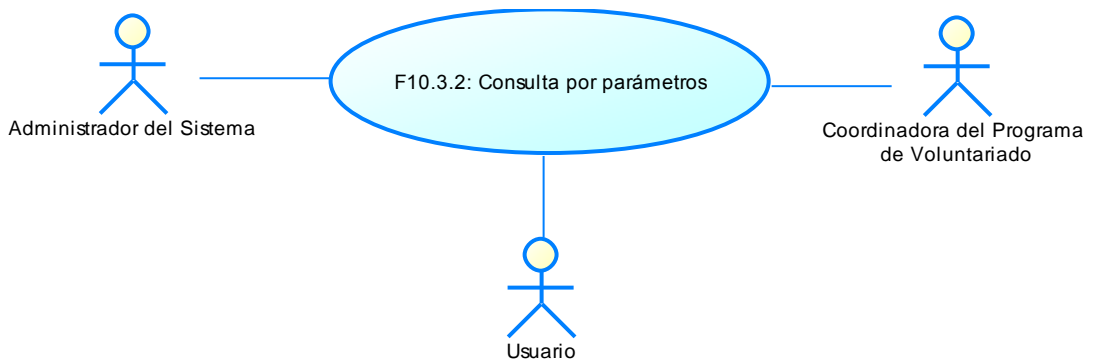


Figura 2.51: Diagrama a Detalle Consulta por parámetros registro de asistencia

Descripción: Se podrán consultar los registros de asistencia que existen en el sistema por medio de parámetros que se ingresen.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Consultas del menú principal	2	Presenta lista con los tipos de consultas	
3	Selecciona Parámetros	4	Presenta pantalla de consulta por parámetros	
5	Selecciona la pestaña de Registros de asistencia	6	Carga todos los proyectos que existen en el sistema	E1
7	Selecciona el proyecto del cual quiere seleccionar los subproyectos	8	Carga los subproyectos asociados al proyecto seleccionado	E1
9	Selecciona el subproyecto que quiere consultar	10	Carga los participantes que están en el subproyecto seleccionado	E1
11	Selecciona el participante del cual se quiere consultar los registros de asistencia			

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

12	Ingresar los parámetros de búsqueda			
13	Pulsar el botón Consultar	14	Carga los registros de asistencia que contengan los parámetros ingresados	E1, E2

Excepciones:

Código	Mensaje	Alternativa
E1	Parámetros vacíos	Llenar al menos un parámetro
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F10.4: Eliminar registro de asistencia

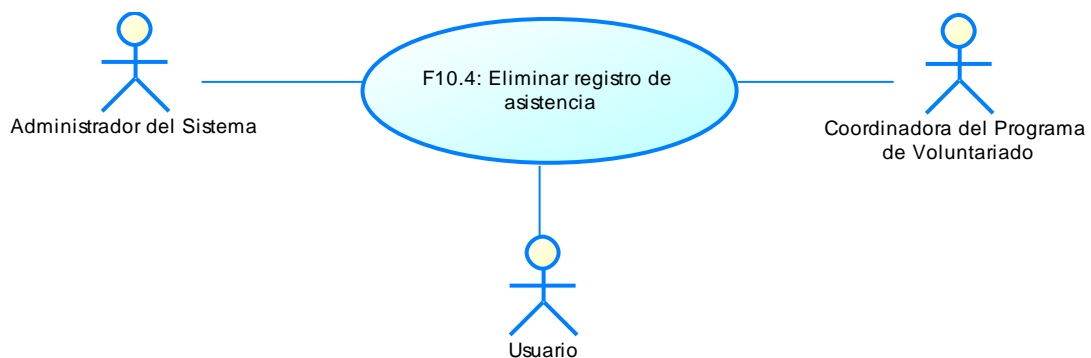


Figura 2.52: Diagrama a Detalle Eliminar registro de asistencia

Descripción: Se podrán eliminar los registros de asistencia existentes en el sistema.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Administración del menú principal	2	Presenta lista con todas las administraciones	
3	Selecciona Registros de asistencia	4	Presenta pantalla de registros de asistencia	
		5	Carga todos los proyectos que existen en el sistema	E1
6	Selecciona el proyecto del cual quiere seleccionar los subproyectos	7	Carga los subproyectos asociados al proyecto seleccionado	E1
8	Selecciona el subproyecto que quiere consultar	9	Carga los participantes que están en el subproyecto seleccionado	E1
10	Selecciona el participante del cual se quiere eliminar	11	Carga los registros de asistencias del participante	E1

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	un registro de asistencia		seleccionado	
12	Selecciona el registro de asistencia que va a ser eliminado	13	Aparece otro botón con la etiqueta Eliminar	
14	Pulsa el botón Eliminar	15	Presenta un mensaje para confirmar la eliminación del registro de asistencia	
16	Pulsa Sí en el mensaje	17	Presenta un mensaje indicando que el registro de asistencia se ha eliminado exitosamente	E1

Flujo Alternativo:

Paso	Caso de uso
11	F10.1 o F10.2

Excepciones:

Código	Mensaje	Alternativa
E1	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F11: Generación de reportes

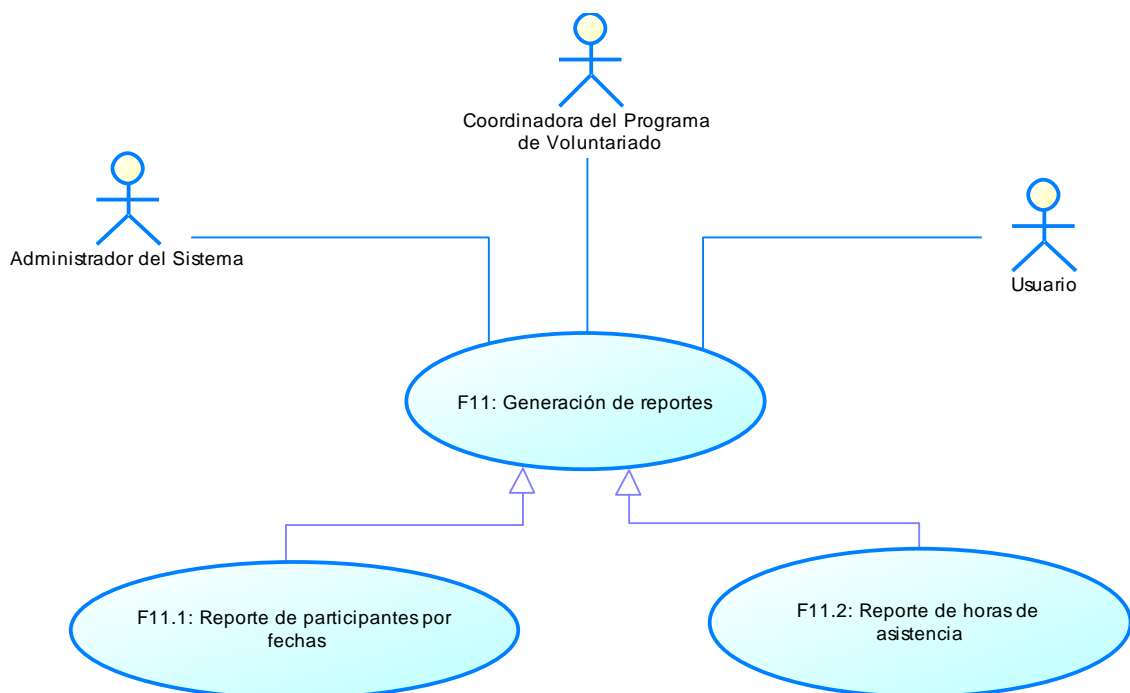


Figura 2.53: Diagrama de Casos de Uso Generación de reportes

DETALLE:

F11.1: Reporte de participantes por fechas



Figura 2.54: Diagrama a Detalle Reporte de participantes por fechas

Descripción: Se podrán crear reportes de los voluntarios y estudiantes dependiendo de en qué proyectos y subproyectos estén y en qué fechas hayan ingresado.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Reportes del menú principal	2	Presenta lista con todos los reportes	
3	Selecciona Reportes de participantes por fechas	4	Presenta pantalla de creación de reportes	
		5	Carga los proyectos y subproyectos existentes en el sistema	E2
6	Selecciona el proyecto o subproyecto del cual quiere consultar los participantes	7	Desbloquea los demás campos	
8	Selecciona el tipo de fecha que quiere consultar			
9	Selecciona el rango de fechas para crear el reporte			
10	Pulsa el botón Consultar	11	Carga los voluntarios y estudiantes que contengan los parámetros ingresados para el reporte	E2
		12	Desbloquea el botón de Generar Reporte	
13	Pulsa el botón Generar Reporte	14	Aparece un cuadro de diálogo para seleccionar la ruta donde guardar el reporte	E1, E2
15	Pulsa Guardar	16	Abre automáticamente el reporte en formato PDF	

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

F11.2: Reporte de horas de asistencia

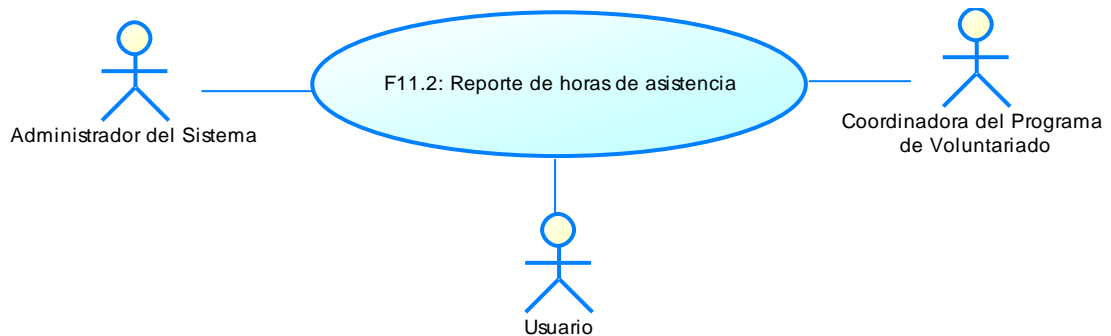


Figura 2.55: Diagrama a Detalle Reporte de horas de asistencia

Descripción: Se podrán crear reportes de las horas de asistencia de los voluntarios y estudiantes en los diferentes proyectos en los que estén envueltos.

Actores: Administrador del Sistema, Coordinadora del Programa de Voluntariado y Usuario

Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona opción Reportes del menú principal	2	Presenta lista con todos los reportes	
3	Selecciona Reportes de horas de asistencia	4	Presenta pantalla de creación de reportes	
5		6	Carga los proyectos y subproyectos existentes en el sistema	E2
7	Selecciona el proyecto o subproyecto del cual quiere consultar los participantes	8	Desbloquea los demás campos	
		9	Carga los voluntarios y estudiantes asociados al proyecto o subproyecto seleccionado	E2
10	Selecciona el voluntario o estudiante del cual quiere crear el reporte de horas de asistencia			

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL
PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

11	Selecciona el rango de fechas para crear el reporte			
12	Pulsa el botón Consultar	13	Carga los registros de asistencia según el rango de fechas seleccionado	E2
		14	Calcula el total de horas de los registros de asistencia cargados	E2
		15	Desbloquea el botón de Generar Reporte	
16	Pulsa el botón Generar Reporte	17	Aparece un cuadro de diálogo para seleccionar la ruta donde guardar el reporte	E1, E2
18	Pulsa Guardar	19	Abre automáticamente el reporte en formato PDF	

Excepciones:

Código	Mensaje	Alternativa
E1	Datos obligatorios vacíos	Llenar los campos restantes
E2	Error en la base de datos	Cerrar la aplicación y volverla a iniciar

2.2.2.3. Diagrama de clases a nivel conceptual

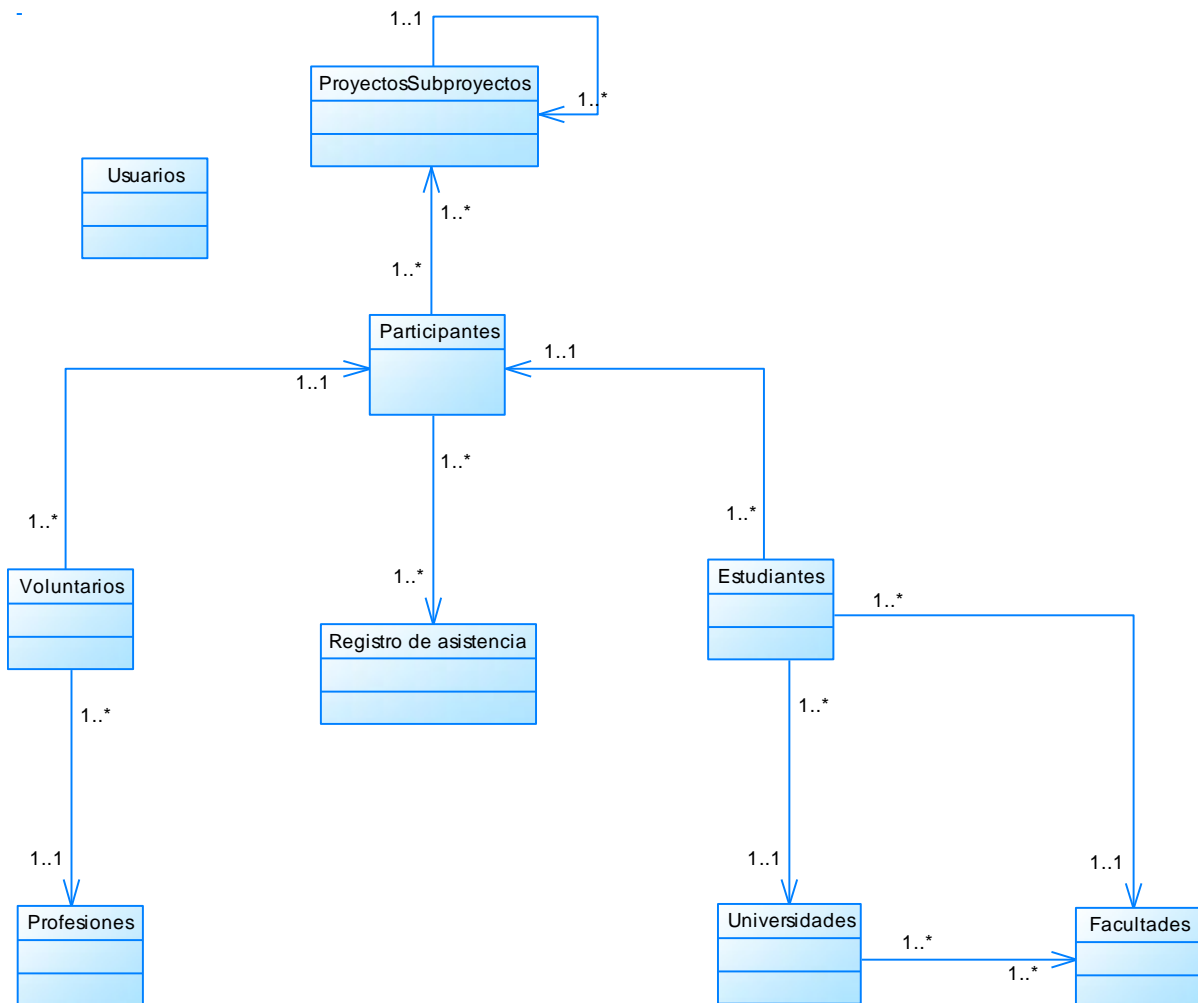


Figura 2.56: Diagrama de Clases a Nivel Conceptual

2.2.3. Requerimientos no funcionales

2.2.3.1. Restricciones de diseño e implementación

- Se utilizará el enfoque metodológico del Modelo en Cascada.
- Se utilizará el paradigma de Programación Orientada a Objetos.
- Se utilizará la arquitectura de programación en Tres Capas.
- Como lenguaje de programación se utilizará Java 7.
- Para el desarrollo del producto se utilizará el IDE⁸ NetBeans 8.0.2.
- Como editor UML⁹ se utilizará PowerDesigner versión 15.3.
- Se utilizará como Motor de Base de Datos a MySQL con sus versiones para el conector ODBC 5.3.4. y el MySQL Workbench 6.2.4.

⁸ IDE (Integrated Development Environment) es una aplicación de software...

⁹ UML (Unified Modeling Language) es el lenguaje de modelado de sistemas...

2.2.3.2. *Requerimientos especiales*



Figura 2.57: Cuadro de Requerimientos Especiales

Fuente: <http://ingenieralbg.blogspot.com/2013/05/requerimientos-funcionales-y-no.html>

Requerimientos del producto

- **Usabilidad**

En el sistema se puede apreciar que todas sus pantallas están bien descritas con palabras de uso común para los usuarios finales.

Cada formulario está apropiadamente controlado para que en los campos donde deben ingresar solo números no permita ingresar letras y viceversa. Además cuenta con indicaciones de cómo se debe ingresar los datos en ciertos campos, y en algunas opciones se facilita el ingreso con listas desplegables.

Los botones están claramente etiquetados según la función que desempeñan, y algunas pantallas son independientes de la pantalla principal, para que su administración sea más amigable.

Los mensajes de errores están bien definidos e indican que acción se debe realizar para solucionarlo.

Todos estos elementos han hecho que la interfaz sea más amigable y de fácil manejo para el usuario.

- **Requerimientos de Eficiencia**

- **Rendimiento**

El sistema al ser una aplicación de escritorio, aprovecha de mejor forma los recursos de hardware y software del equipo.

Para poder definir un funcionamiento adecuado de la aplicación se tomó como prueba, realizar 50 veces cada proceso de todos los módulos; es decir, hacer login¹⁰ con un usuario, una carga de datos existentes en la base, una creación de datos nuevos, una modificación de esos datos, una consulta de datos generales y con parámetros, y una eliminación de datos creados.

Los tiempos de respuestas que se definieron para los procesos del sistema son los siguientes:

Tiempo de respuesta muy alto < = 1 segundo

Tiempo de respuesta aceptable de 1 hasta 5 segundos

Tiempo de respuesta muy bajo > = 5 segundos

Los tiempos que se obtuvieron en la prueba son los siguientes:

PROCESO	TIEMPO DE RESPUESTA	OBSERVACIÓN
Login	1 seg.	Tiempo de respuesta aceptable para el usuario
Carga de datos	2 seg.	Tiempo de respuesta aceptable para el usuario
Creación de datos nuevos	1 seg.	Tiempo de respuesta aceptable para el usuario
Modificación de datos	1 seg.	Tiempo de respuesta aceptable para el usuario
Consulta de datos generales y parámetros	2 seg.	Tiempo de respuesta aceptable para el usuario
Eliminación de datos	1 seg.	Tiempo de respuesta aceptable para el usuario

La distribución del volumen de trabajo que cada proceso tuvo con la base de datos en la prueba realizada, es la siguiente:

- _ 5% Login
- _ 25% Carga de datos
- _ 20% Creación de datos nuevos
- _ 15% Modificación de datos
- _ 25% Consulta de datos generales y parámetros
- _ 10% Eliminación de datos

¹⁰ Login es el proceso mediante el cual...

Como la aplicación es instalada en cada máquina, solo un usuario a la vez puede hacer login. Por lo tanto, con 50 transacciones que se realizaron en la prueba anterior, no existió ninguna falla en el funcionamiento y se puede garantizar que fue adecuado.

➤ **Espacio**

Los requerimientos de espacio necesario en el disco duro de las máquinas para que el sistema completo funcione correctamente, depende de los programas que se instale y de los datos que serán almacenados.

Los programas que se deben instalar son:

- _ La aplicación → 10 MB
- _ Motor de base de datos MySQL con las herramientas necesarias → 320 MB

Para los datos que serán almacenados, la cantidad de información estimada que se ingresará a la base de datos en un año serán los siguientes:

- _ Los registros que más se ingresarán son los de voluntarios, estudiantes y registros de asistencia. Para un año la cantidad total proyectada de registros será de 50.000, que son unos 400 MB.

El total de espacio necesario es de 730 MB aproximadamente. Por lo tanto, las máquinas podrán satisfacer los requerimientos de espacio sin ningún problema, ya que poseen un disco duro mínimo de 80 GB.

• **Fiabilidad**

El sistema al ser una aplicación de escritorio es más robusto y estable, dependiendo principalmente de las características del equipo en el cual se está operando.

Posee pantallas con información clara y formularios muy bien descritos con métodos de validación de entrada de datos, que evitan que se ingrese mal la información.

El sistema está configurado para que cuando suceda algún error, no permita realizar la acción solicitada y se pueda volver a intentar.

Todos las advertencias y los errores que se suscitan están manejados con mensajes de diálogo, que indican al usuario si las acciones son exitosas o no.

- **Portabilidad**

Este sistema al ser una aplicación de escritorio no posee una buena portabilidad ya que debe ser instalada en cada una de las máquinas donde quieran usarla.

Requerimientos organizacionales

- **Entrega**

El sistema al ser un prototipo funcional¹¹, será presentado a la FMDJ para definir si el funcionamiento del prototipo es óptimo y puede aplicarse para su uso.

El proceso de entrega será la instalación del sistema completo en la máquina de la Coordinadora del Programa de Voluntariado, donde se procederá a hacer una capacitación del manejo de la aplicación.

Se procederá a realizar un ejemplo de todo el proceso del ingreso de un voluntario en la FMDJ, utilizando la aplicación.

Se probará todos los módulos y se generará todos los imprimibles que la aplicación puede generar.

- **Implementación**

La FMDJ definió como estándar de implementación de las pantallas de la aplicación, el uso de imágenes de fondo empleadas en su página web.

Para la implementación de los formularios, se definió los mismos campos que se usan en las plantillas de Excel, e inclusive en el mismo orden en que son ingresadas.

La principal responsable de los estándares de la aplicación es la Coordinadora del Programa de Voluntariado, quien será la persona que la utilizará con mayor frecuencia.

Ella designará quienes serán los otros usuarios responsables de utilizar la aplicación, y que perfil deberán utilizar para poder ingresar.

También podrá considerar cambios en la aplicación dependiendo de nuevas necesidades que se deban implementar.

¹¹ Prototipo funcional es un sistema que sin tener todas...

- **Estándares**

Para el desarrollo del sistema, no hubo un requerimiento específico en el lenguaje de programación que se debía utilizar, por lo que se sugirió la utilización de Java, ya que para esta aplicación se consideró que se debía usar un lenguaje orientado a objetos, que posee flexibilidad en su uso, de naturaleza Open Source¹², y que sea potente y seguro.

De igual manera, al no tener un requerimiento específico del gestor de base de datos que se debía utilizar, se recomendó la utilización de MySQL, ya que aparte de ser de naturaleza Open Source, ofrece un soporte completo para cualquier necesidad de desarrollo.

Requerimientos externos

- **Interoperabilidad**

Los estándares en cuanto a interoperabilidad fue asegurarse que cada módulo presente la misma consistencia visual en cuanto a los datos que deben ingresarse. Es decir, que se mantenga similitud en los campos de texto y los mismos botones para cada operación que se deba realizar.

También que si desde un módulo se realiza alguna modificación a algún dato, este cambio se vea reflejado en otro módulo que maneje este dato.

- **Éticos**

El desarrollo del sistema se lo realizó acoplándose a la norma principal impuesta por la FMDJ. La cual indica que solo el perfil de la Coordinadora del Programa de Voluntariado tiene acceso a administrar los proyectos y subproyectos que se manejan en la fundación. Es decir, solo la Coordinadora podrá crear, modificar o eliminar la información referente a los proyectos que se están llevando a cabo en el Programa de Voluntariado.

Para controlar esto se consideró realizar la implementación de una autenticación de usuario por perfiles, y así evitar que otros usuarios puedan alterar esta información.

- **Requerimientos Legislativos**

- **Privacidad**

El sistema solo será instalado en las máquinas de las personas que trabajen en el Programa de Voluntariado. Para lo cual, la Coordinadora será la encargada de asignar qué persona puede solicitar la instalación del sistema y

¹² Open Source (Código abierto) es la expresión con la que...

cuál máquina utilizará. Además de crear un usuario de acceso y definir el perfil de usuario que utilizará según las funciones que vaya a desempeñar.

En cuanto a la privacidad de los datos, solo la máquina de la Coordinadora tendrá la base de datos centralizada protegida por contraseña, y solo podrá acceder con la ayuda del Administrador del Sistema.

➤ **Seguridad**

El sistema cuenta con una validación de ingreso mediante clave de usuario, dando acceso a las funcionalidades según el perfil del usuario y no se permitirá el acceso de usuarios no registrados al sistema ni a la información de la base de datos.

2.2.4. Otros requerimientos

2.2.4.1. Interfaces gráficas

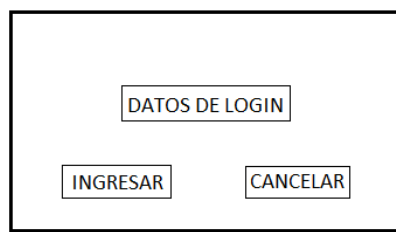


Figura 2.58: Modelo de Interfaz Gráfica de Login

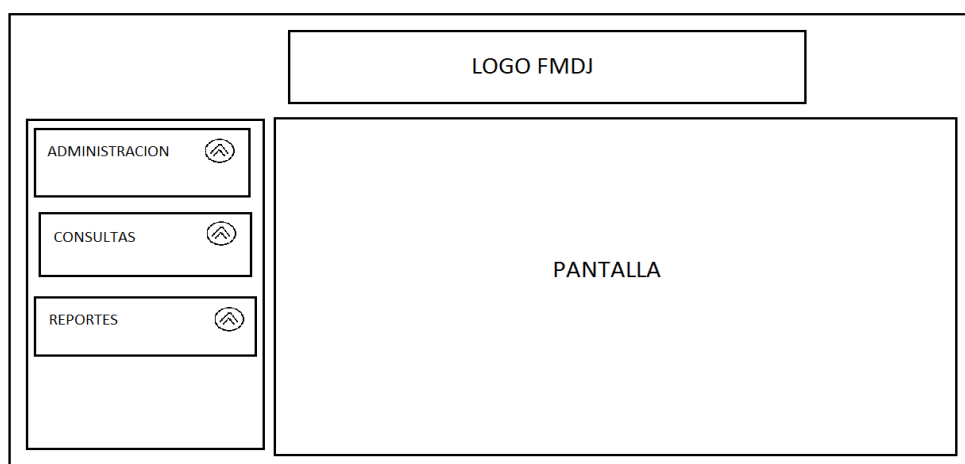


Figura 2.59: Modelo de Interfaz Gráfica de Pantalla Principal

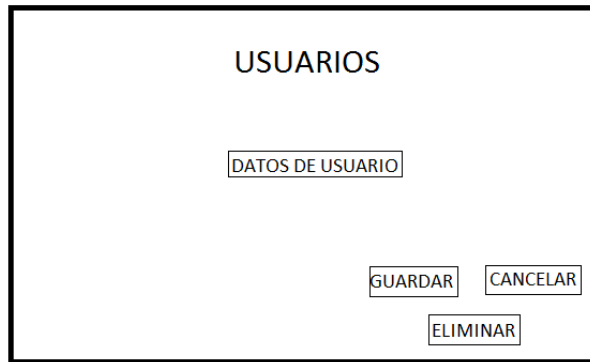


Figura 2.60: Modelo de Interfaz Gráfica de Pantalla de Usuarios

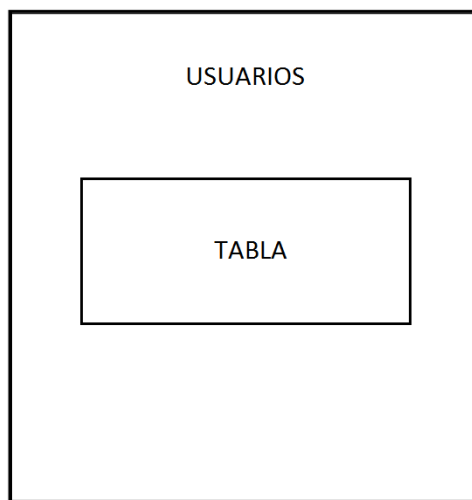


Figura 2.61: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Usuarios

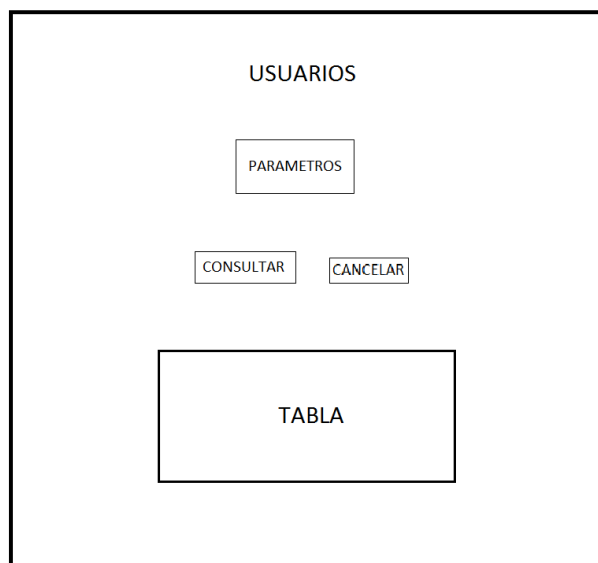


Figura 2.62: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Usuarios

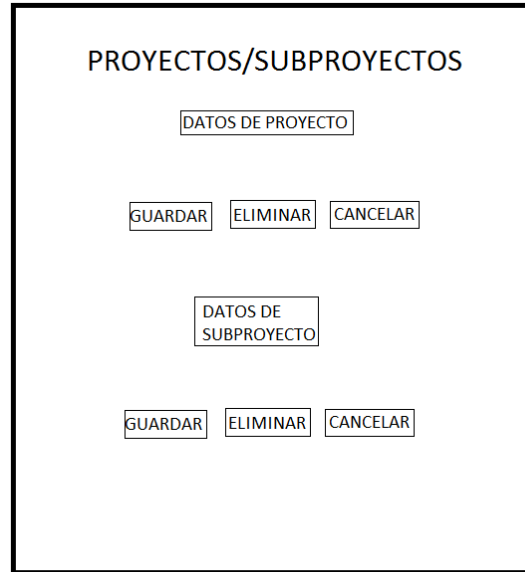


Figura 2.63: Modelo de Interfaz Gráfica de Pantalla de Proyectos/Subproyectos

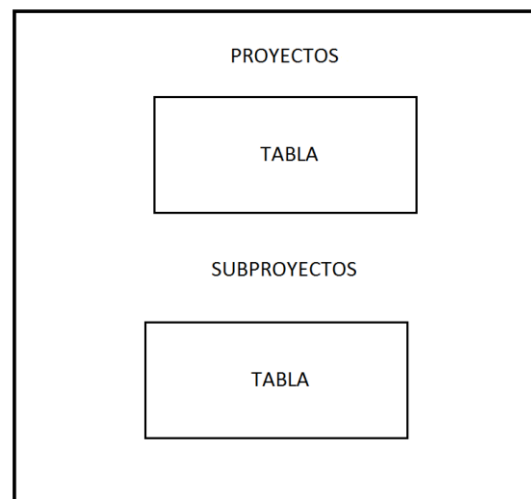


Figura 2.64: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Proyectos/Subproyectos

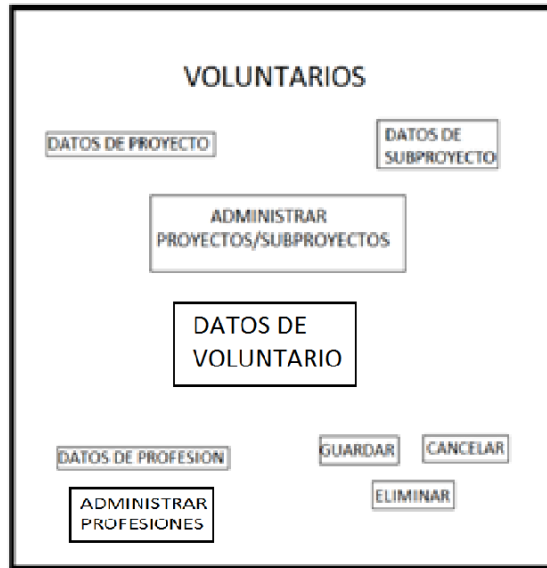


Figura 2.65: Modelo de Interfaz Gráfica de Pantalla de Voluntarios

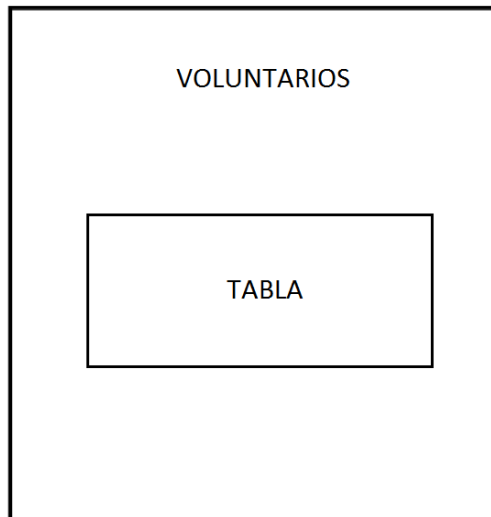


Figura 2.66: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Voluntarios

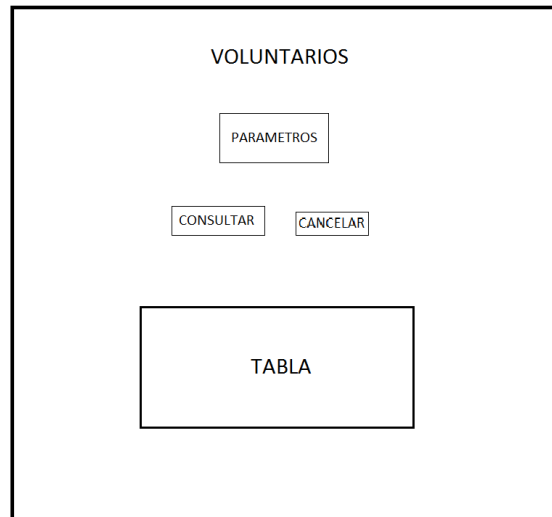


Figura 2.67: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Voluntarios

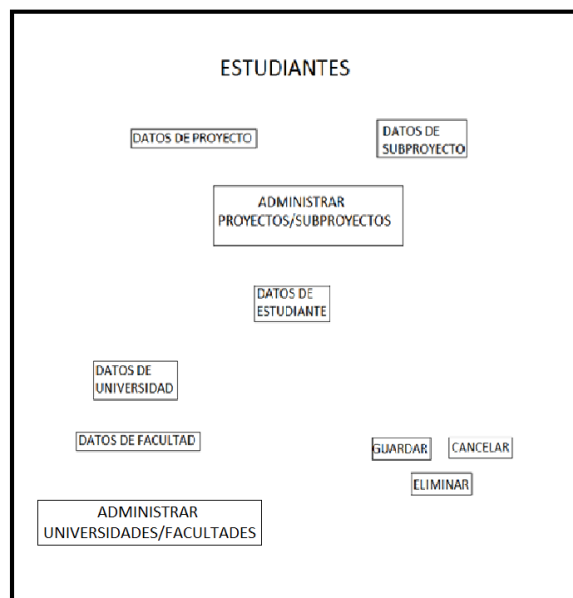


Figura 2.68: Modelo de Interfaz Gráfica de Pantalla de Estudiantes

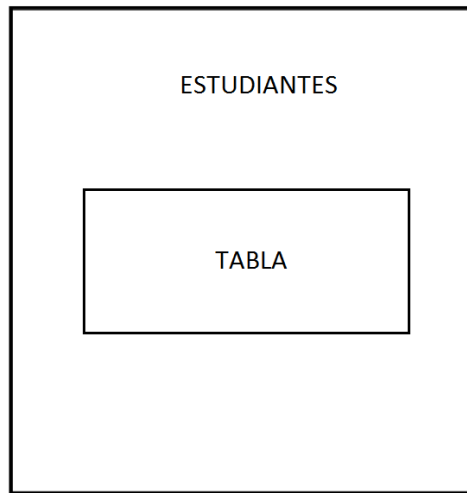


Figura 2.69: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Estudiantes

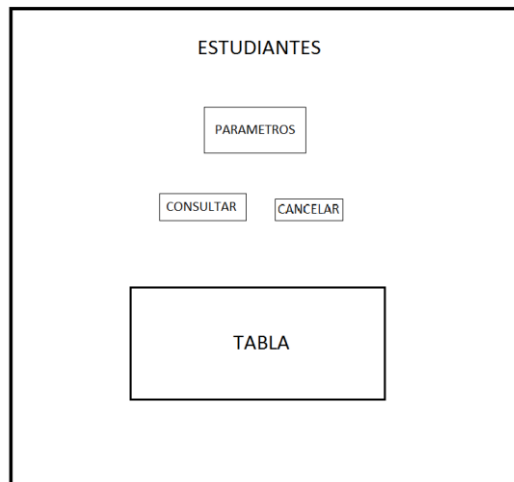


Figura 2.70: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Estudiantes

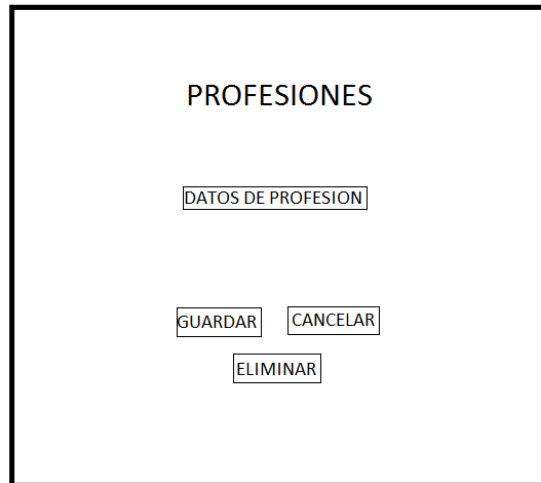


Figura 2.71: Modelo de Interfaz Gráfica de Pantalla de Profesiones

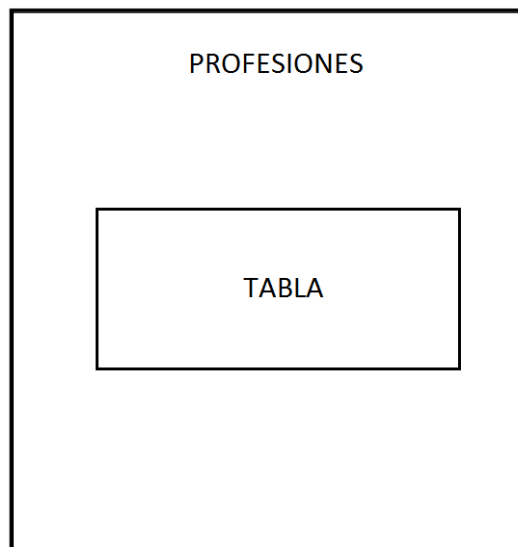


Figura 2.72: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Profesiones

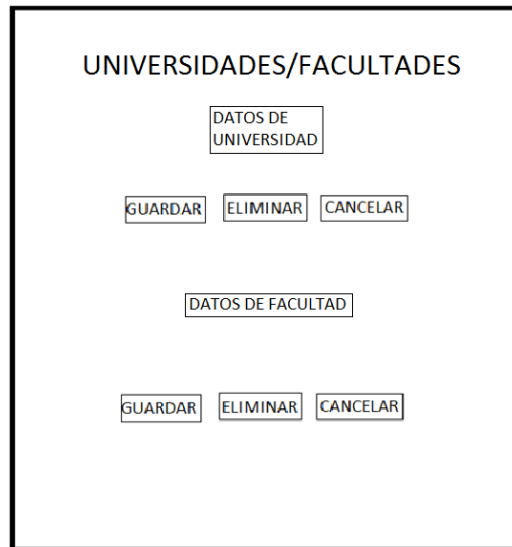


Figura 2.73: Modelo de Interfaz Gráfica de Pantalla de Universidades/Facultades

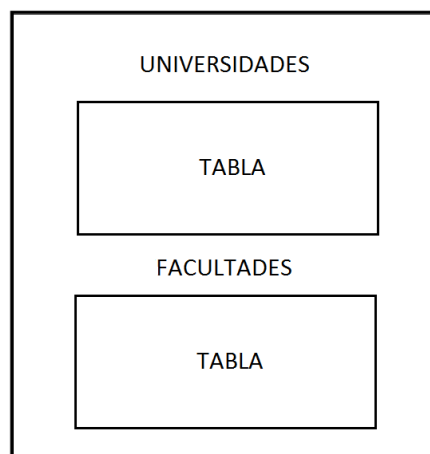


Figura 2.74: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Universidades/Facultades

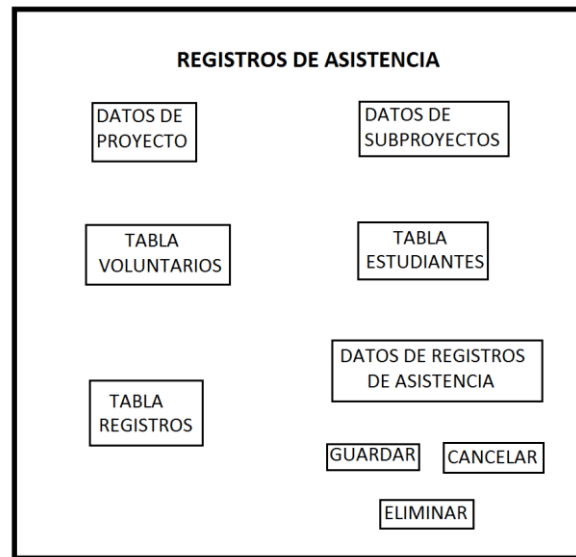


Figura 2.75: Modelo de Interfaz Gráfica de Pantalla de Registros de asistencia



Figura 2.76: Modelo de Interfaz Gráfica de Pantalla de Consulta General de Registros de asistencia

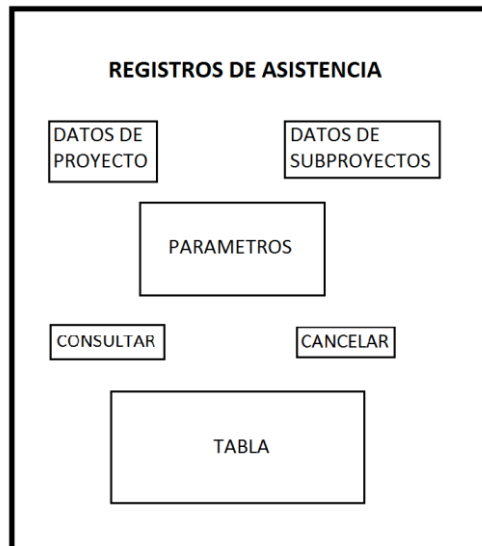


Figura 2.77: Modelo de Interfaz Gráfica de Pantalla de Consulta por parámetros de Registros de asistencia

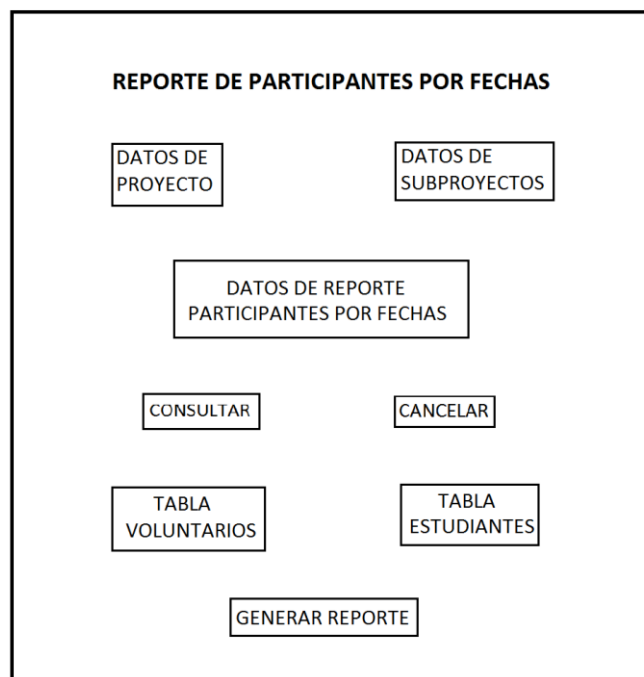


Figura 2.78: Modelo de Interfaz Gráfica de Pantalla de Reporte de participantes por fechas

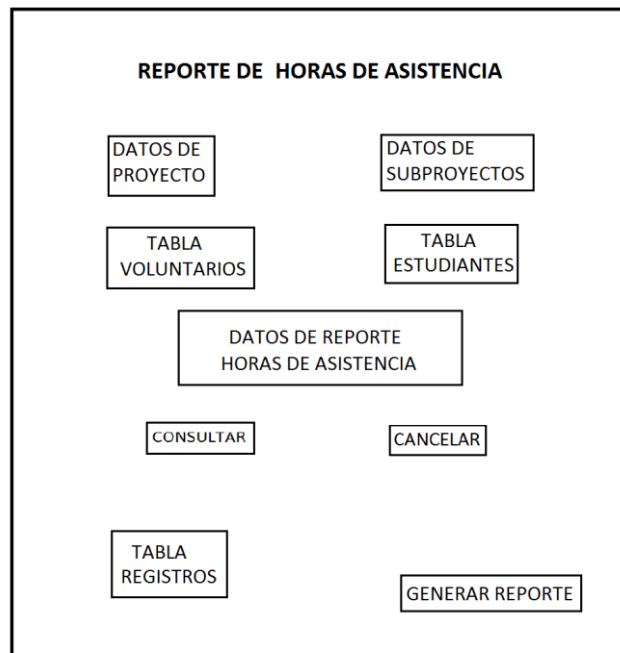


Figura 2.79: Modelo de Interfaz Gráfica de Pantalla de Reporte de horas de asistencia

2.2.4.2. *Requisitos de Hardware y Software*

HARDWARE

- Procesador: Mínimo Pentium 4
- RAM: Mínimo 1 GB
- Disco Duro: Mínimo de 80 GB

SOFTWARE

- JDK 1.7
- JRE 7
- Motor de base de datos MySQL
- MySQL Workbench 6.2.4

CAPITULO 3.- DISEÑO DEL SISTEMA

3.1. Diagramas de paquetes

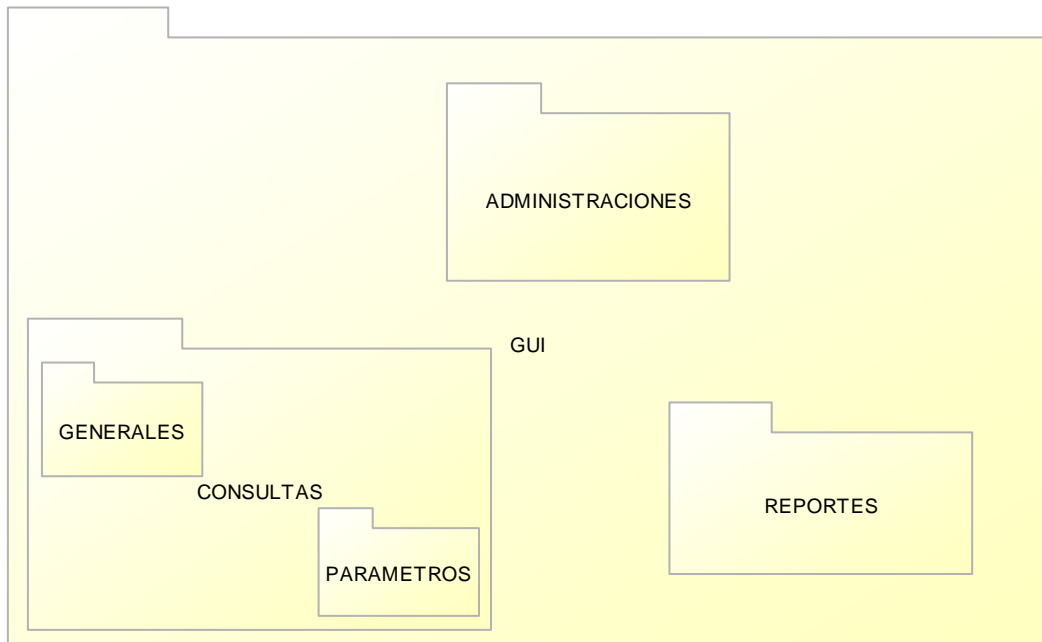


Figura 3.1: Diagrama de paquetes GUI

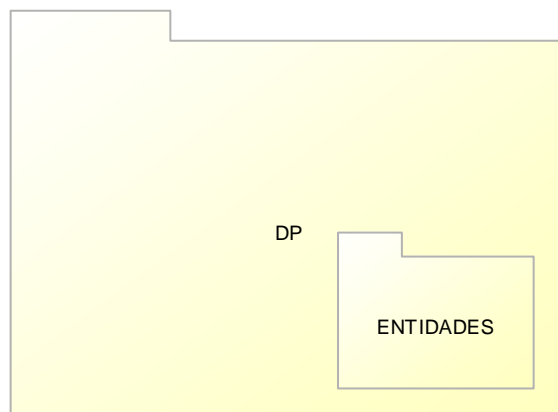


Figura 3.2: Diagrama de paquetes DP

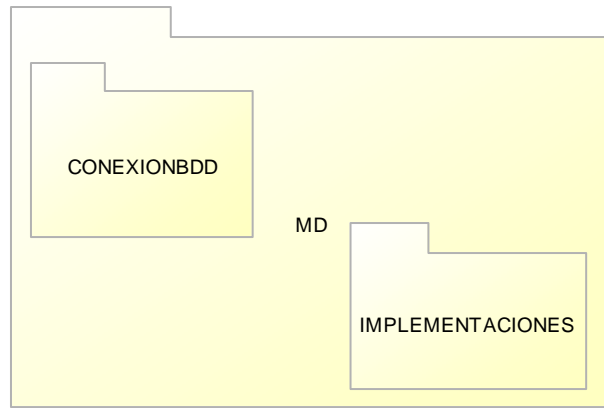


Figura 3.3: Diagrama de paquetes MD



Figura 3.4: Diagrama de paquetes Errores/Imágenes/Funciones

3.2. Diagrama de clases

3.2.1. Clases de Interfaz Gráfica del Usuario

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

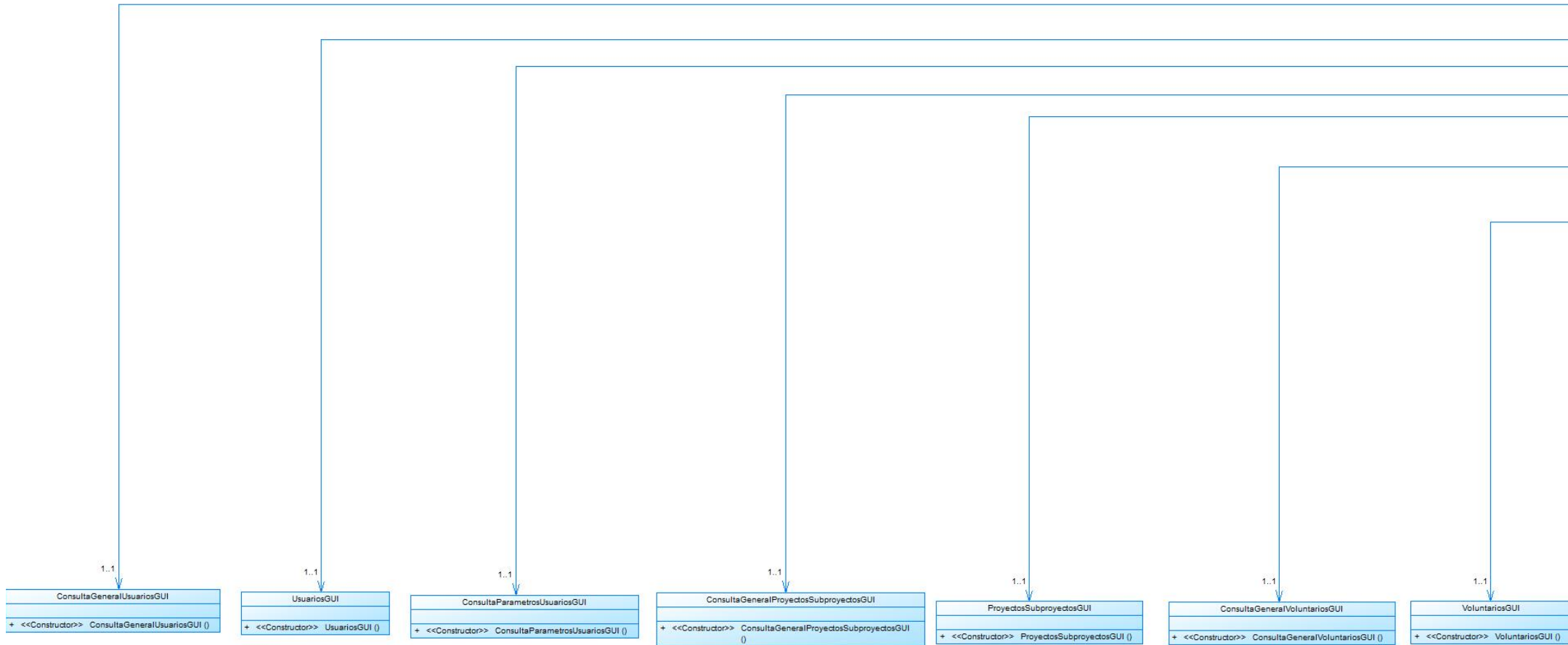


Figura 3.5: Diagrama de Clases de Interfaz Gráfica

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

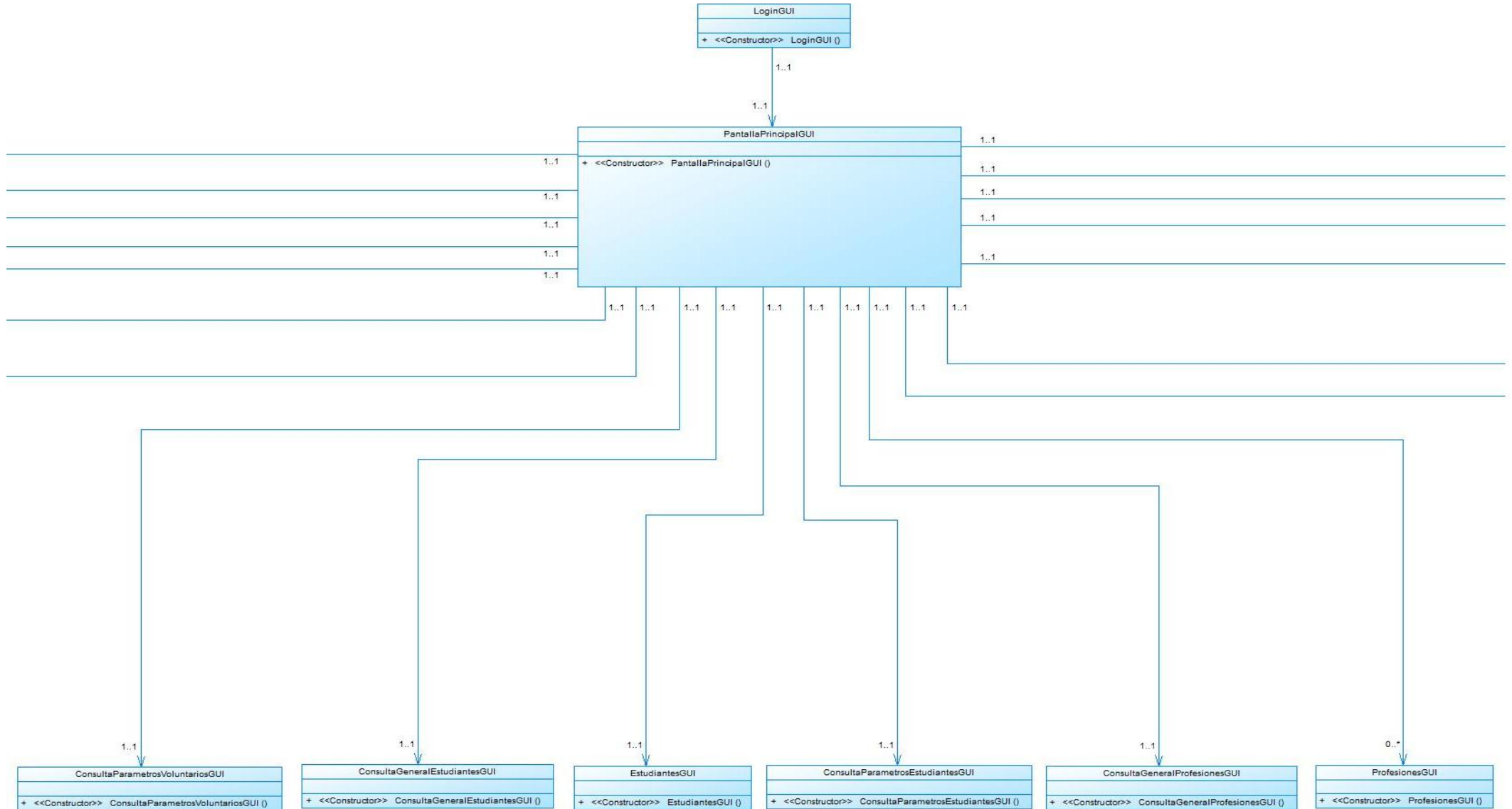


Figura 3.5: Diagrama de Clases de Interfaz Gráfica

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

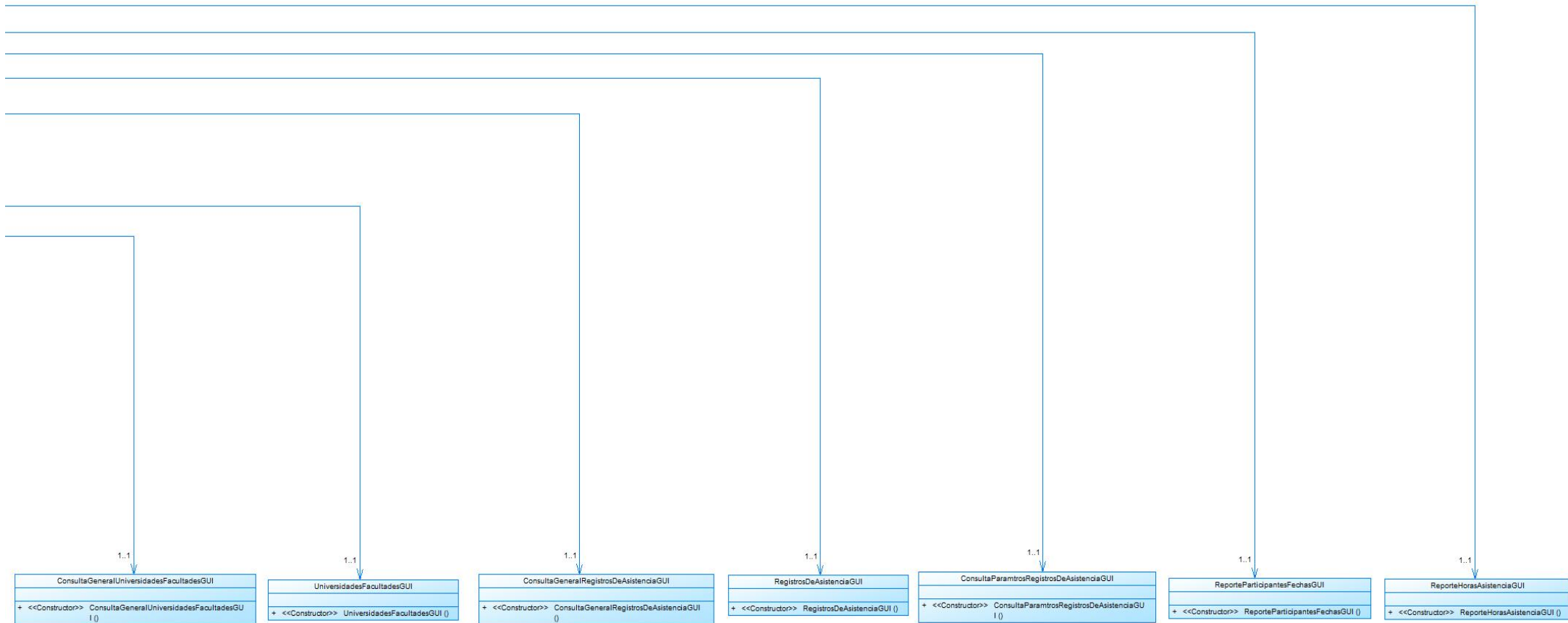


Figura 3.5: Diagrama de Clases de Interfaz Gráfica

3.2.2. Clases del Dominio del Problema

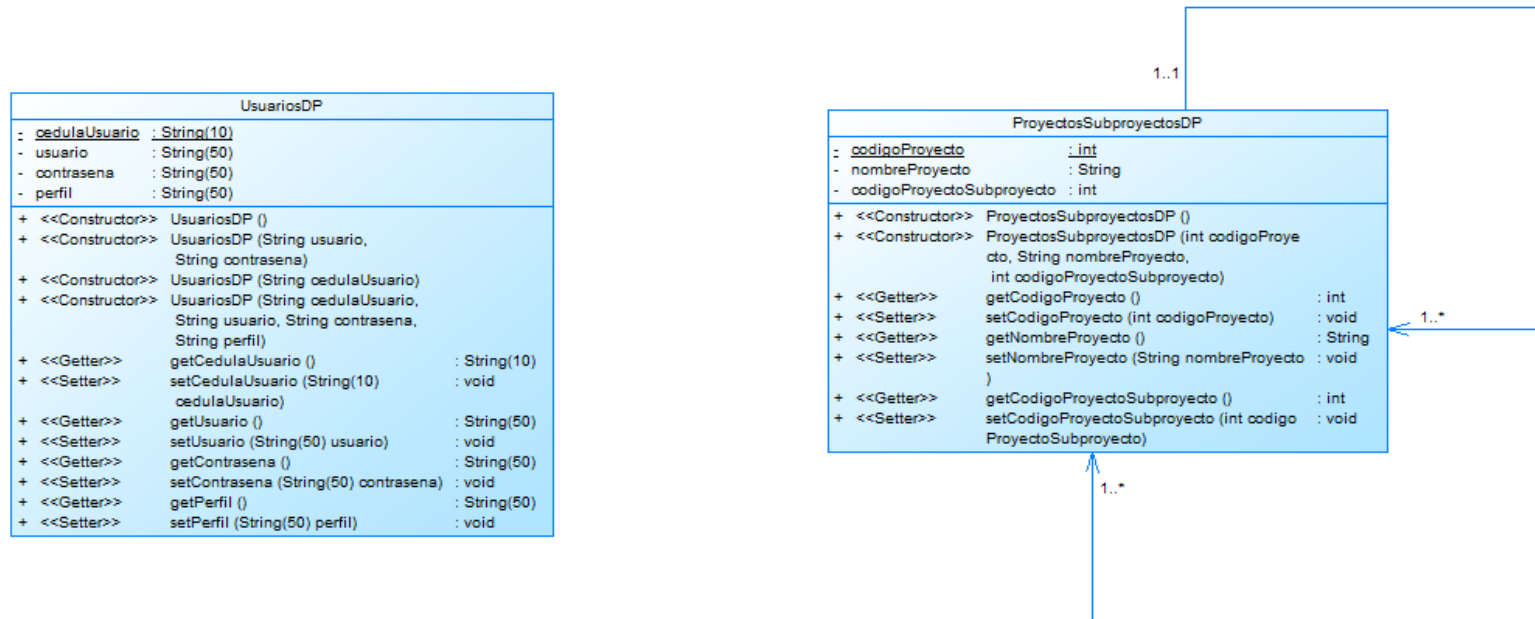


Figura 3.6: Diagrama de Clases del Dominio del Problema

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

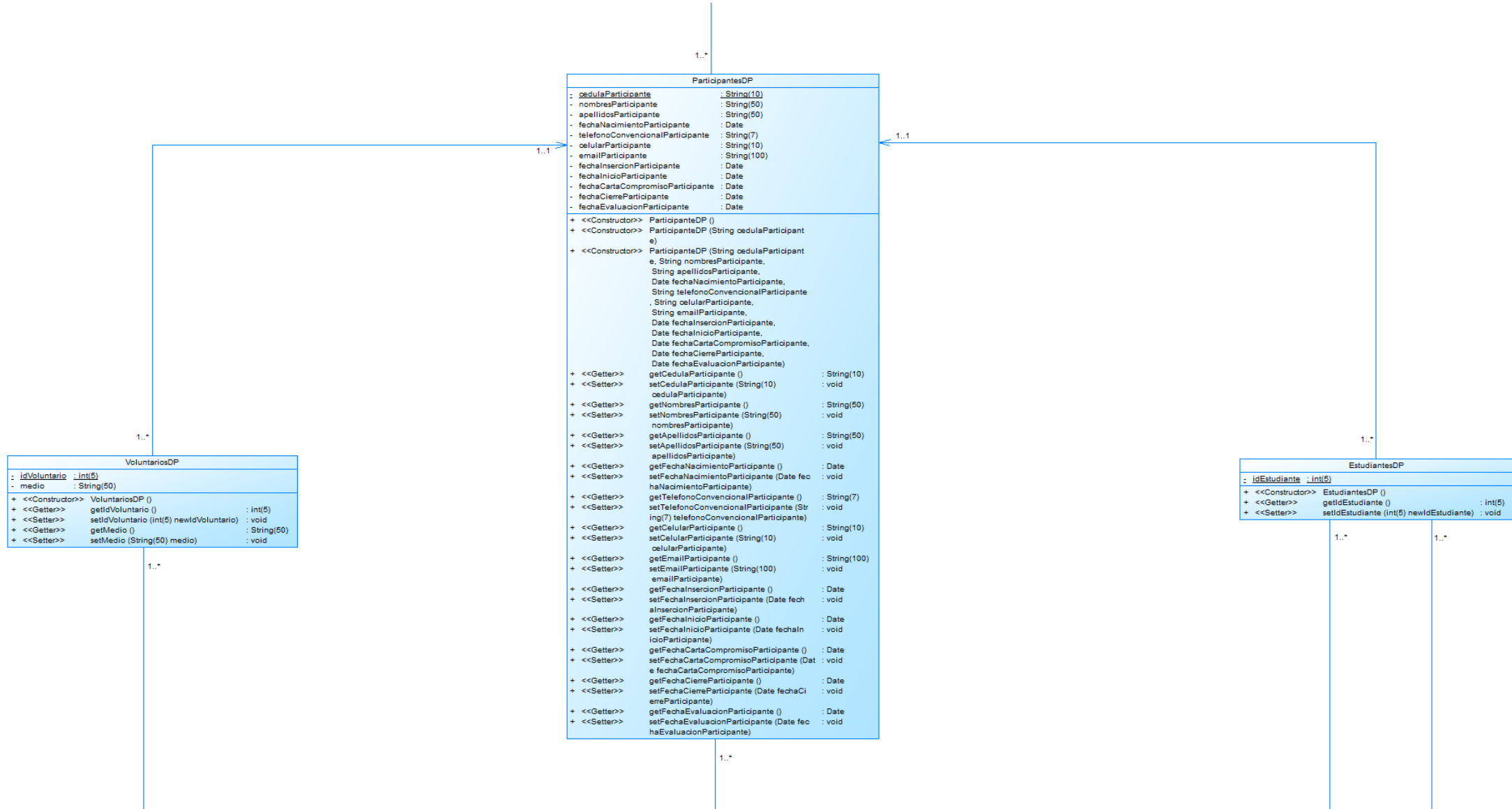


Figura 3.6: Diagrama de Clases del Dominio del Problema

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

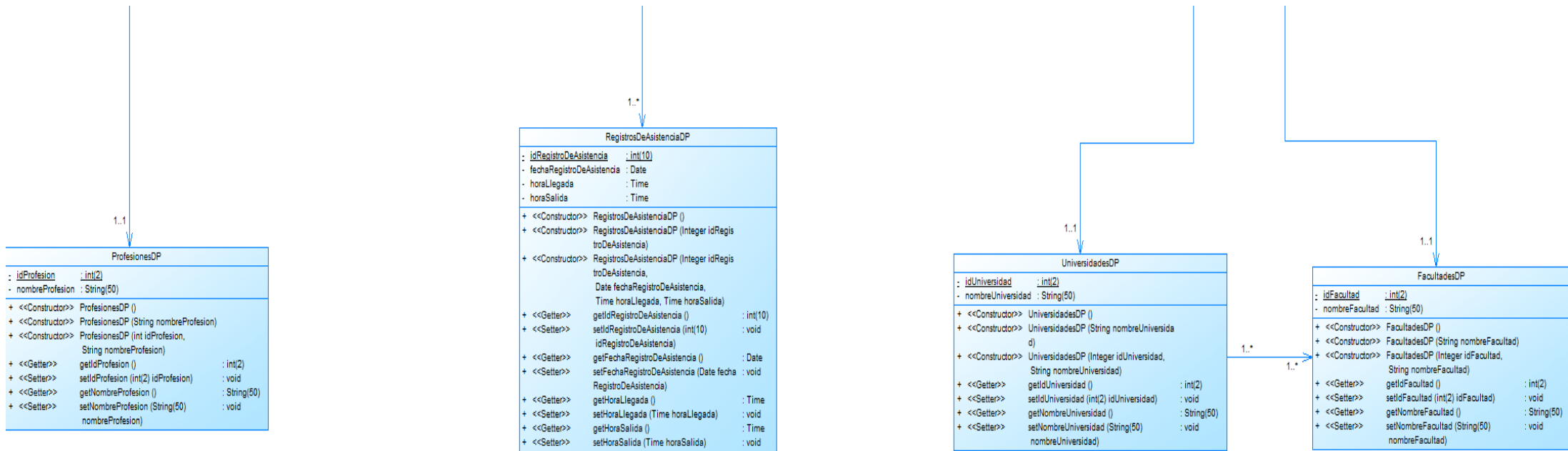


Figura 3.6: Diagrama de Clases del Dominio del Problema

3.2.3. Clases del Manejo de Datos



Figura 3.7: Diagrama de Clases del Manejo de Datos

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

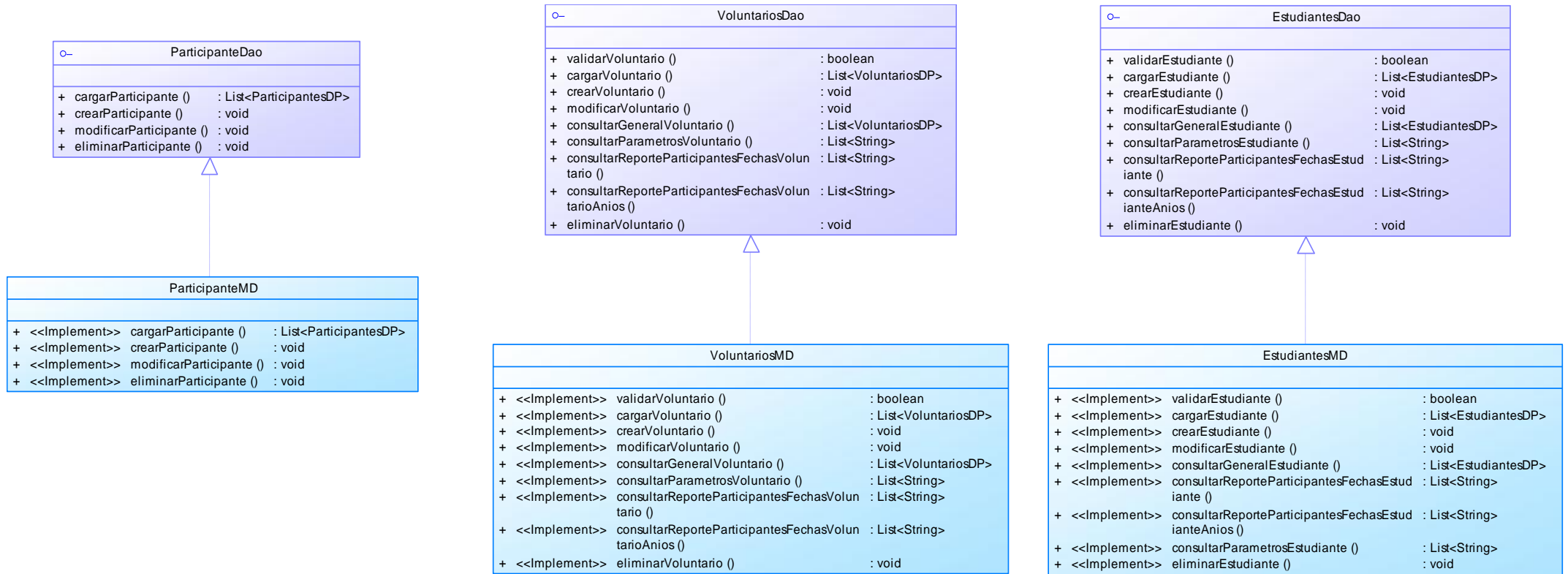


Figura 3.7: Diagrama de Clases del Manejo de Datos

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

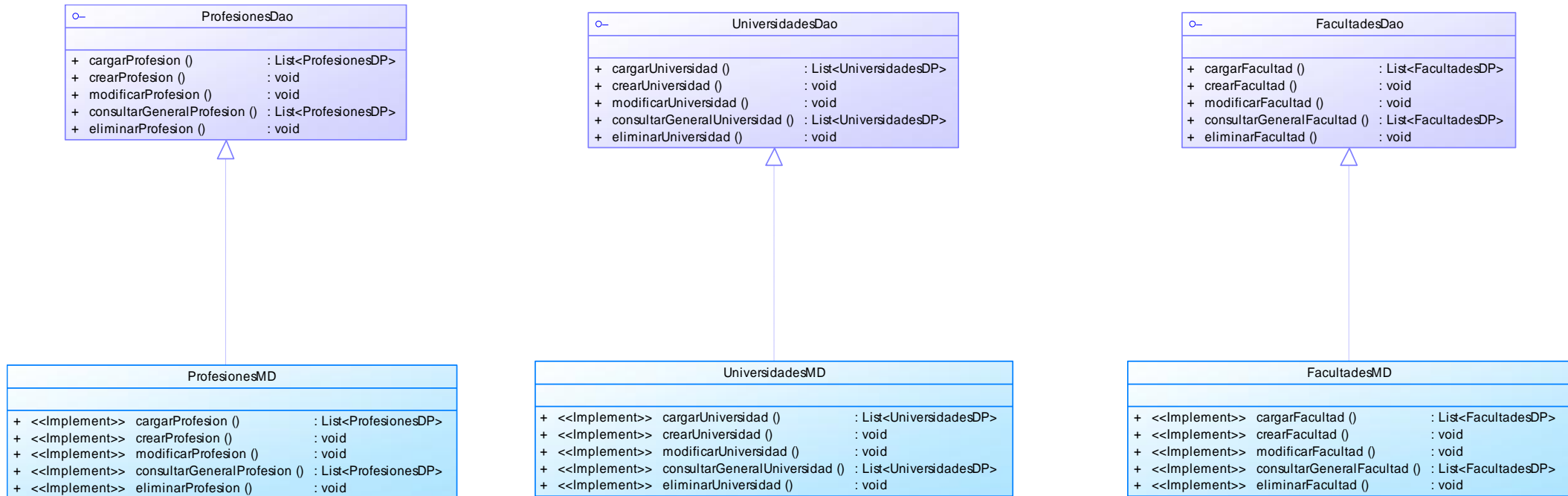


Figura 3.7: Diagrama de Clases del Manejo de Datos

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS



Figura 3.7: Diagrama de Clases del Manejo de Datos

3.3. Diagramas de secuencia

F1: Autenticación de usuario

F1.1: Ingresar al sistema

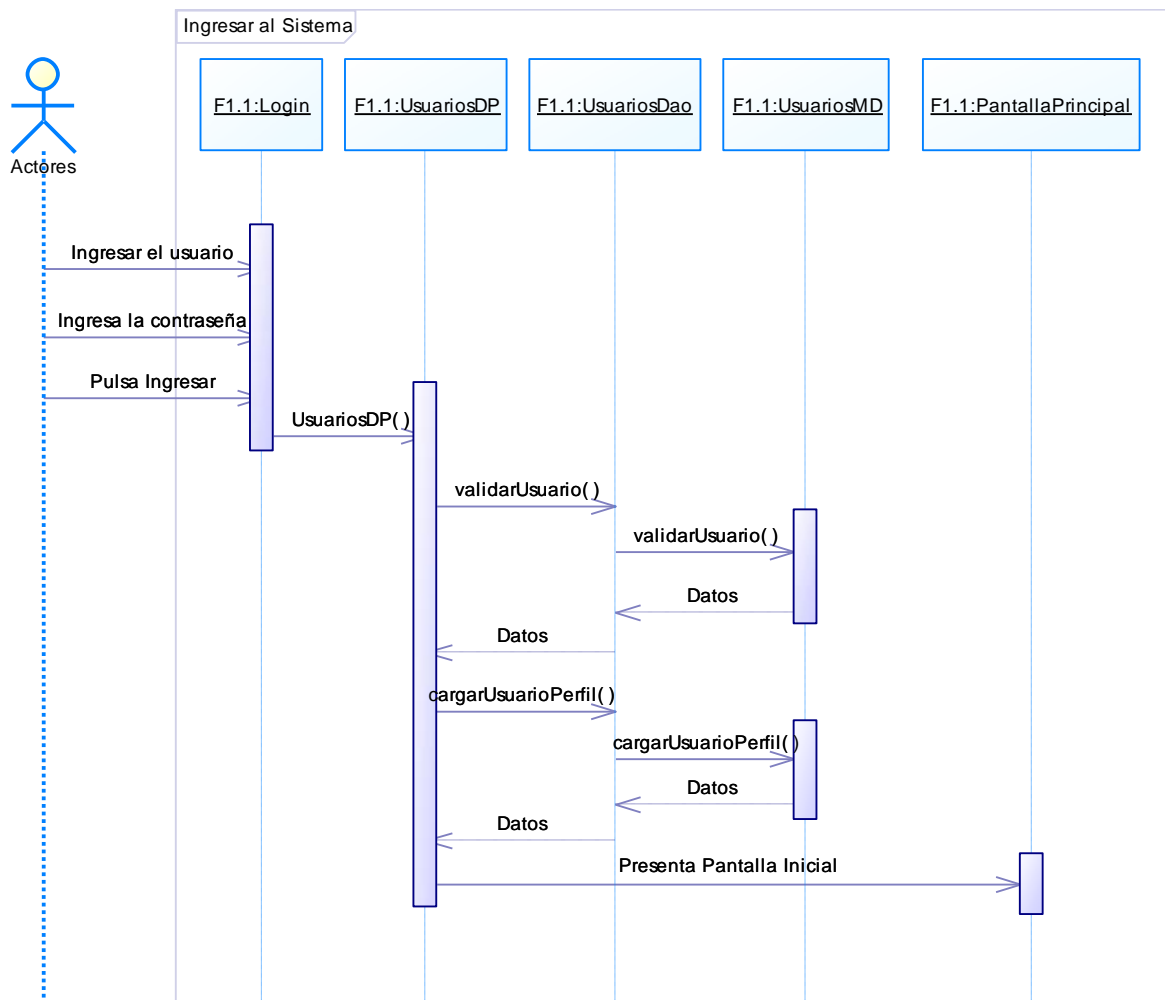


Figura 3.8: Diagrama de Secuencia Ingresar al sistema

F2: Administración de usuarios

F2.1: Crear usuario

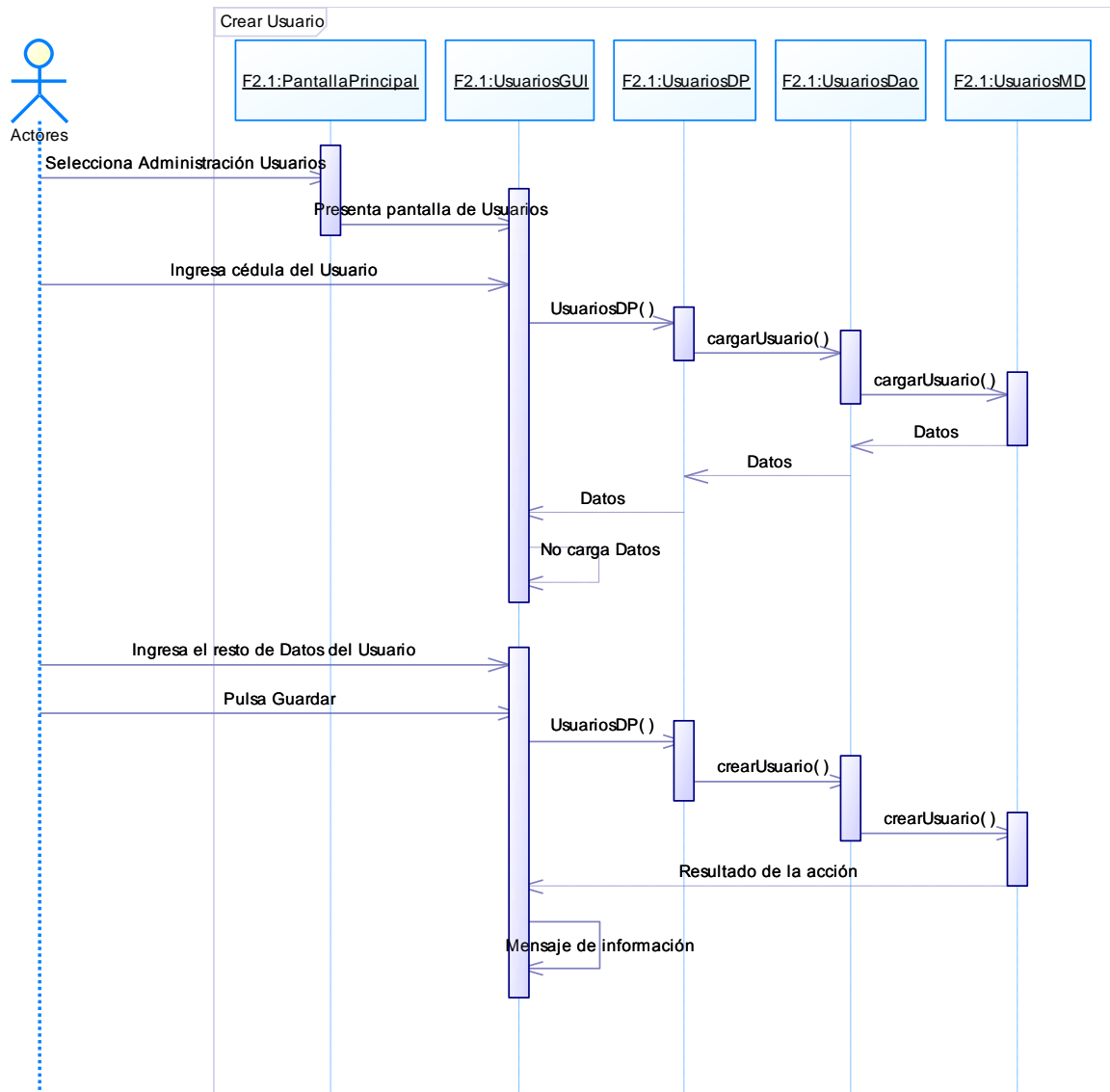


Figura 3.9: Diagrama de Secuencia Crear Usuario

F2.2: Modificar usuario

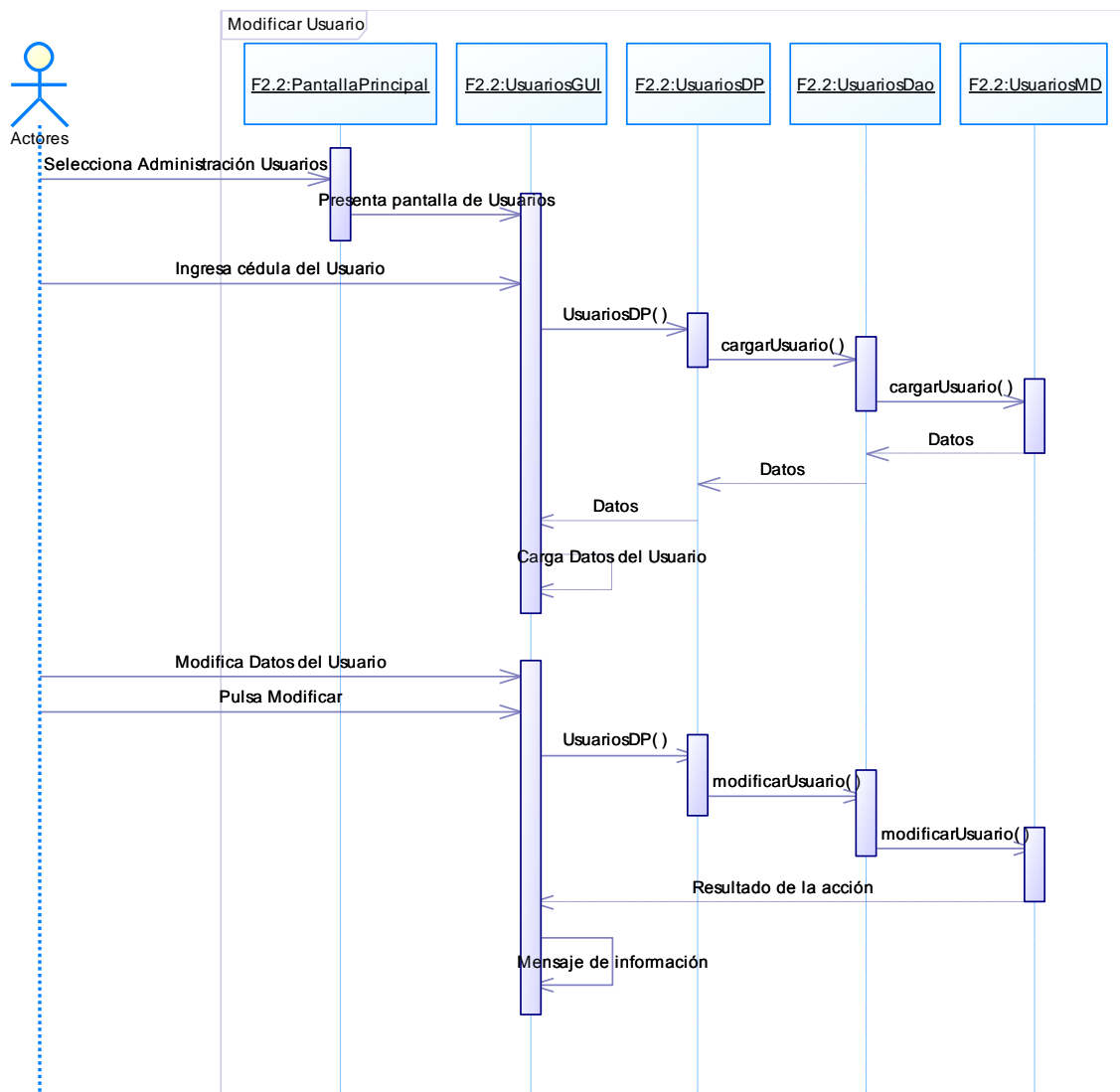


Figura 3.10: Diagrama de Secuencia Modificar Usuario

F2.3: Consultar usuario

F2.3.1: Consulta general

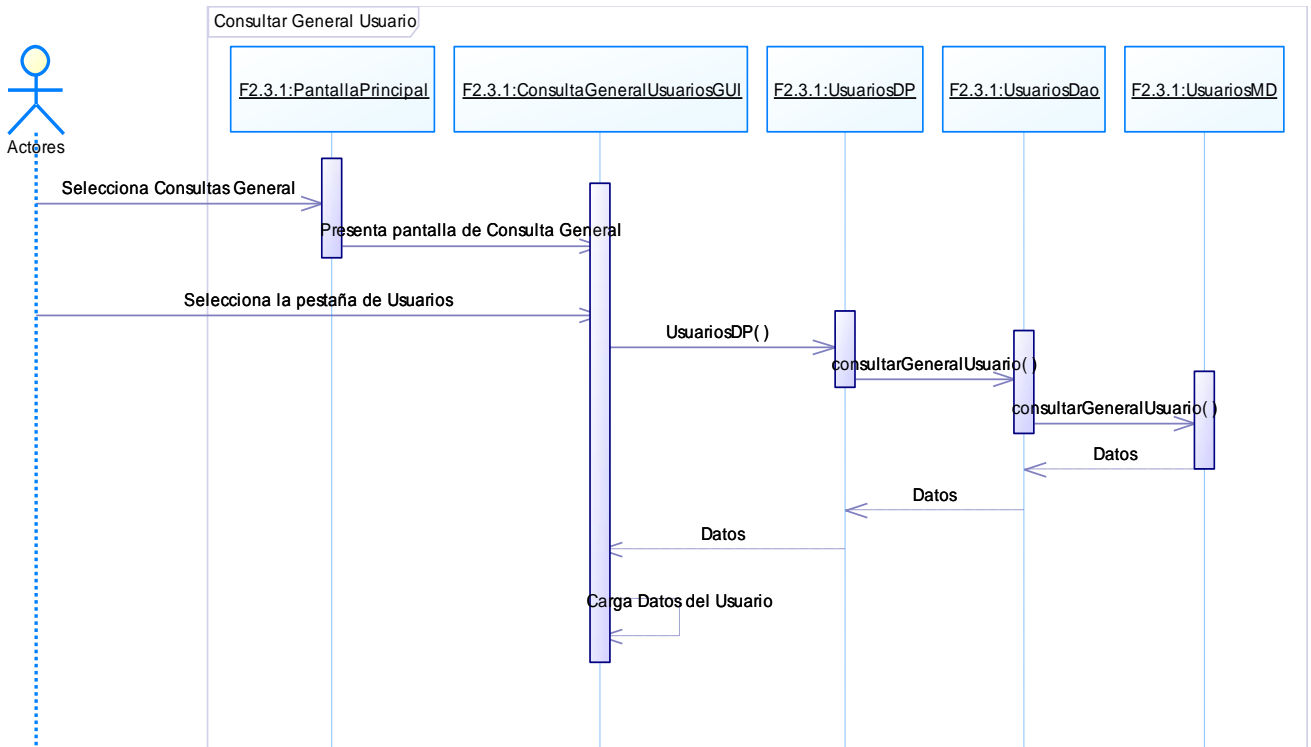


Figura 3.11: Diagrama de Secuencia Consulta General Usuario

F2.3.2: Consulta por parámetros

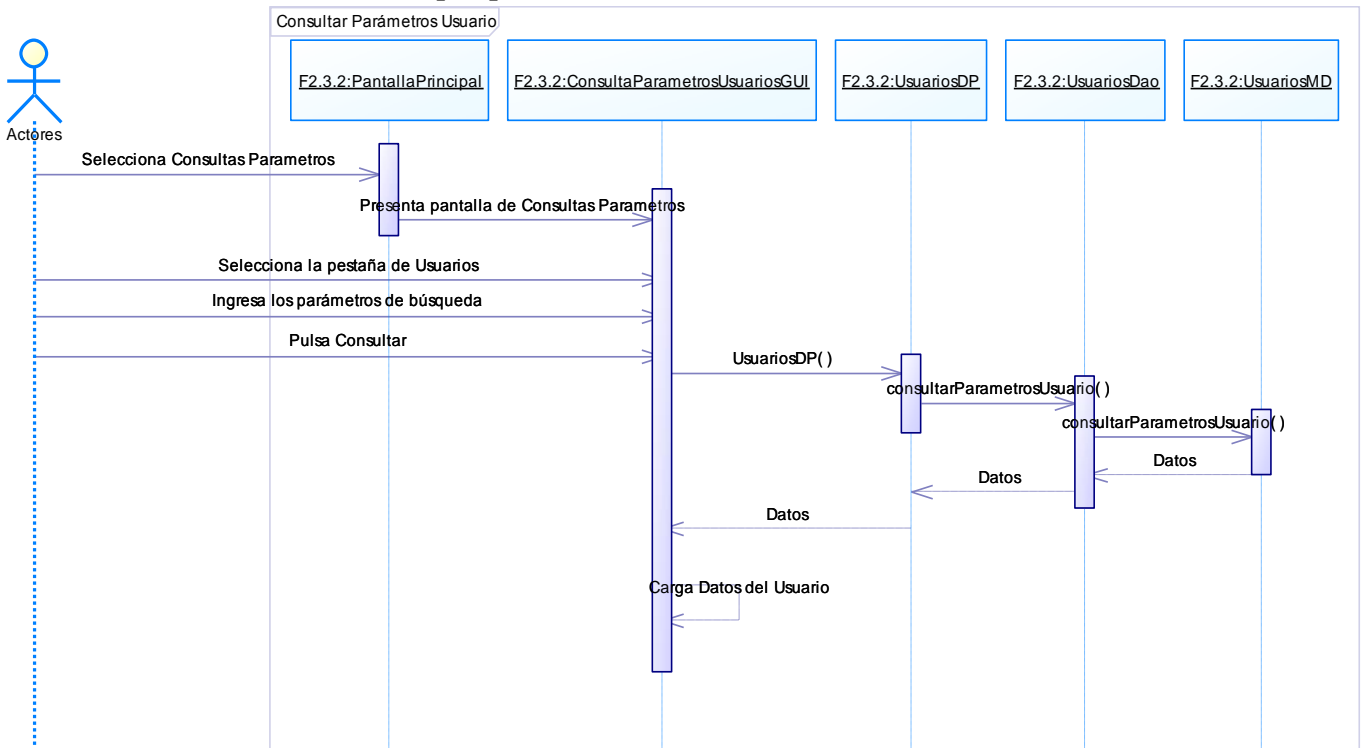


Figura 3.12: Diagrama de Secuencia Consulta por parámetros Usuario

F2.4: Eliminar usuario

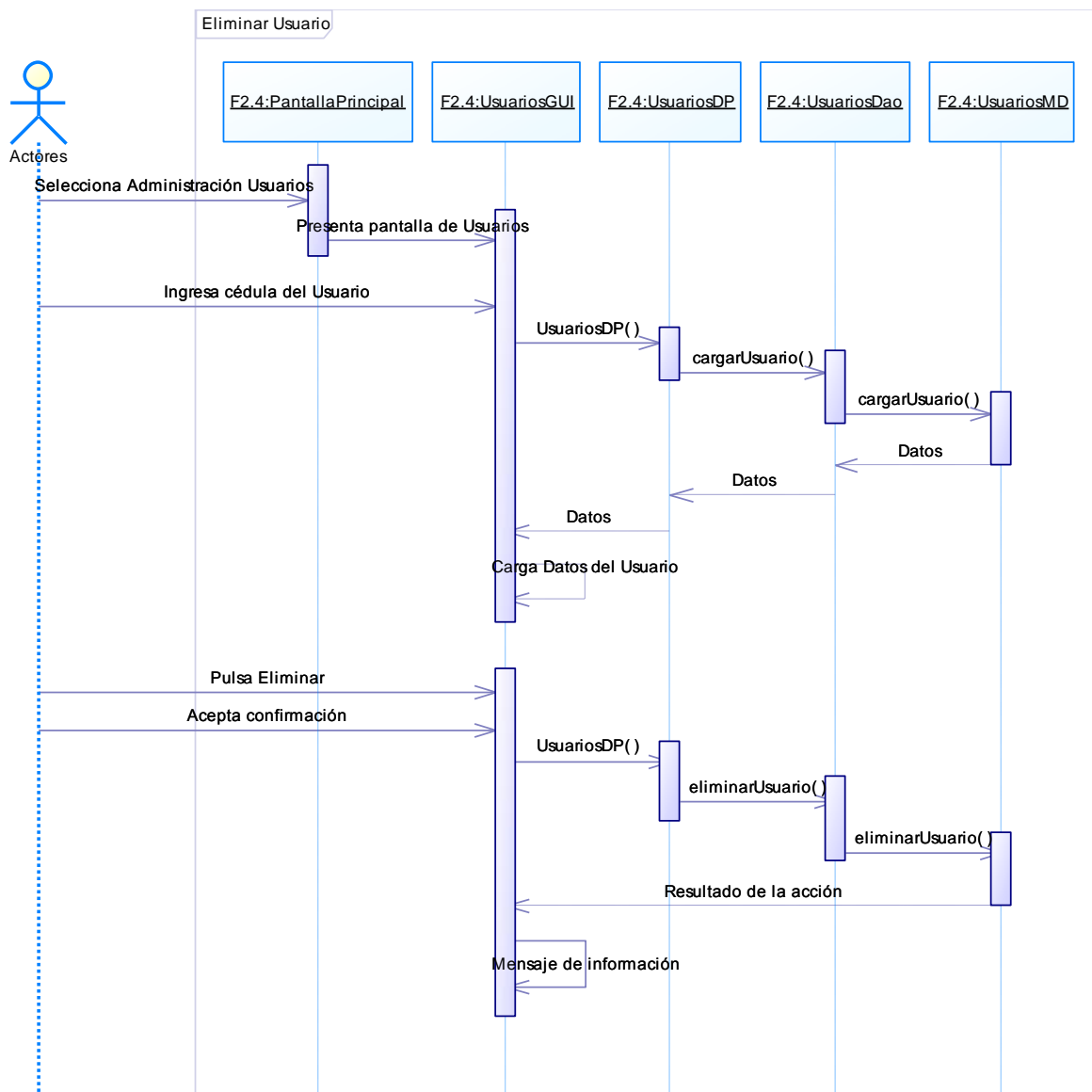


Figura 3.13: Diagrama de Secuencia Eliminar Usuario

F3: Administración de proyectos

F3.1: Crear proyecto

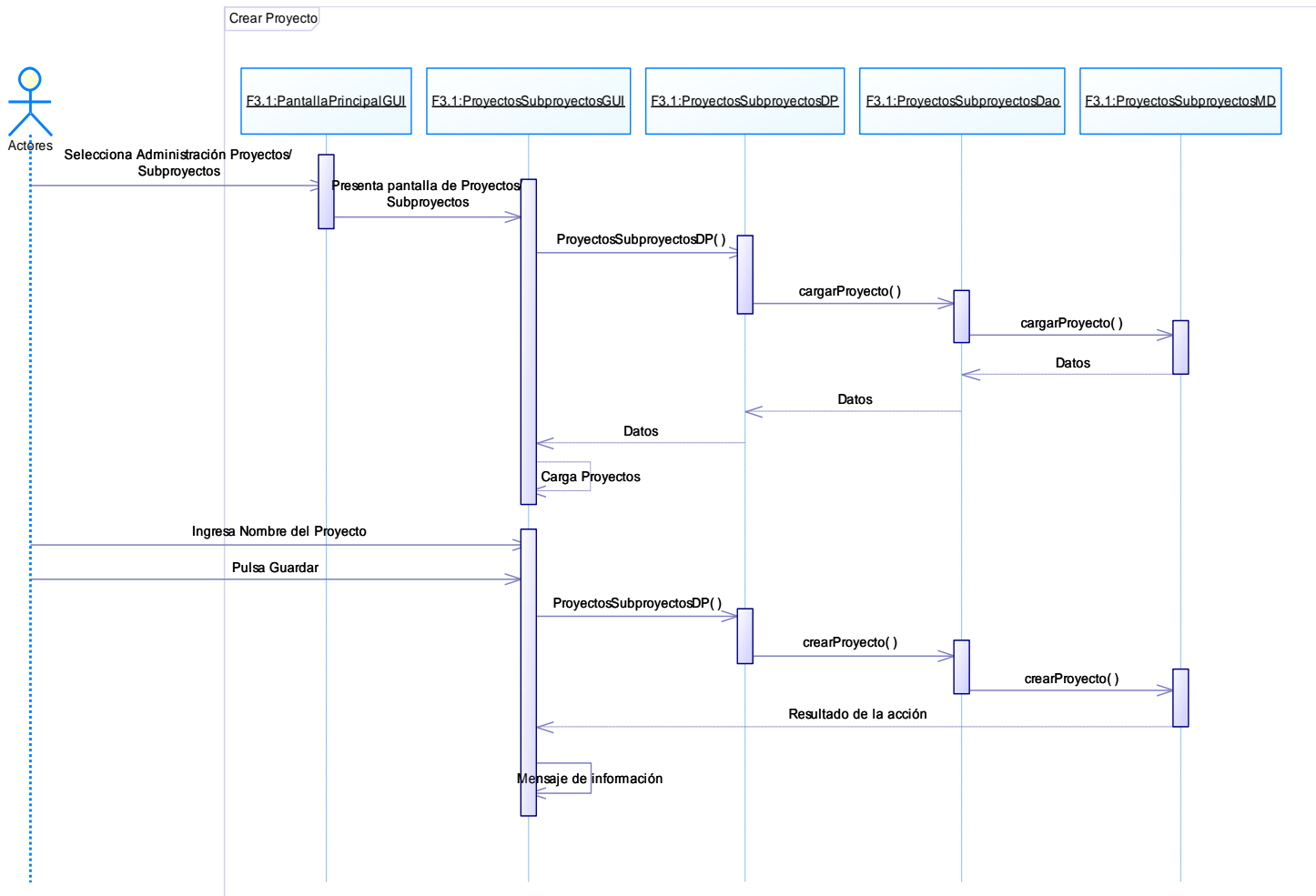


Figura 3.14: Diagrama de Secuencia Crear Proyecto

F3.2: Modificar proyecto

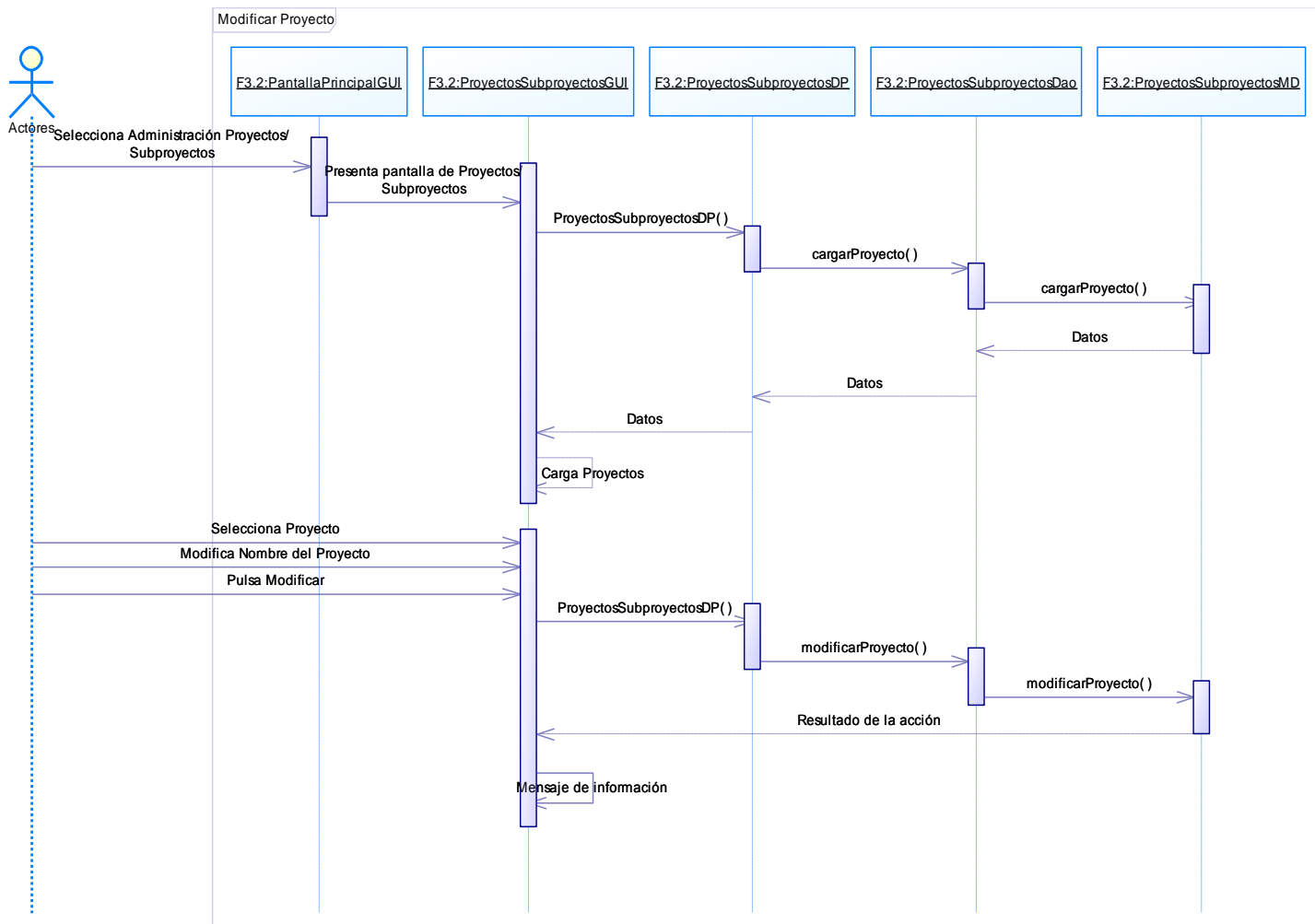


Figura 3.15: Diagrama de Secuencia Modificar Proyecto

F3.3: Consultar proyecto

F3.3.1: Consulta general

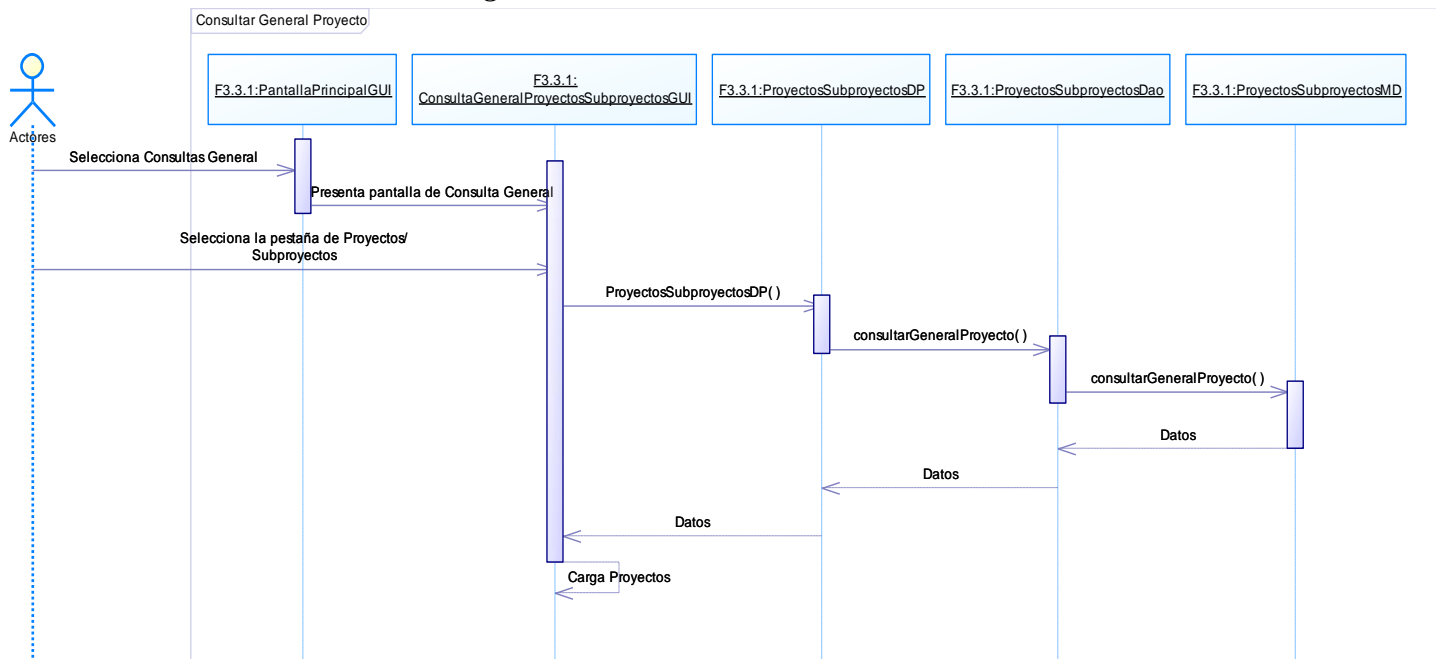


Figura 3.16: Diagrama de Secuencia Consulta general Proyecto

F3.4: Eliminar proyecto

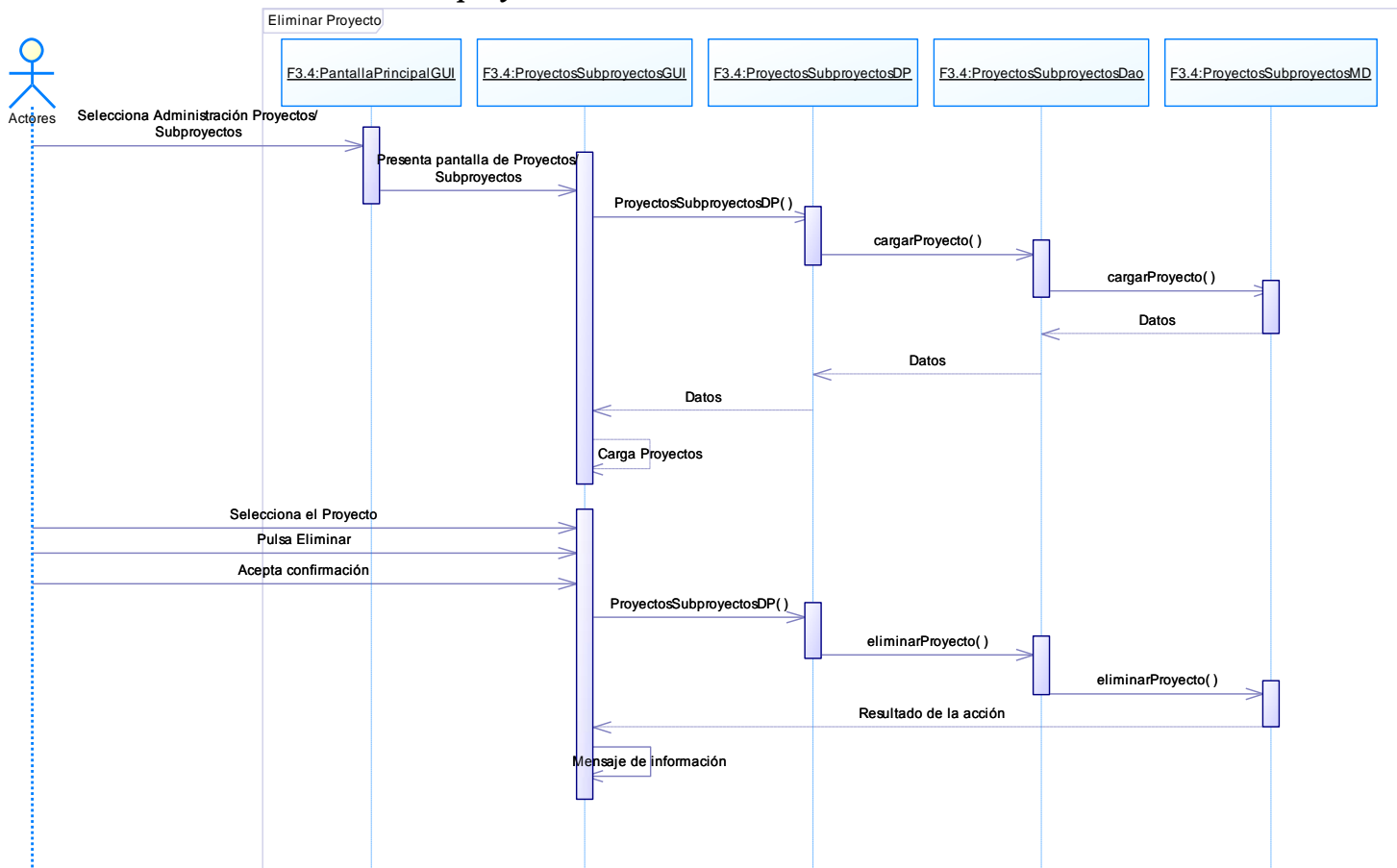


Figura 3.17: Diagrama de Secuencia Eliminar Proyecto

F4: Administración de subproyectos

F4.1: Crear subproyecto

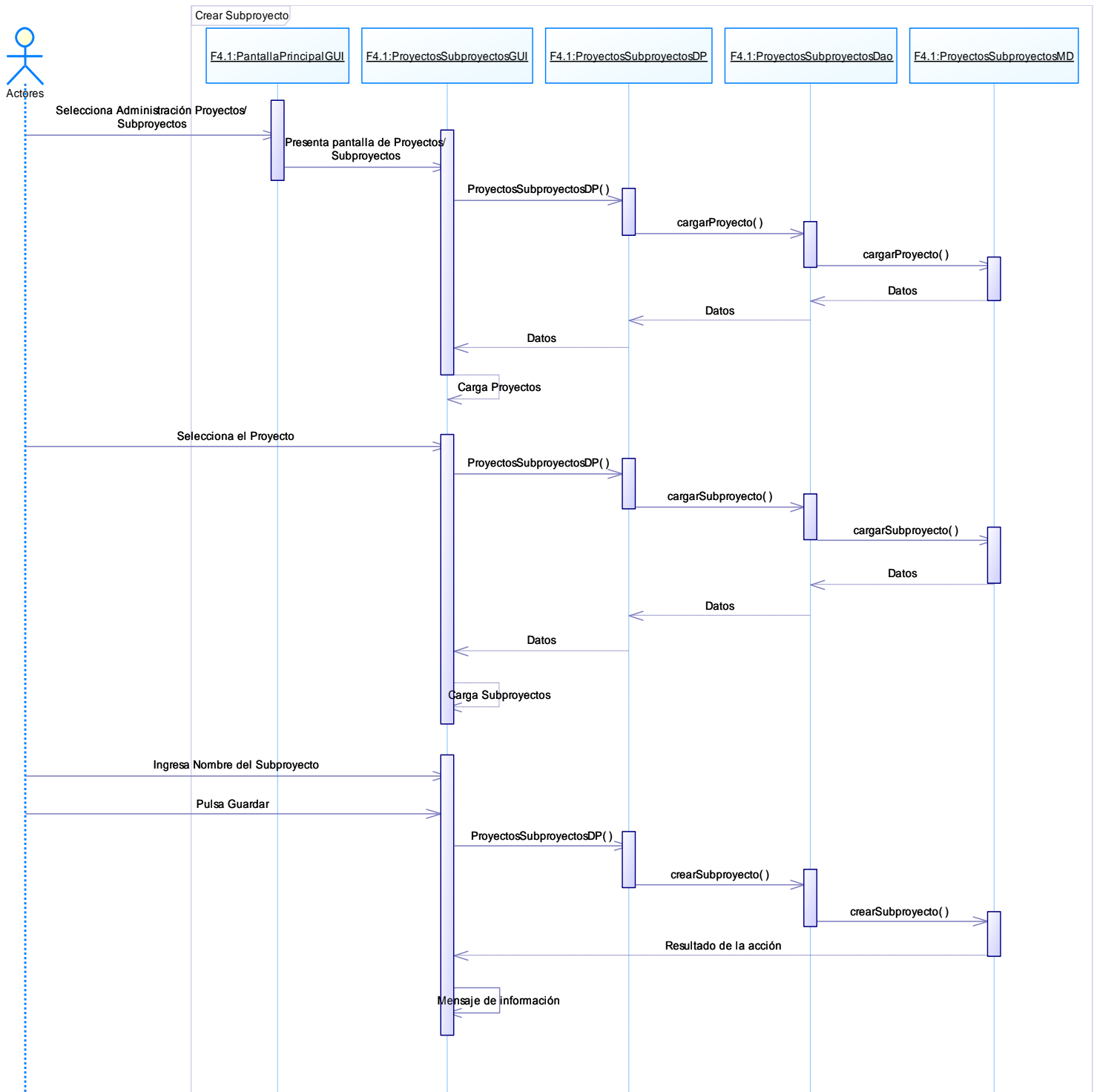


Figura 3.18: Diagrama de Secuencia Crear subproyecto

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

F4.2: Modificar subproyecto

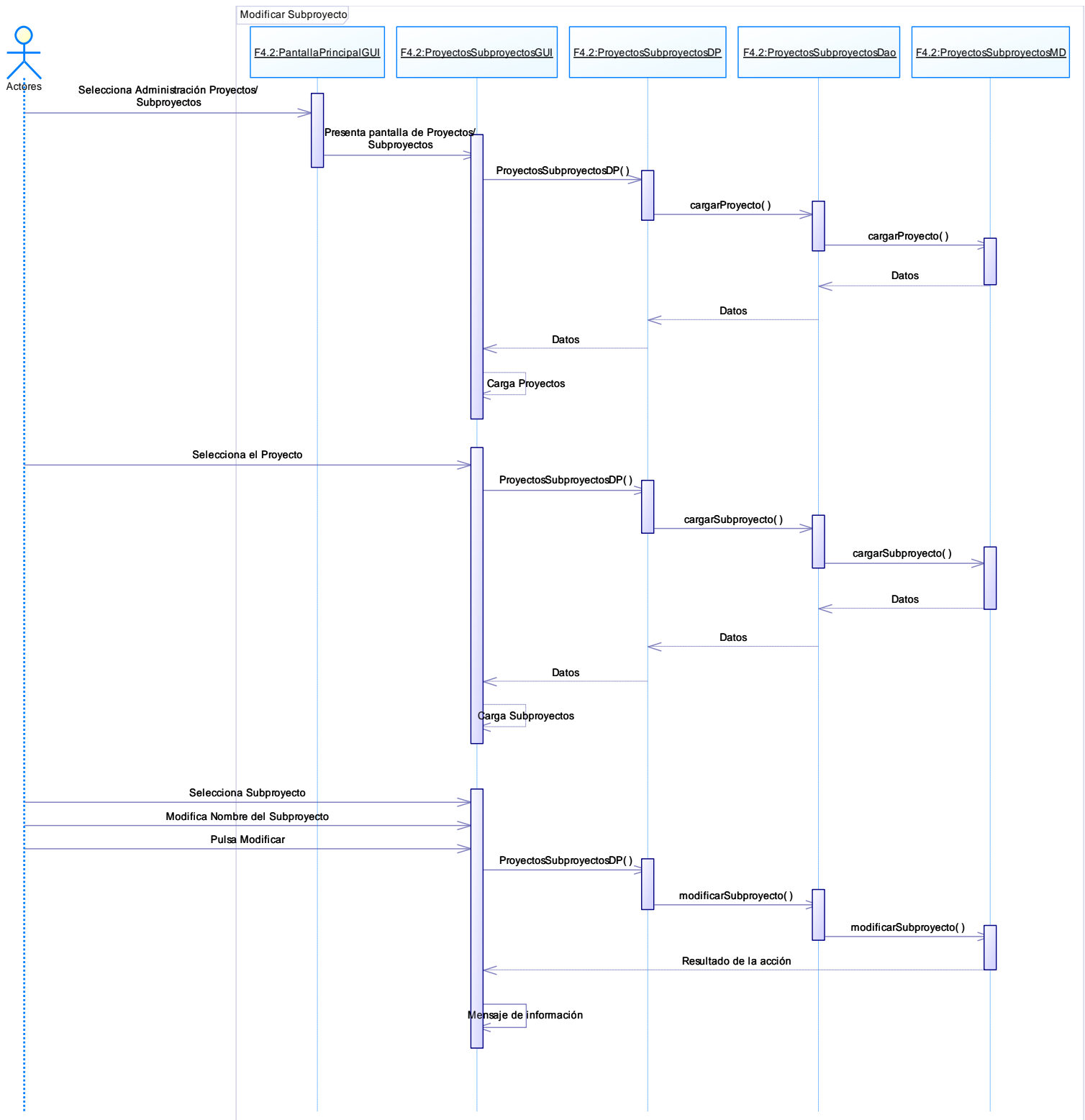


Figura 3.19: Diagrama de Secuencia Modificar subproyecto

F4.3: Consultar subproyecto

F4.3.1: Consulta general

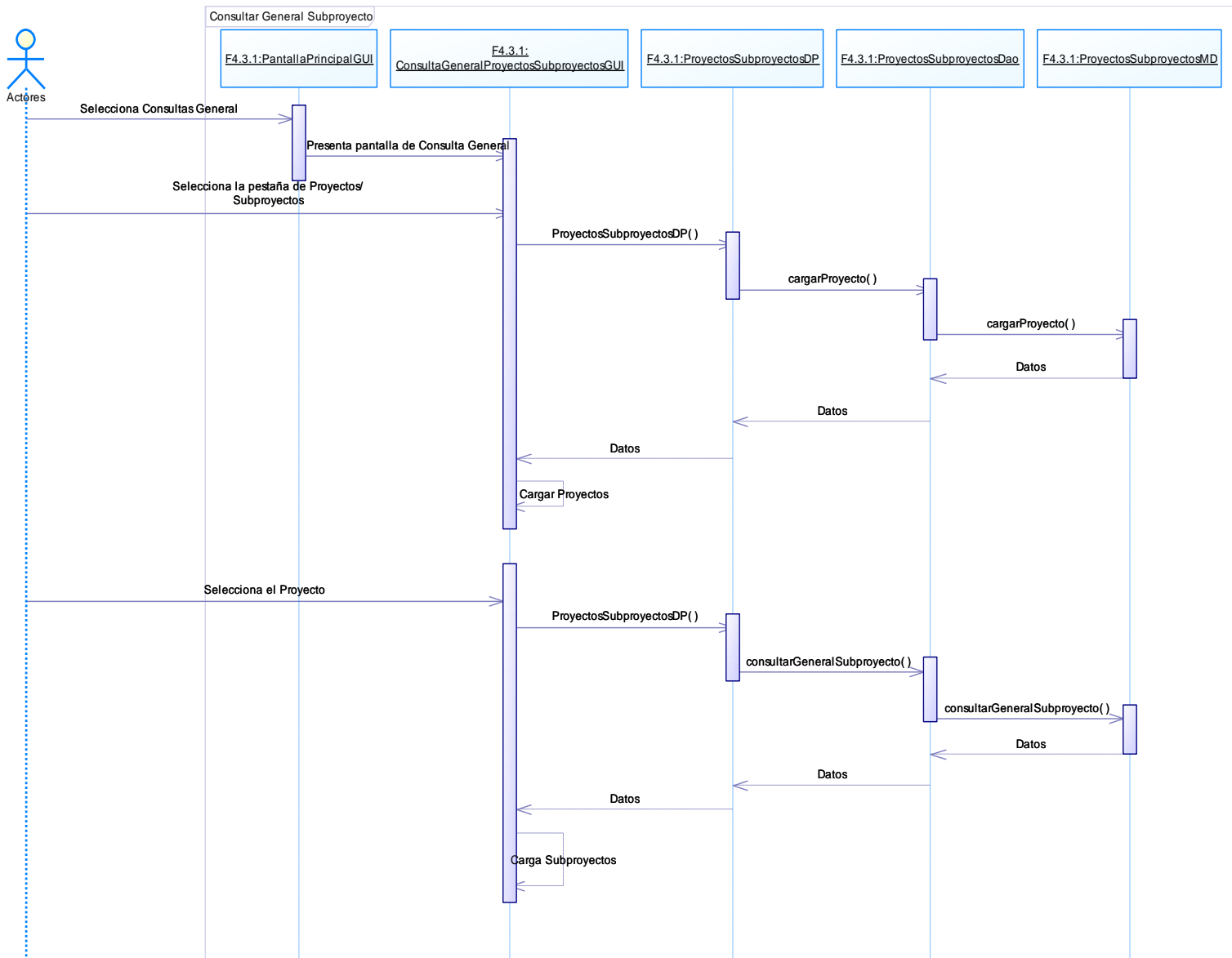


Figura 3.20: Diagrama de Secuencia Consulta general subproyecto

F4.4: Eliminar subproyecto

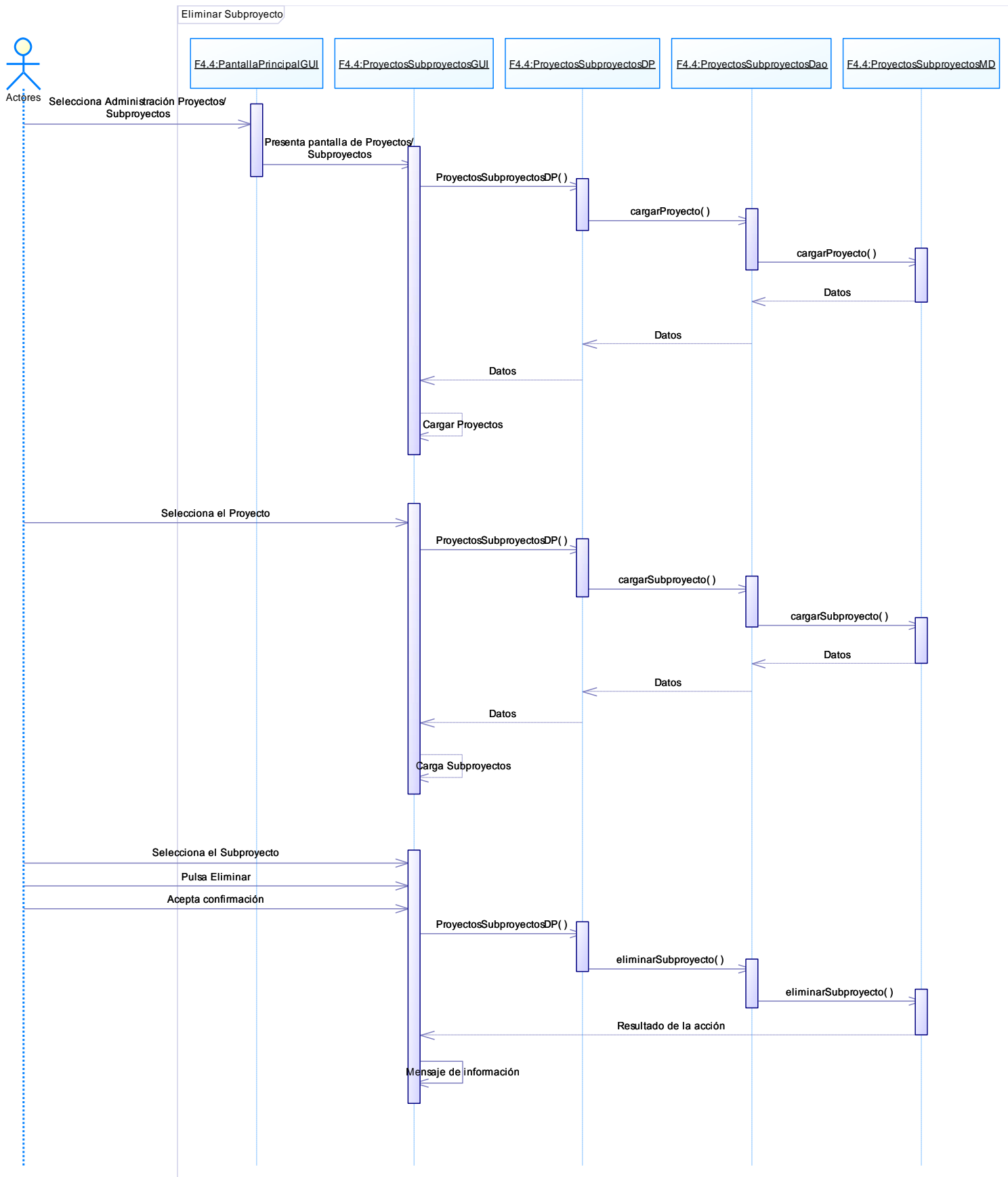
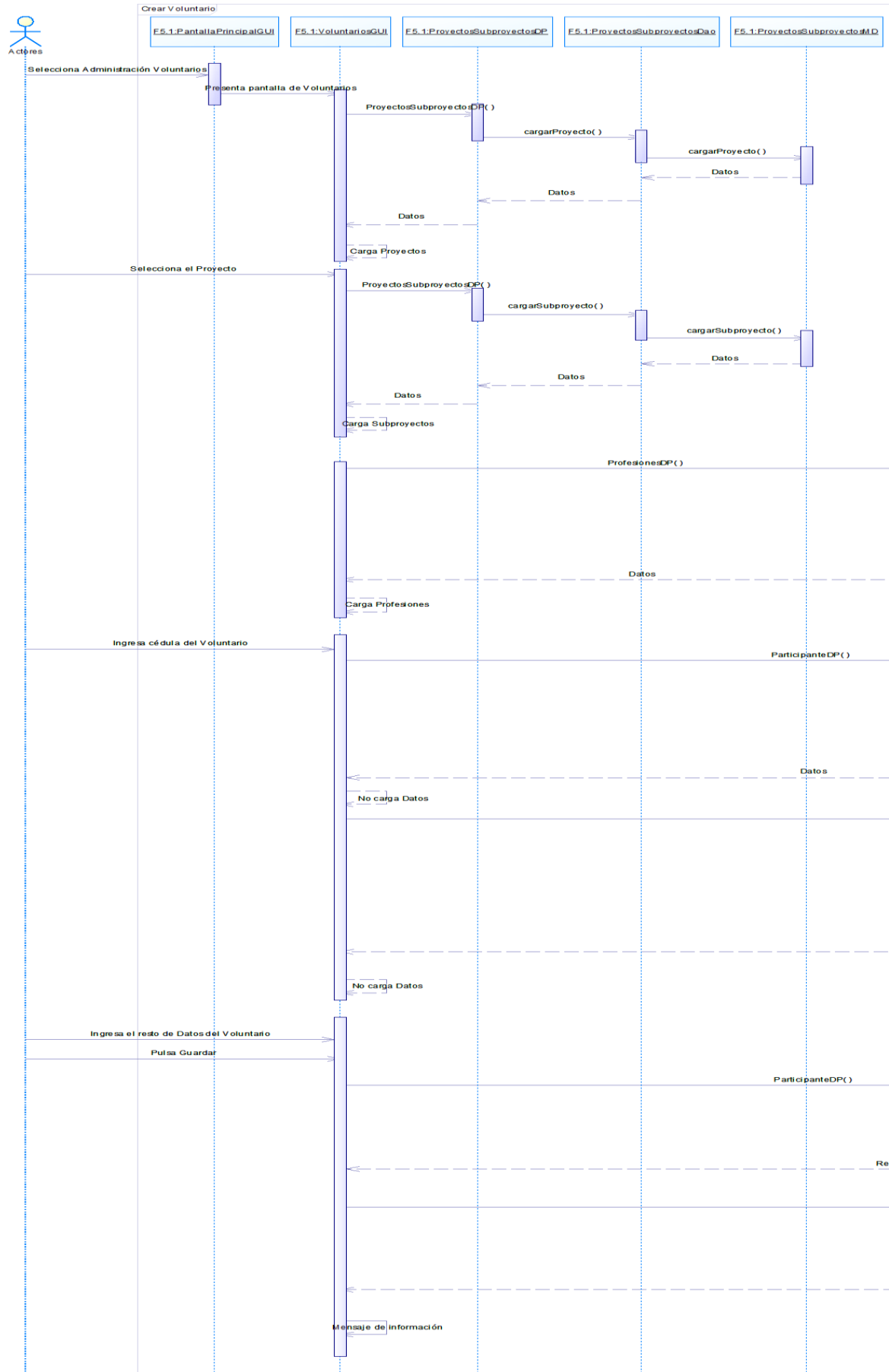


Figura 3.21: Diagrama de Secuencia Eliminar subproyecto

F5: Administración de voluntarios

F5.1: Crear voluntario



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

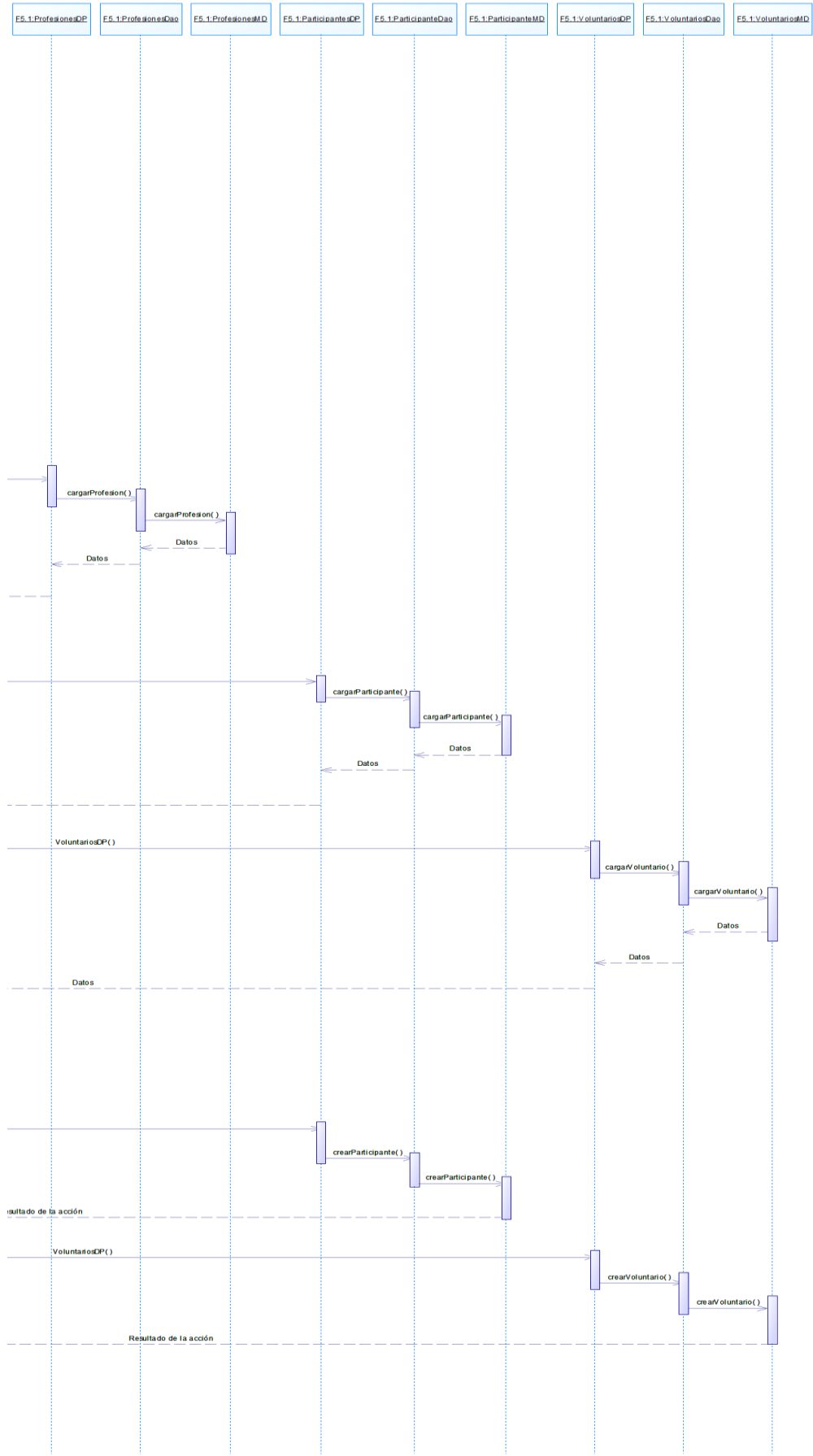
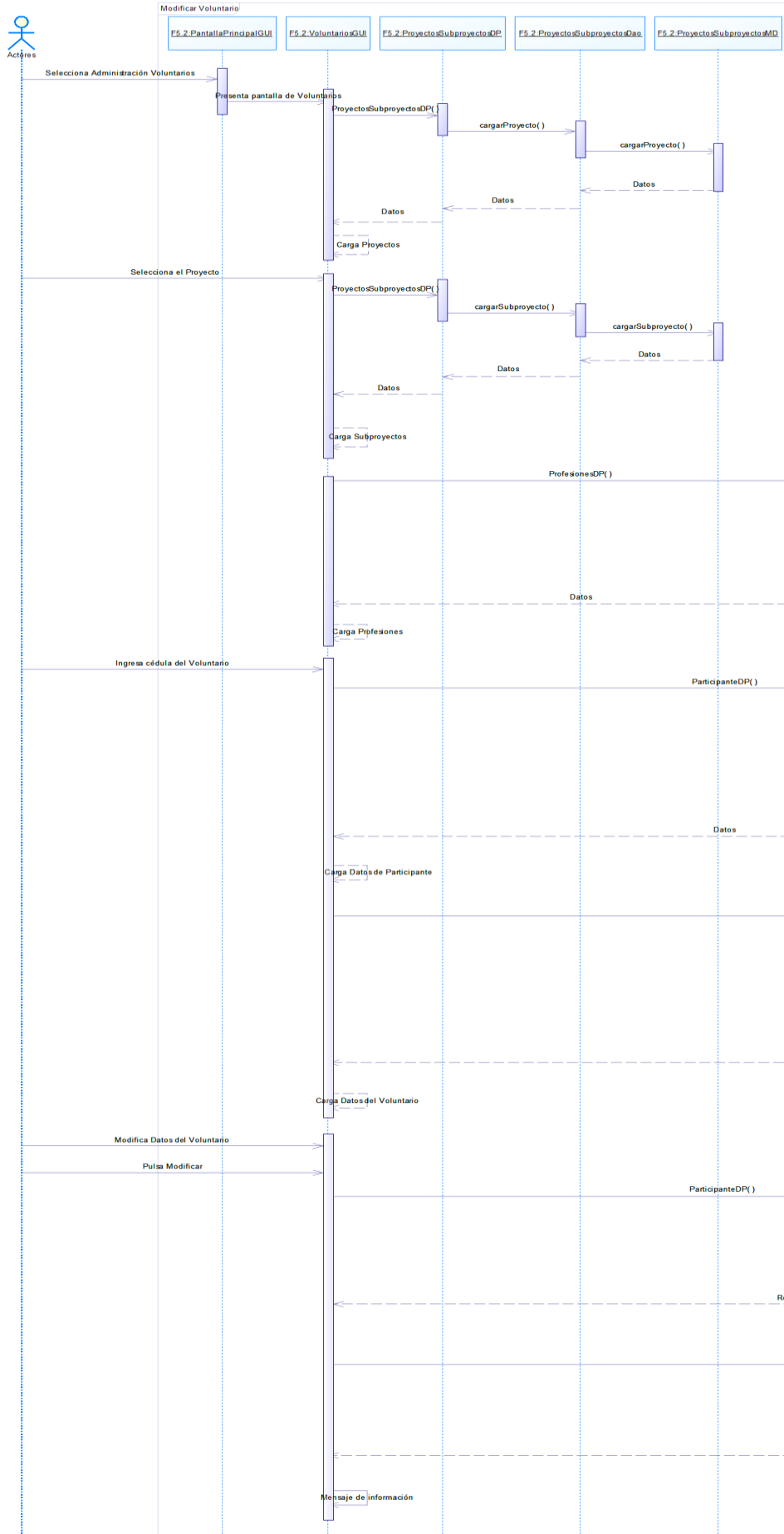


Figura 3.22: Diagrama de Secuencia Crear voluntario

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

F5.2: Modificar voluntario



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

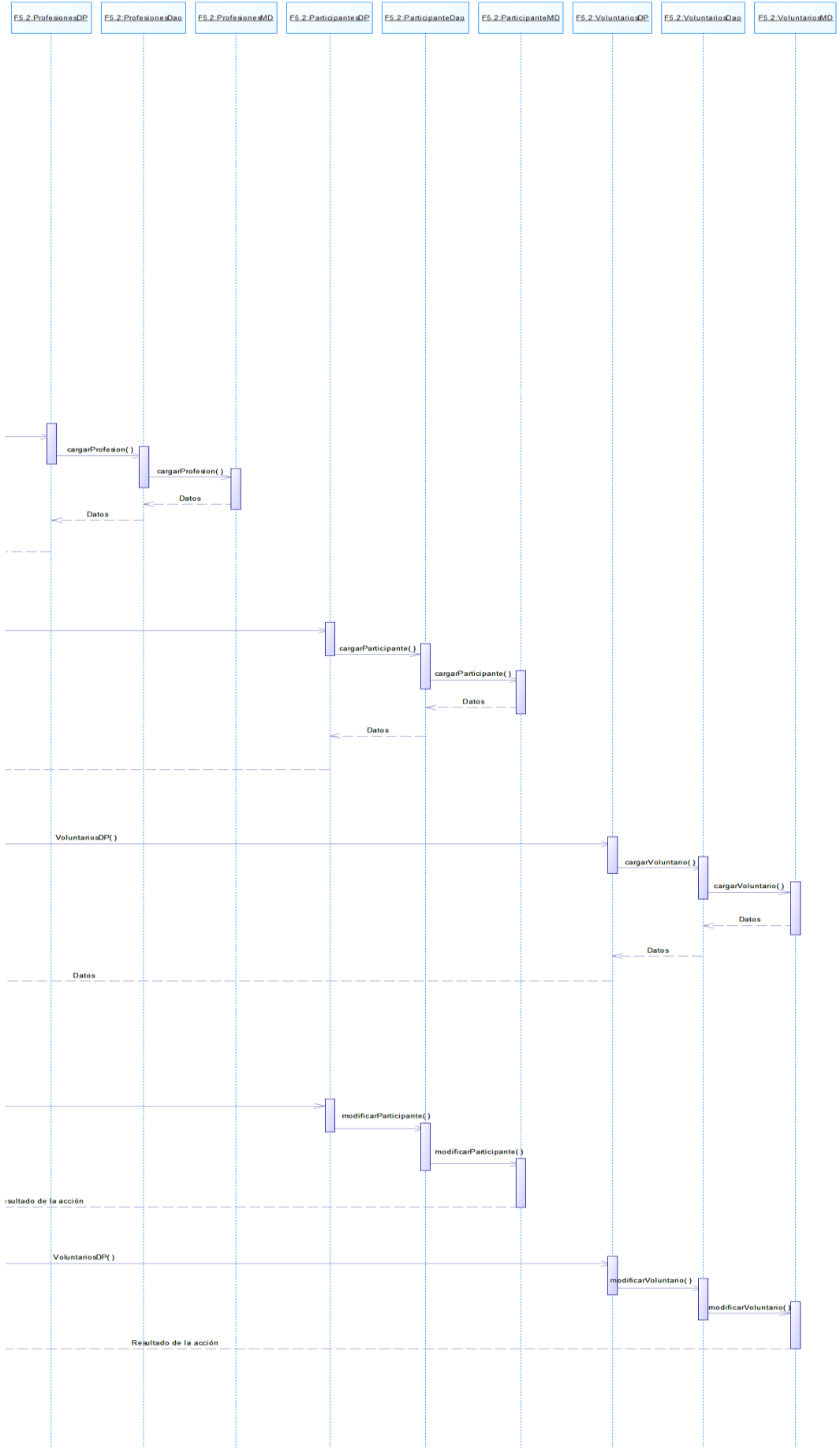
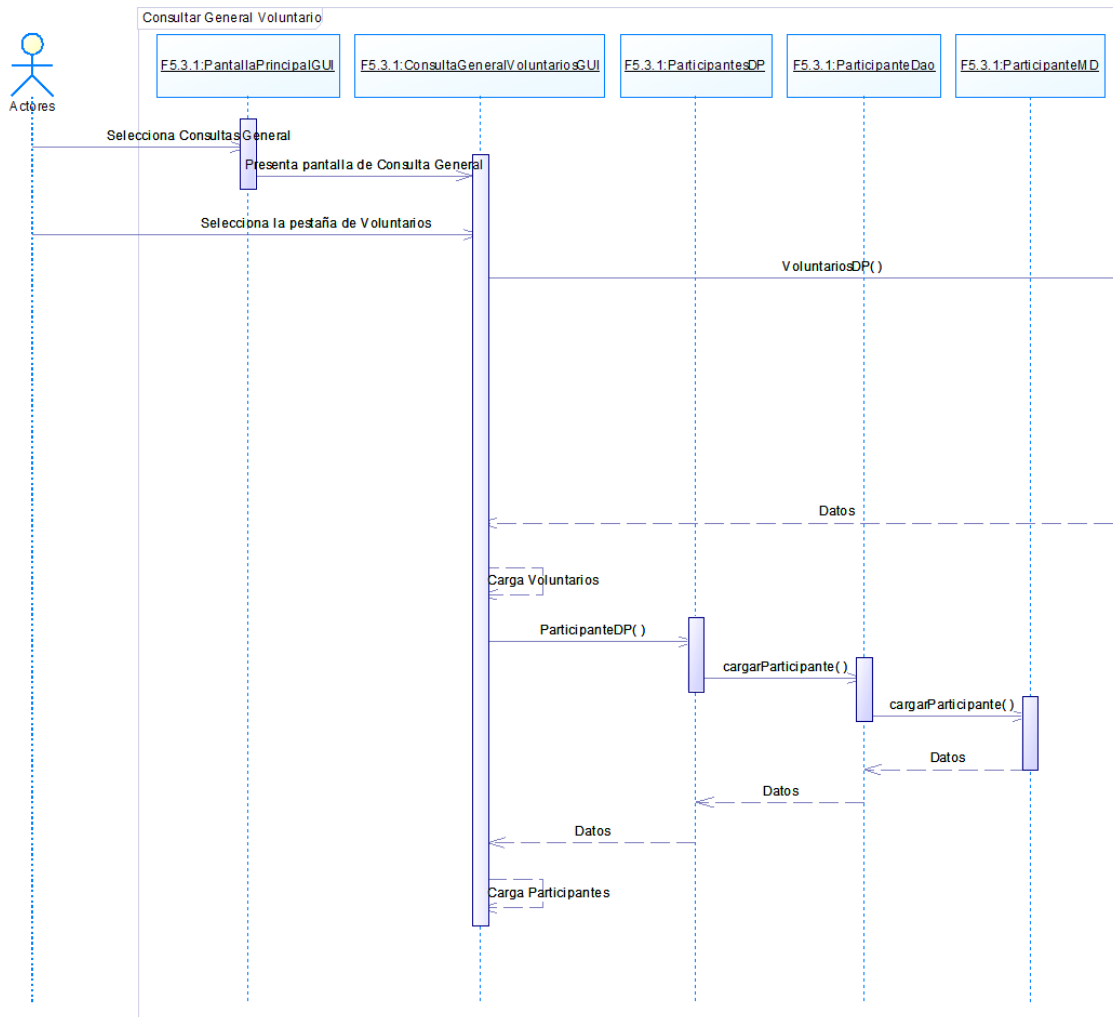


Figura 3.23: Diagrama de Secuencia Modificar voluntario

F5.3: Consultar voluntario

F5.3.1: Consulta general



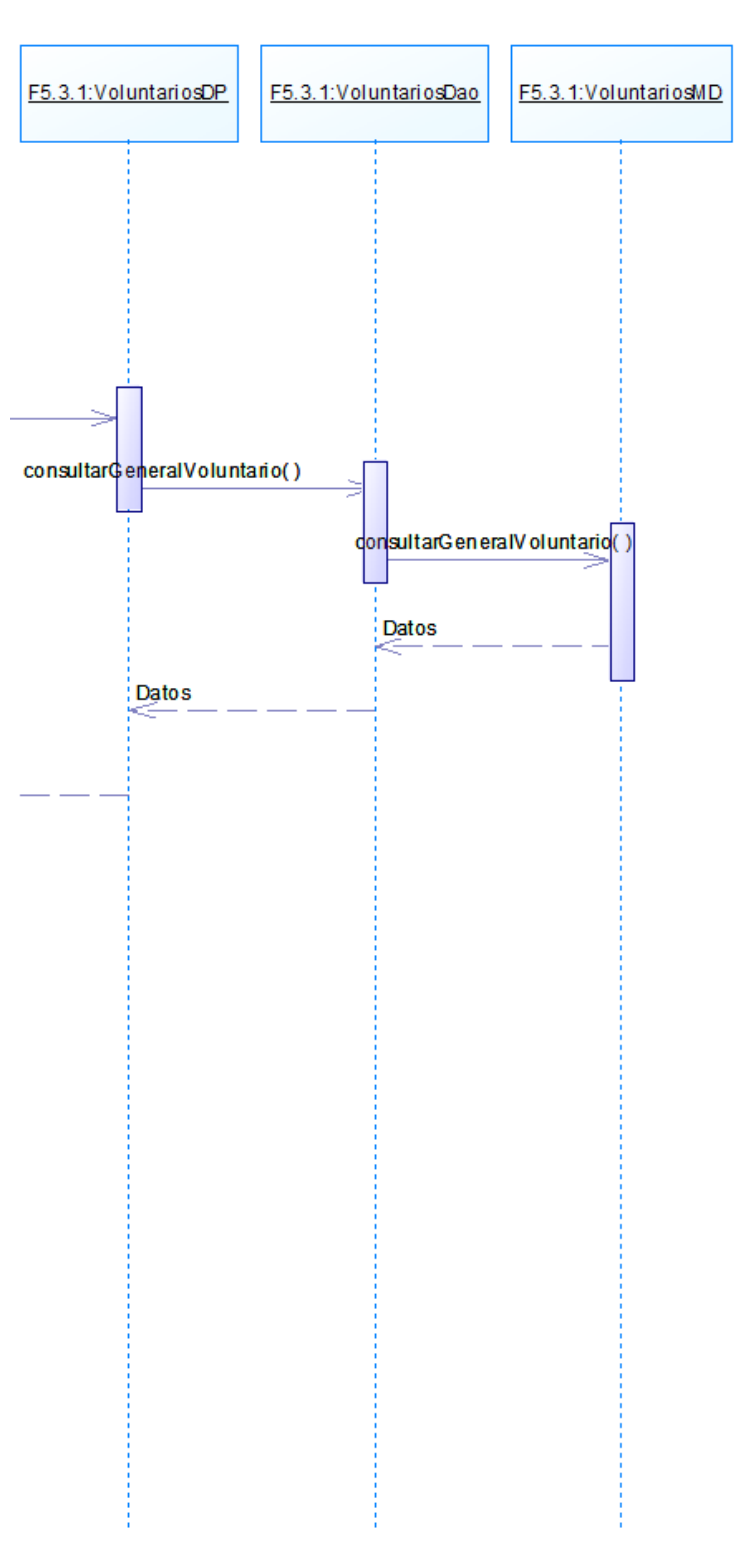
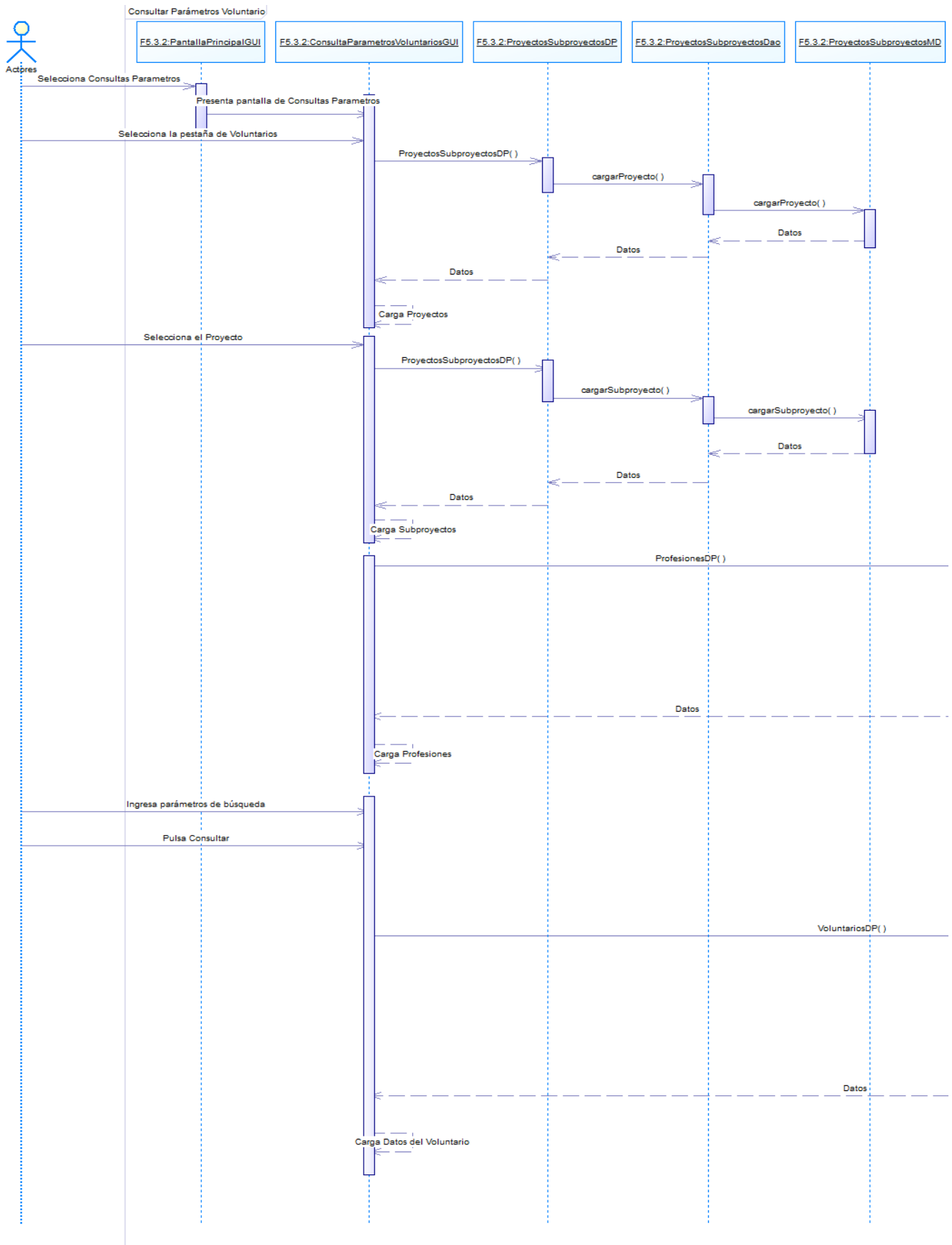


Figura 3.24: Diagrama de Secuencia Consulta general voluntario

F5.3.2: Consulta por parámetros



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

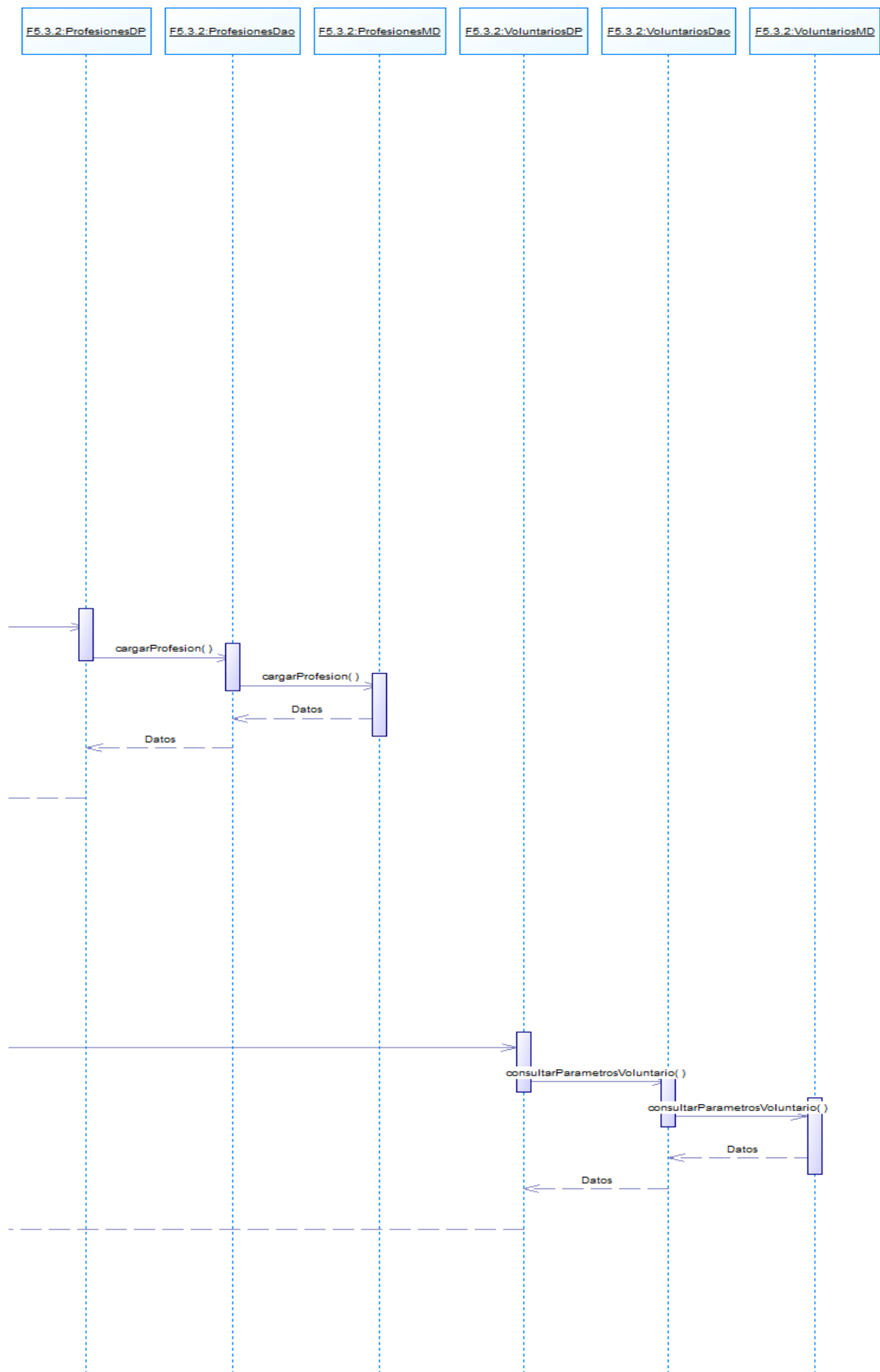
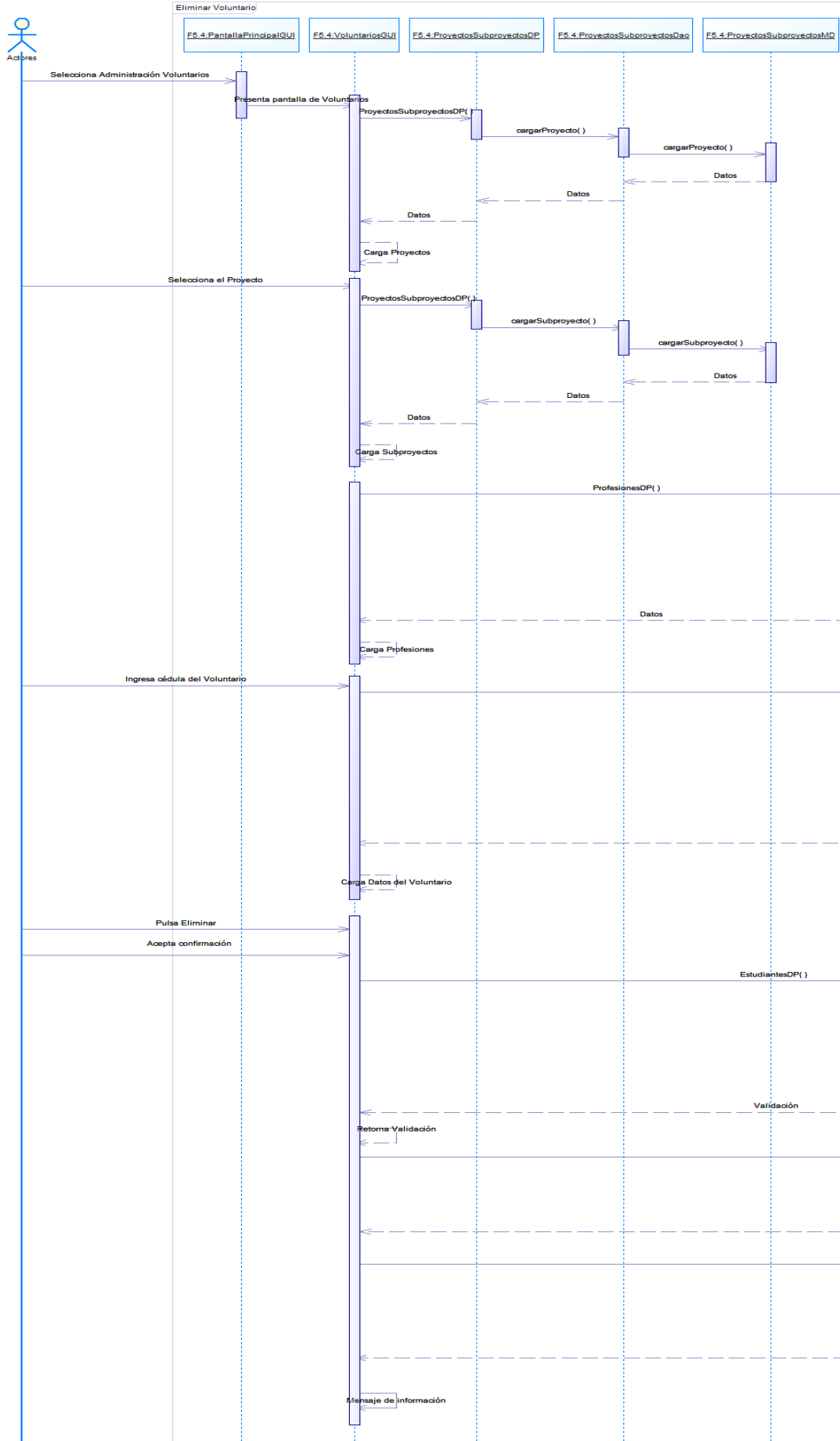
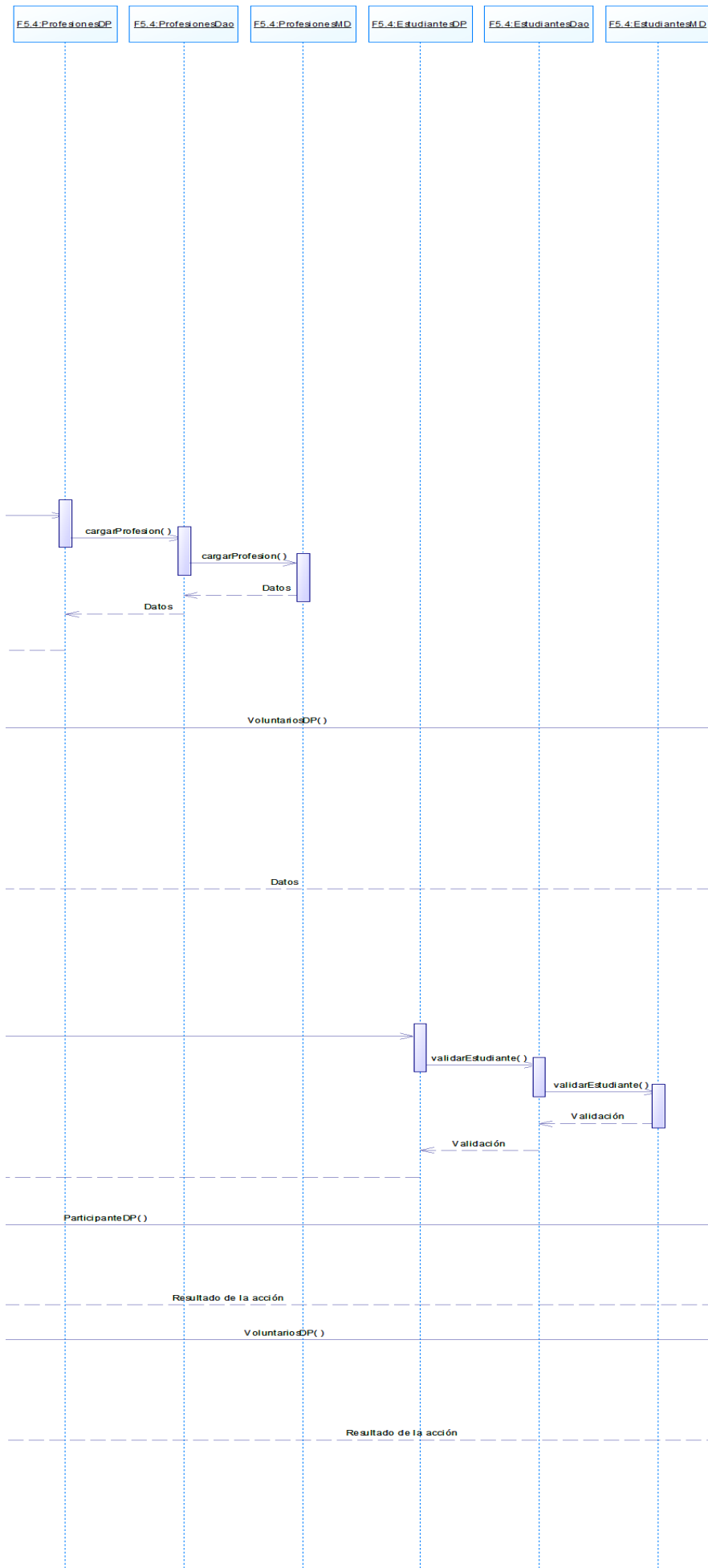


Figura 3.25: Diagrama de Secuencia Consulta por parámetros voluntario

F5.4: Eliminar voluntario



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

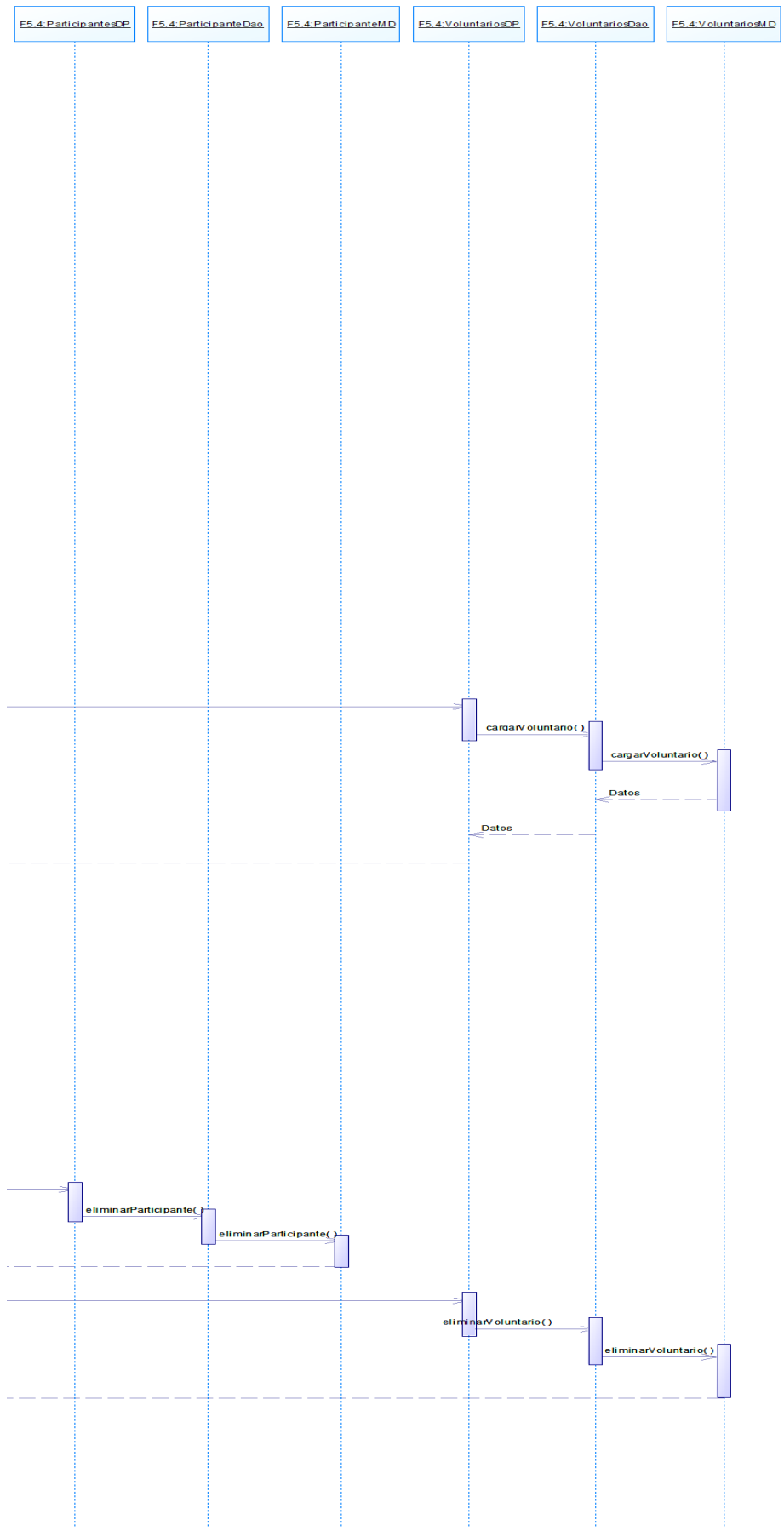
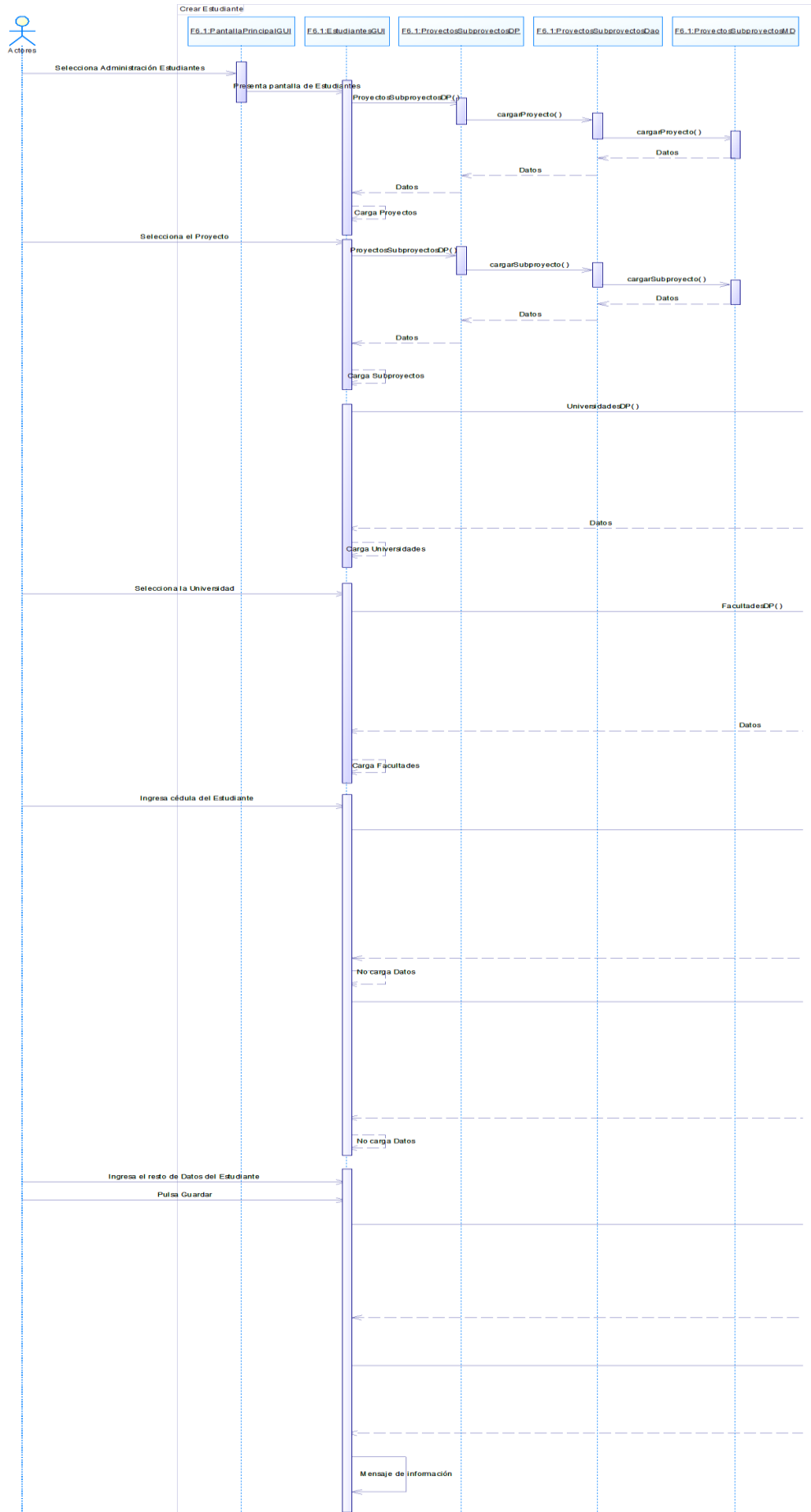


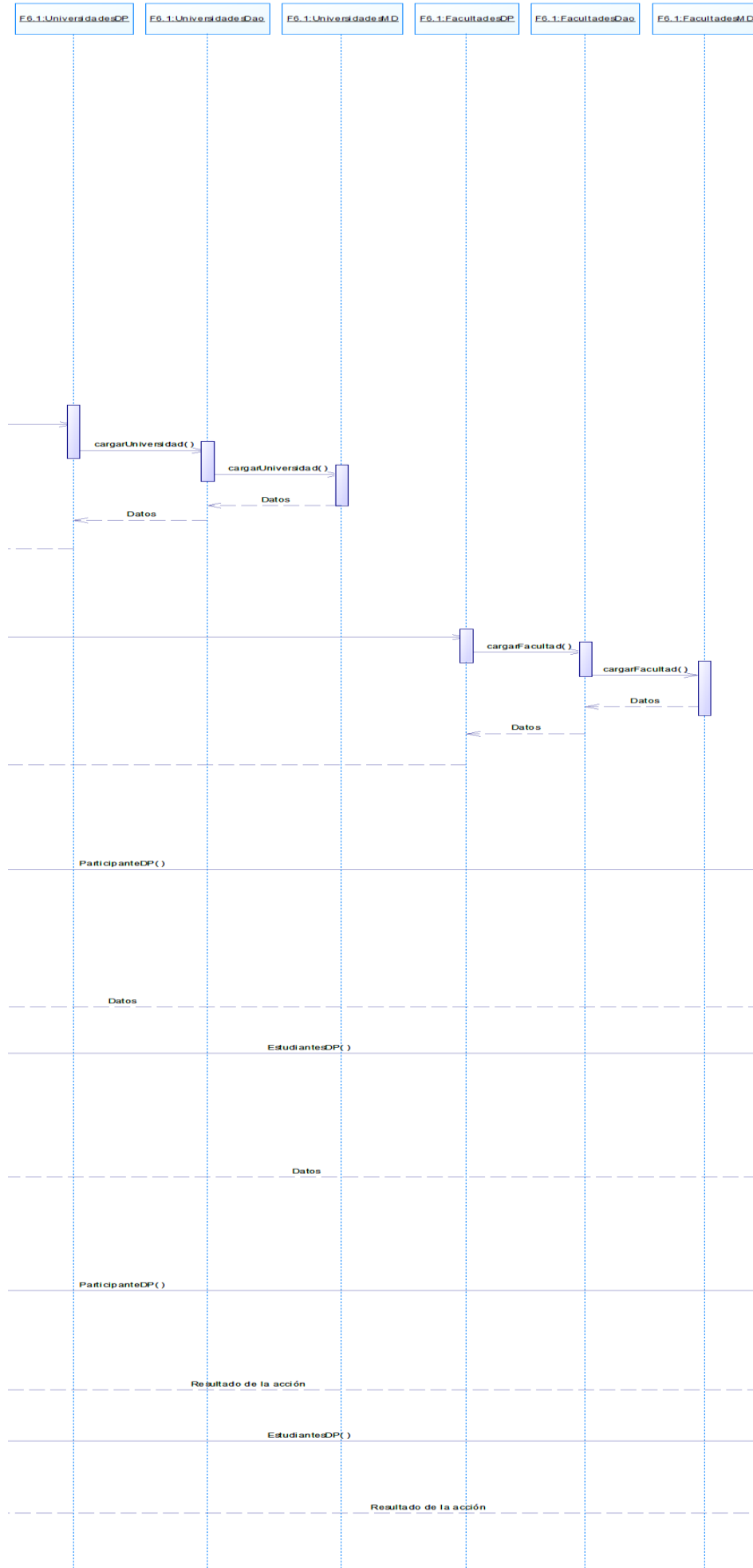
Figura 3.26: Diagrama de Secuencia Eliminar voluntario

F6: Administración de estudiantes

F6.1: Crear estudiante



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

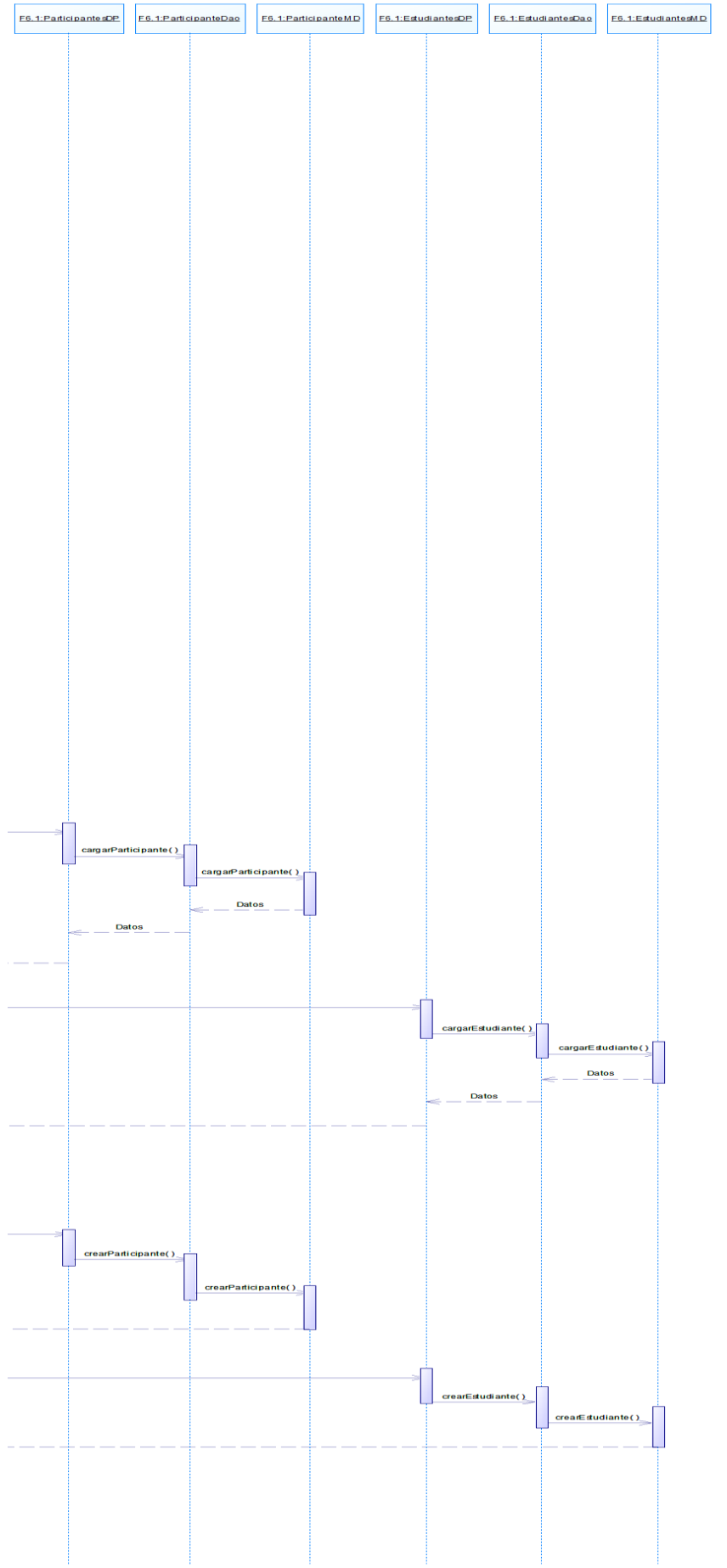
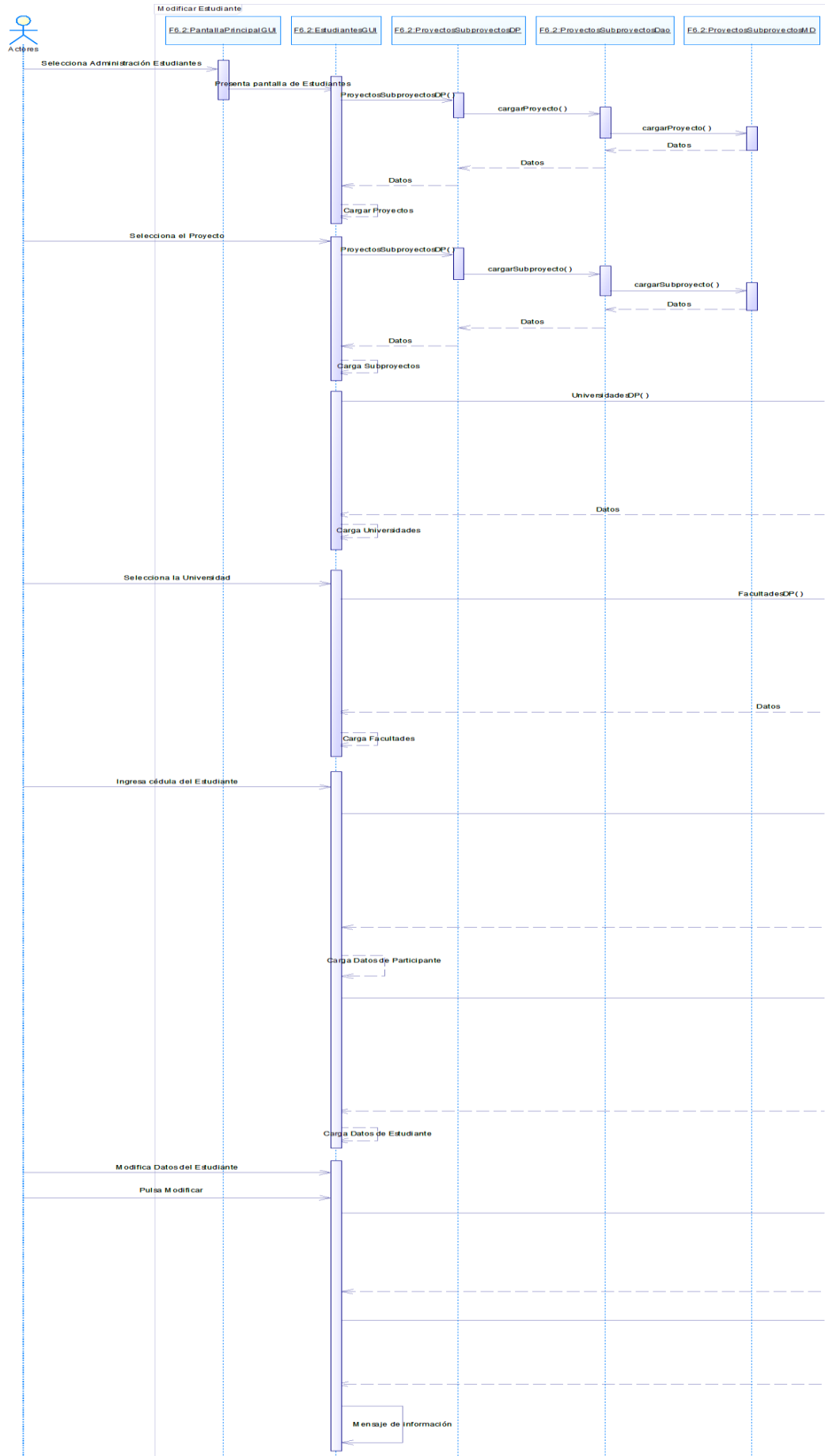
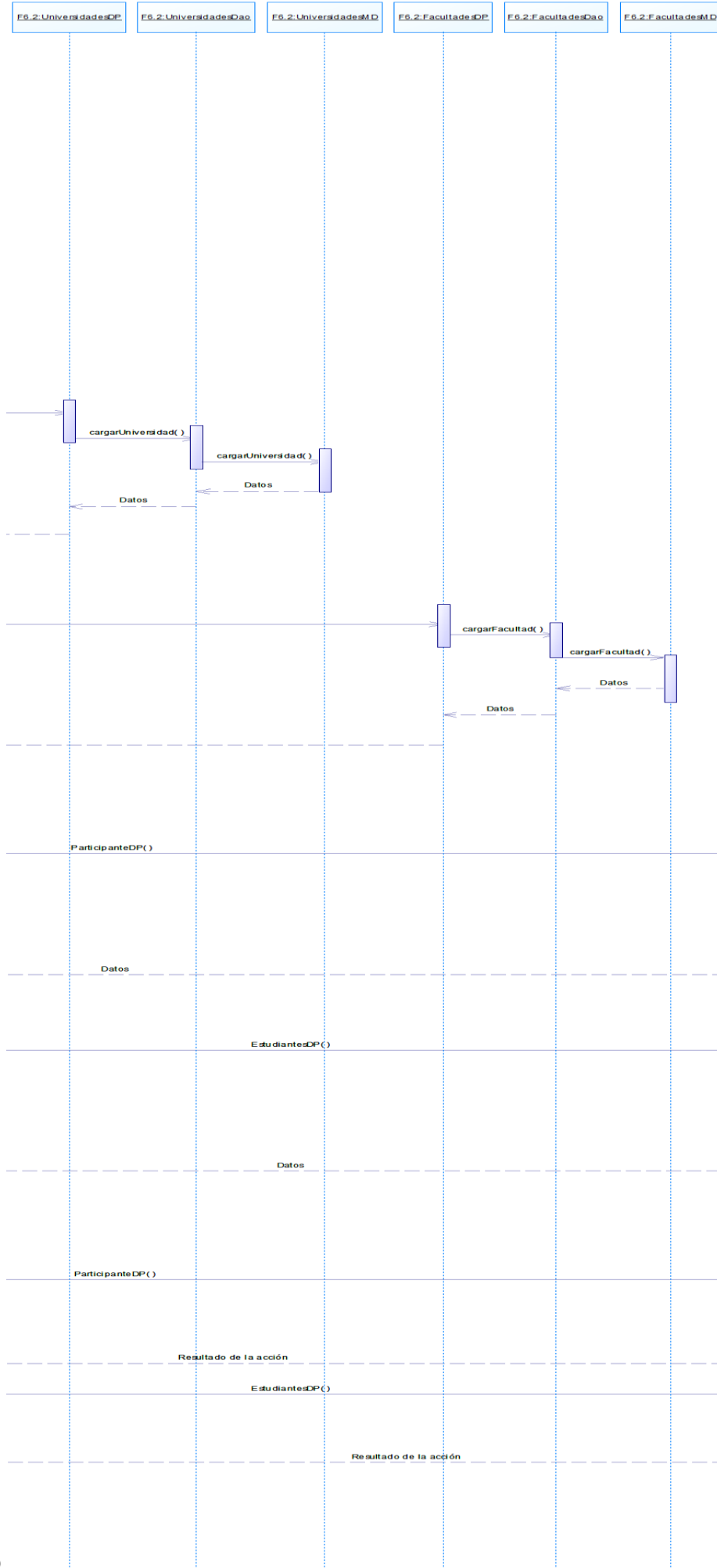


Figura 3.27: Diagrama de Secuencia Crear estudiante

F6.2: Modificar estudiante



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

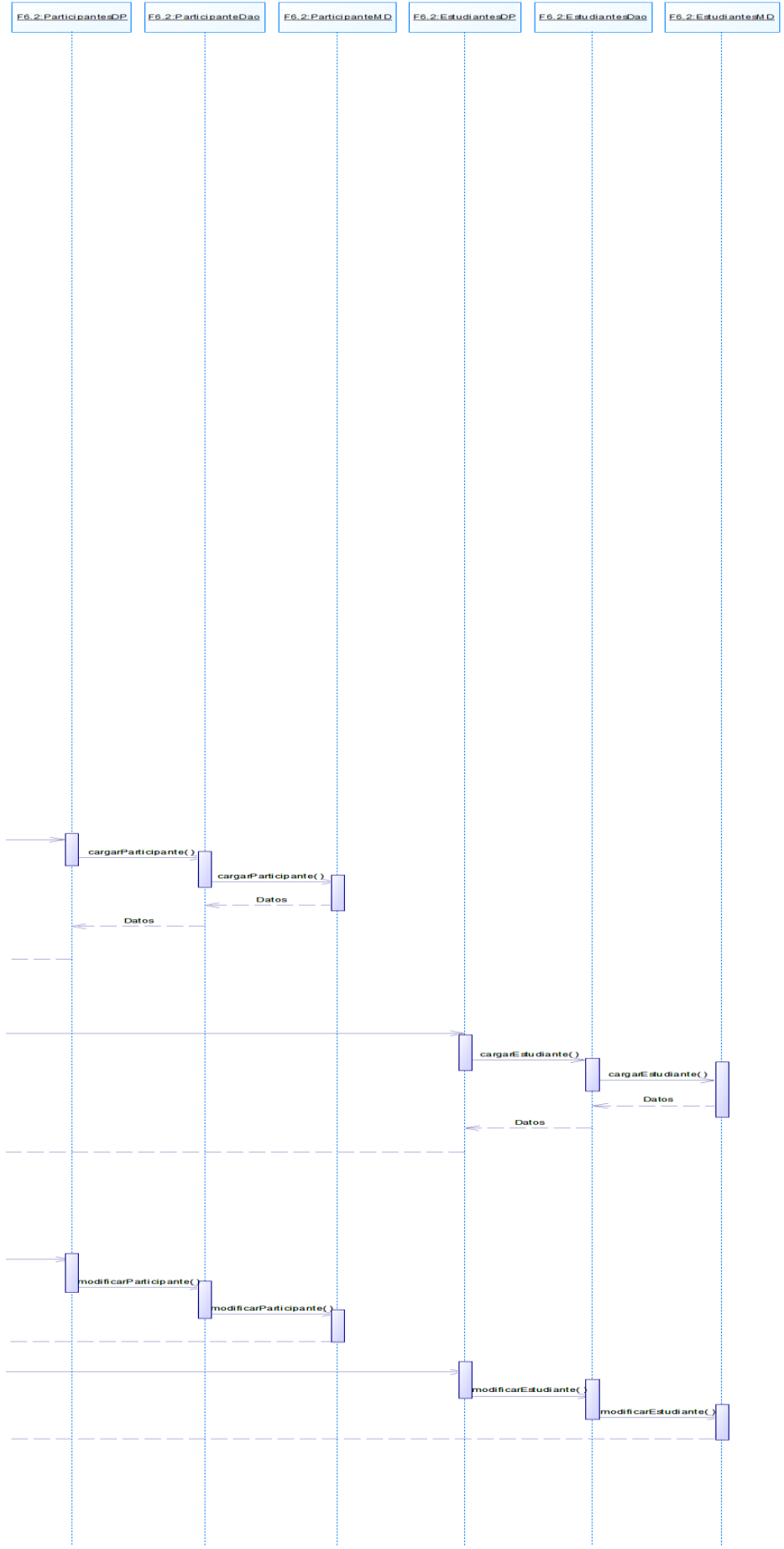
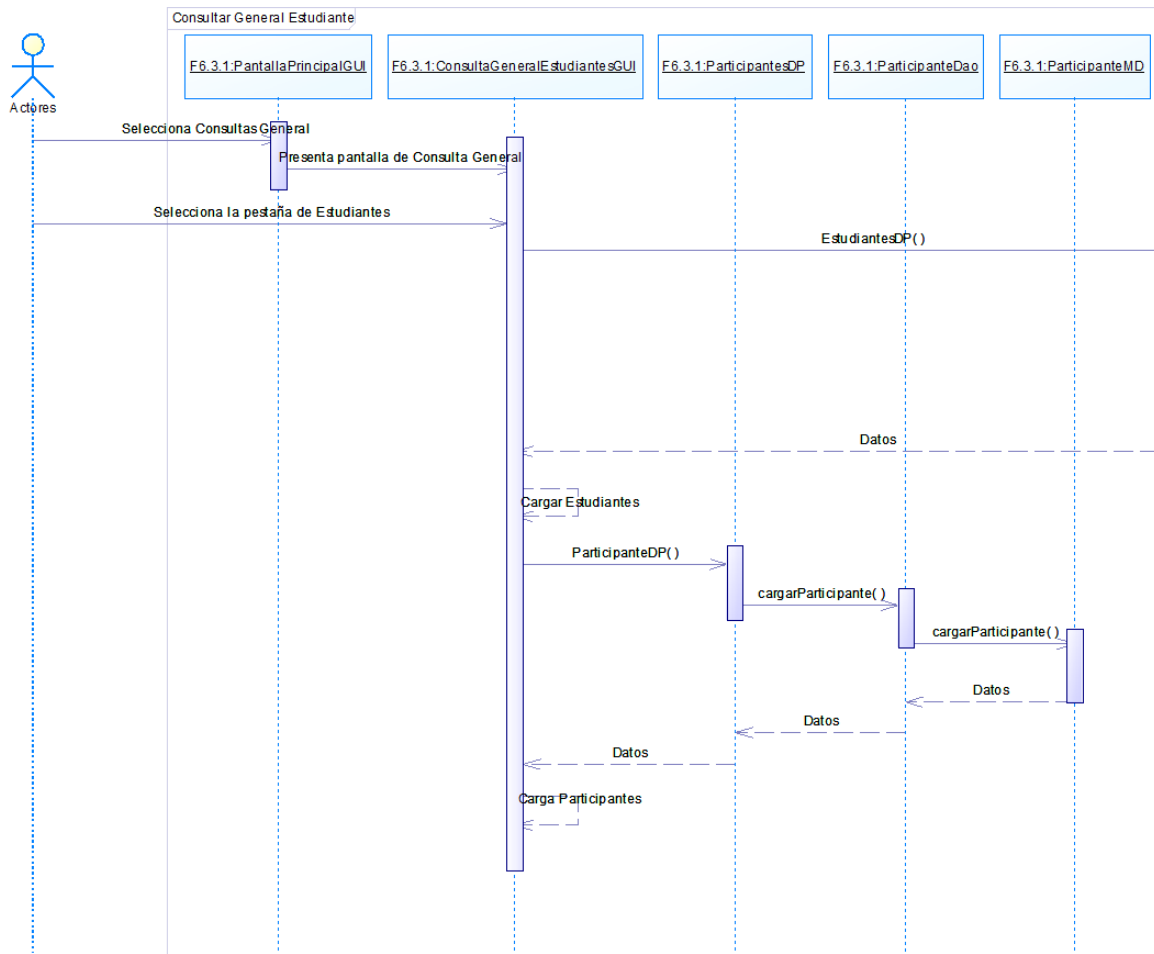


Figura 3.28: Diagrama de Secuencia Modificar estudiante

F6.3: Consultar estudiante

F6.3.1: Consulta general



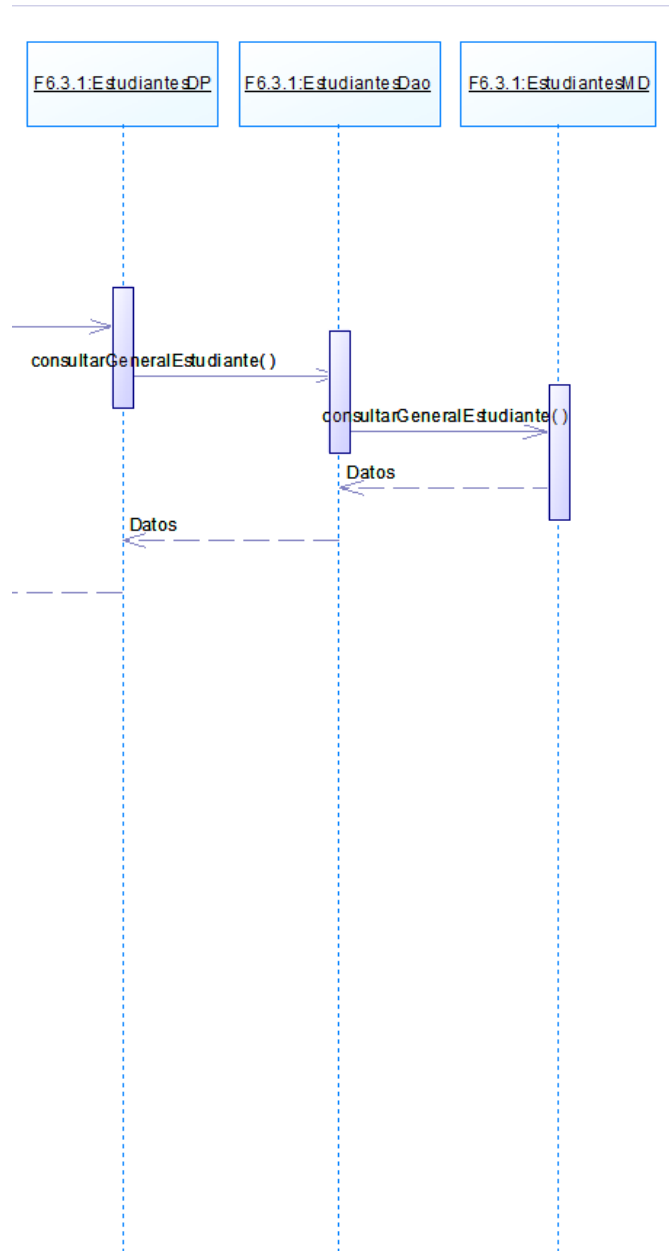
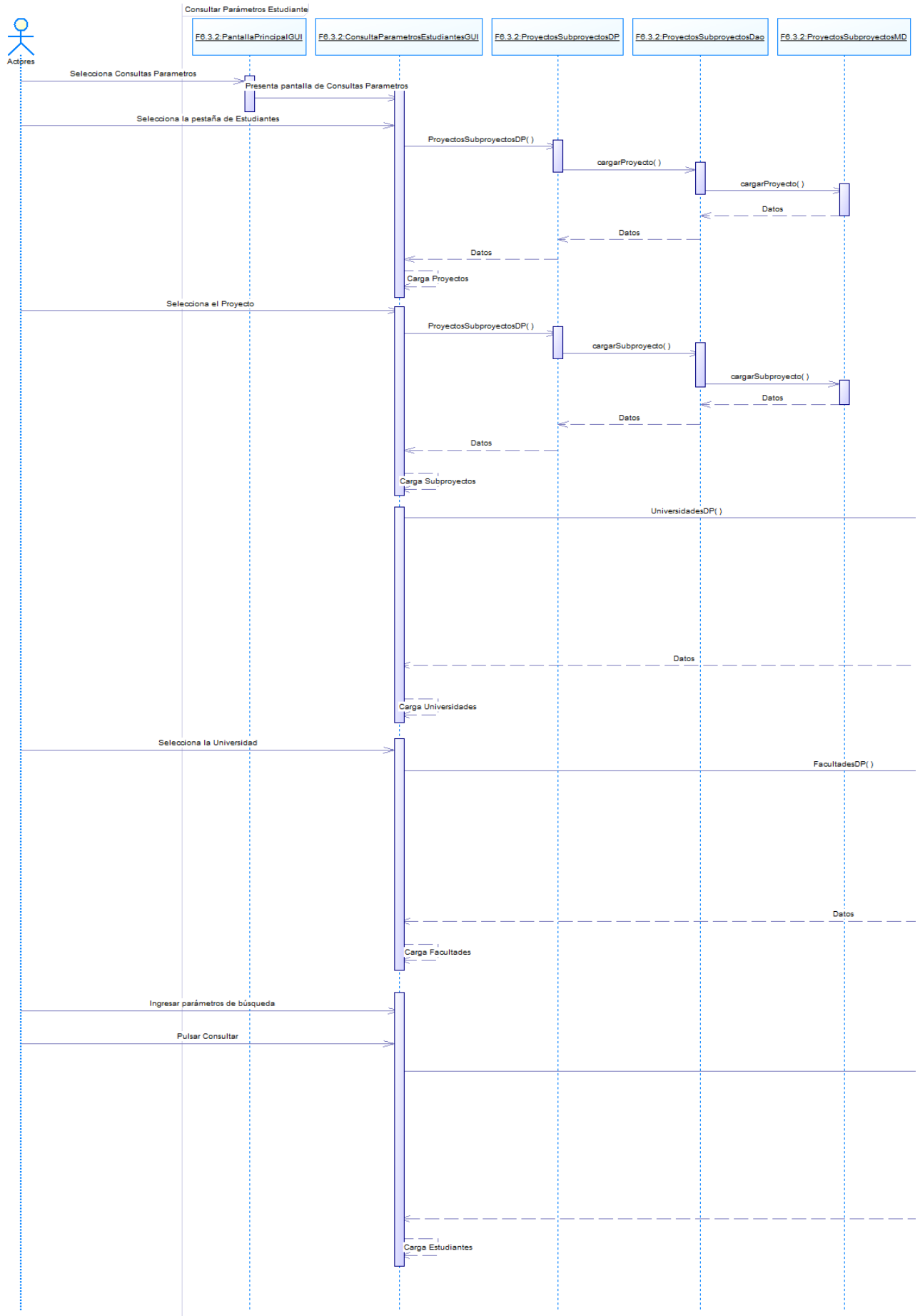


Figura 3.29: Diagrama de Secuencia Consulta general estudiante

F6.3.2: Consulta por parámetros



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

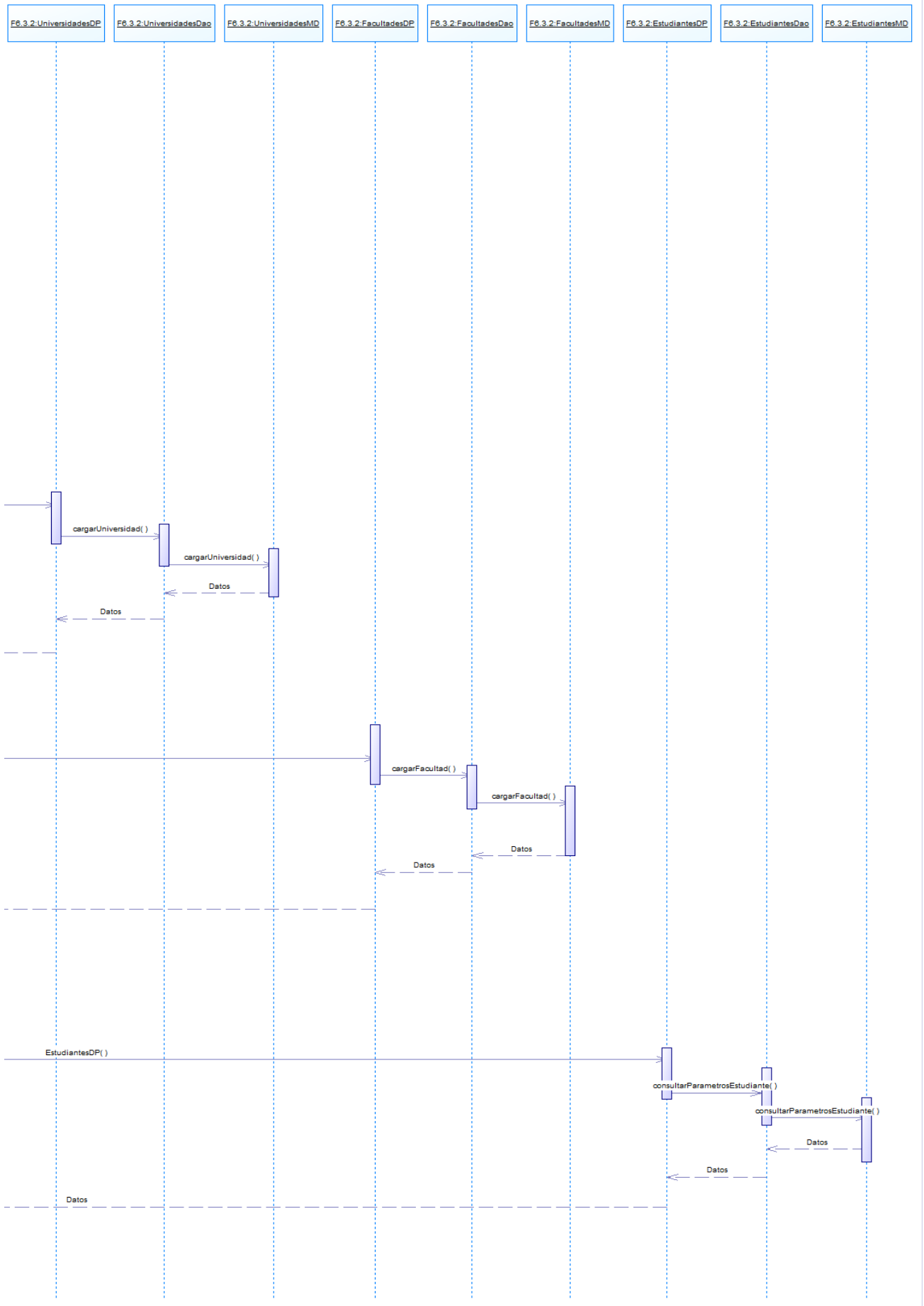
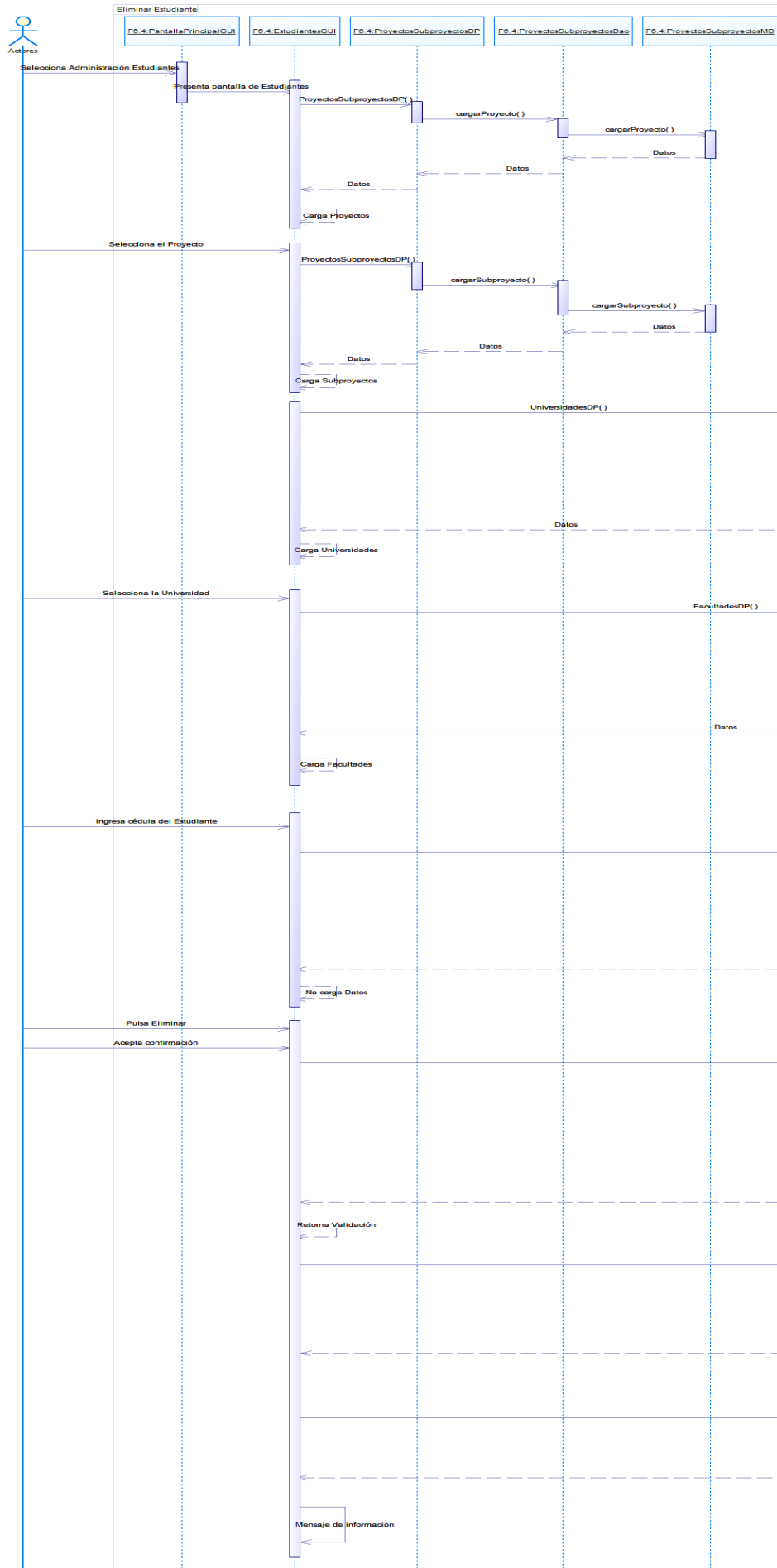
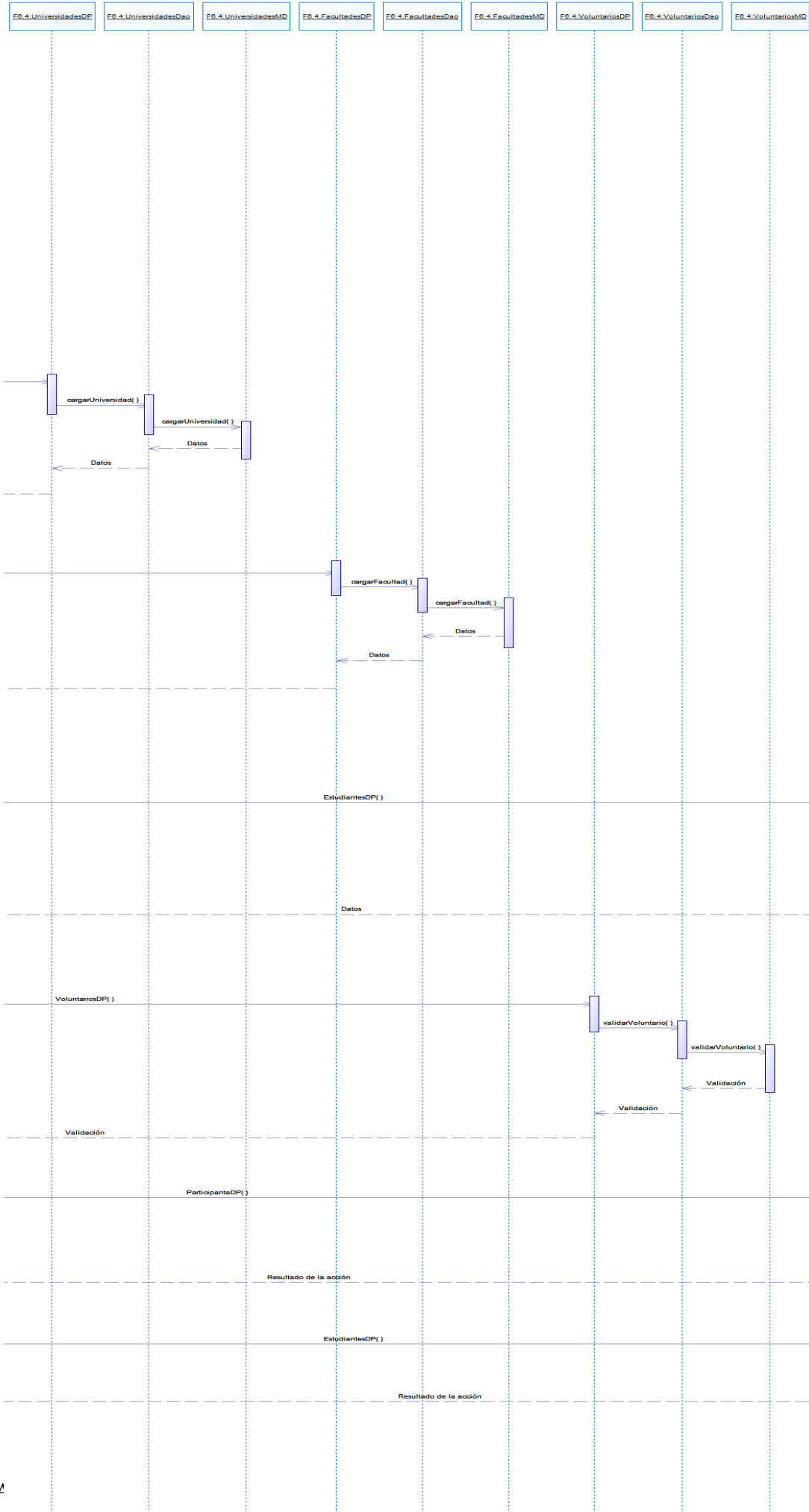


Figura 3.30: Diagrama de Secuencia Consulta por parámetros estudiante

F6.4: Eliminar estudiante



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

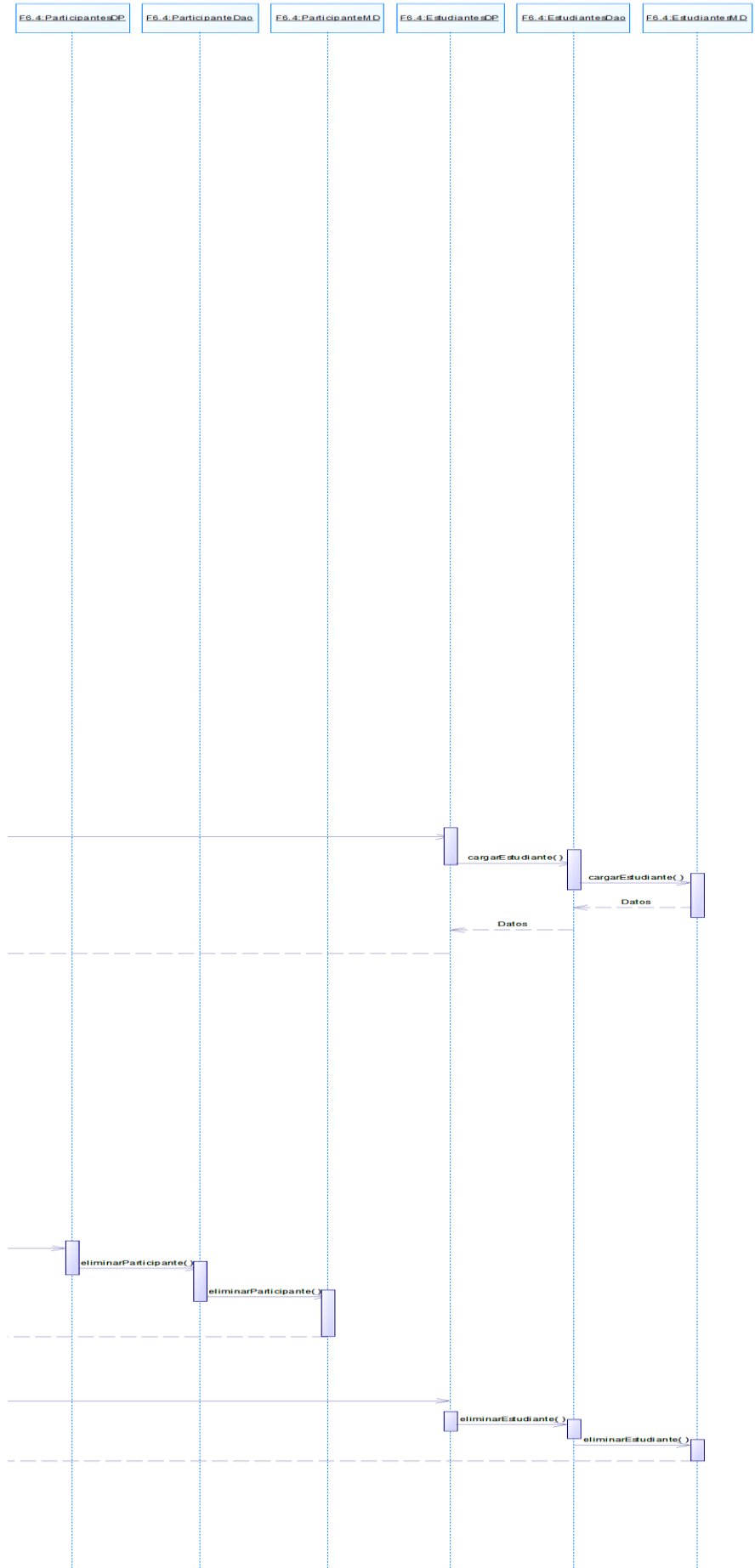


Figura 3.31: Diagrama de Secuencia Eliminar estudiante

F7: Administración de profesiones

F7.1: Crear profesión

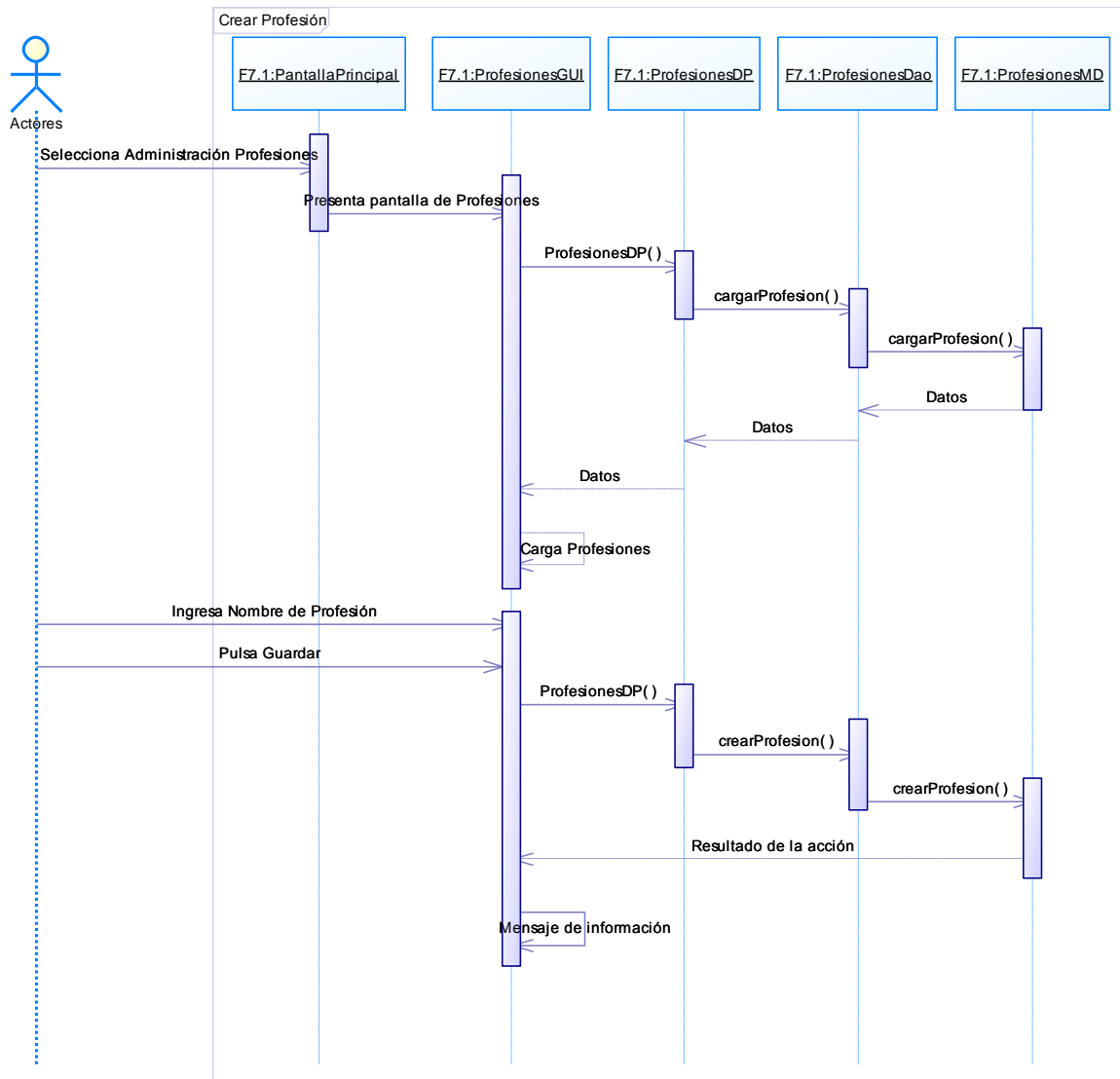


Figura 3.32: Diagrama de Secuencia Crear profesión

F7.2: Modificar profesión

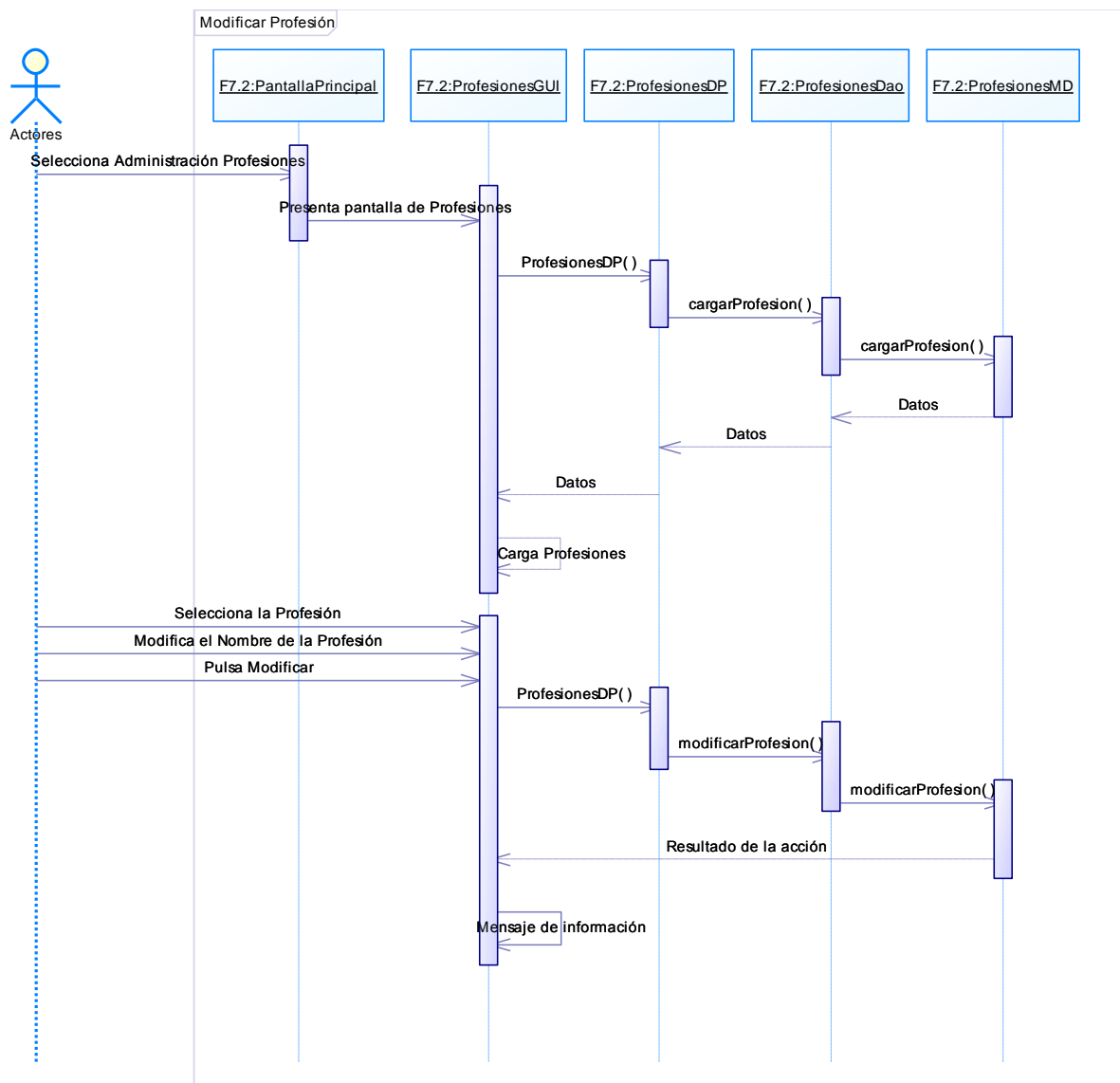


Figura 3.33: Diagrama de Secuencia Modificar profesión

F7.3: Consultar profesión

F7.3.1: Consulta general

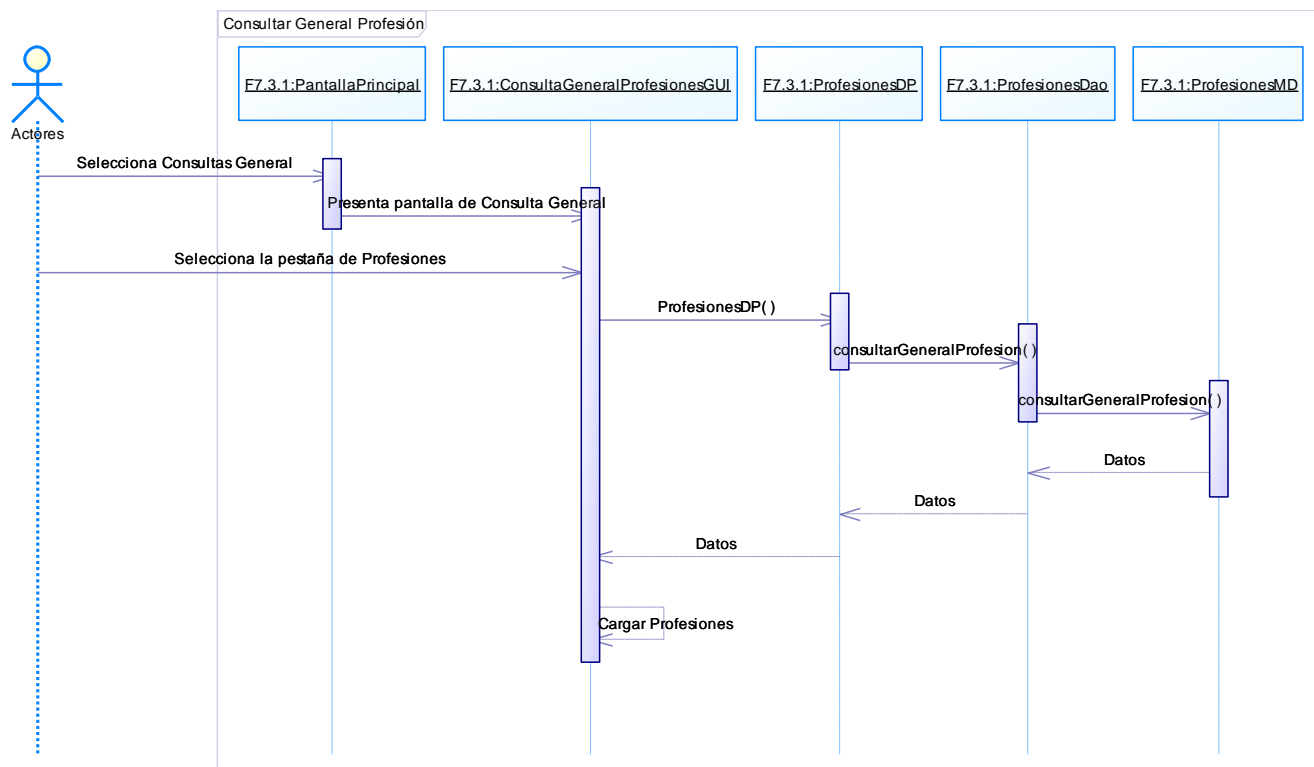


Figura 3.34: Diagrama de Secuencia Consulta general profesión

F7.4: Eliminar profesión

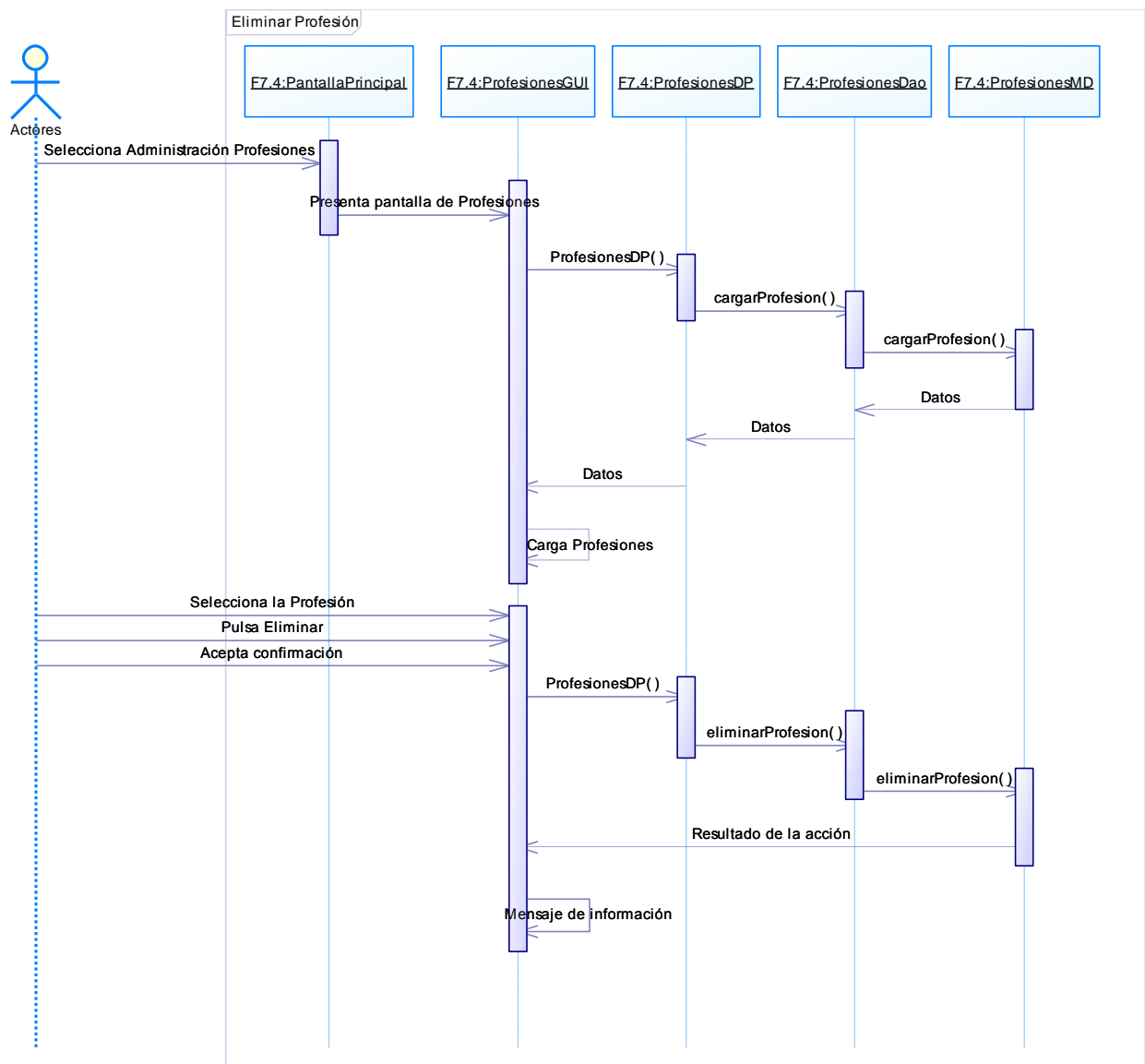


Figura 3.35: Diagrama de Secuencia Eliminar profesión

F8: Administración de universidades

F8.1: Crear universidad

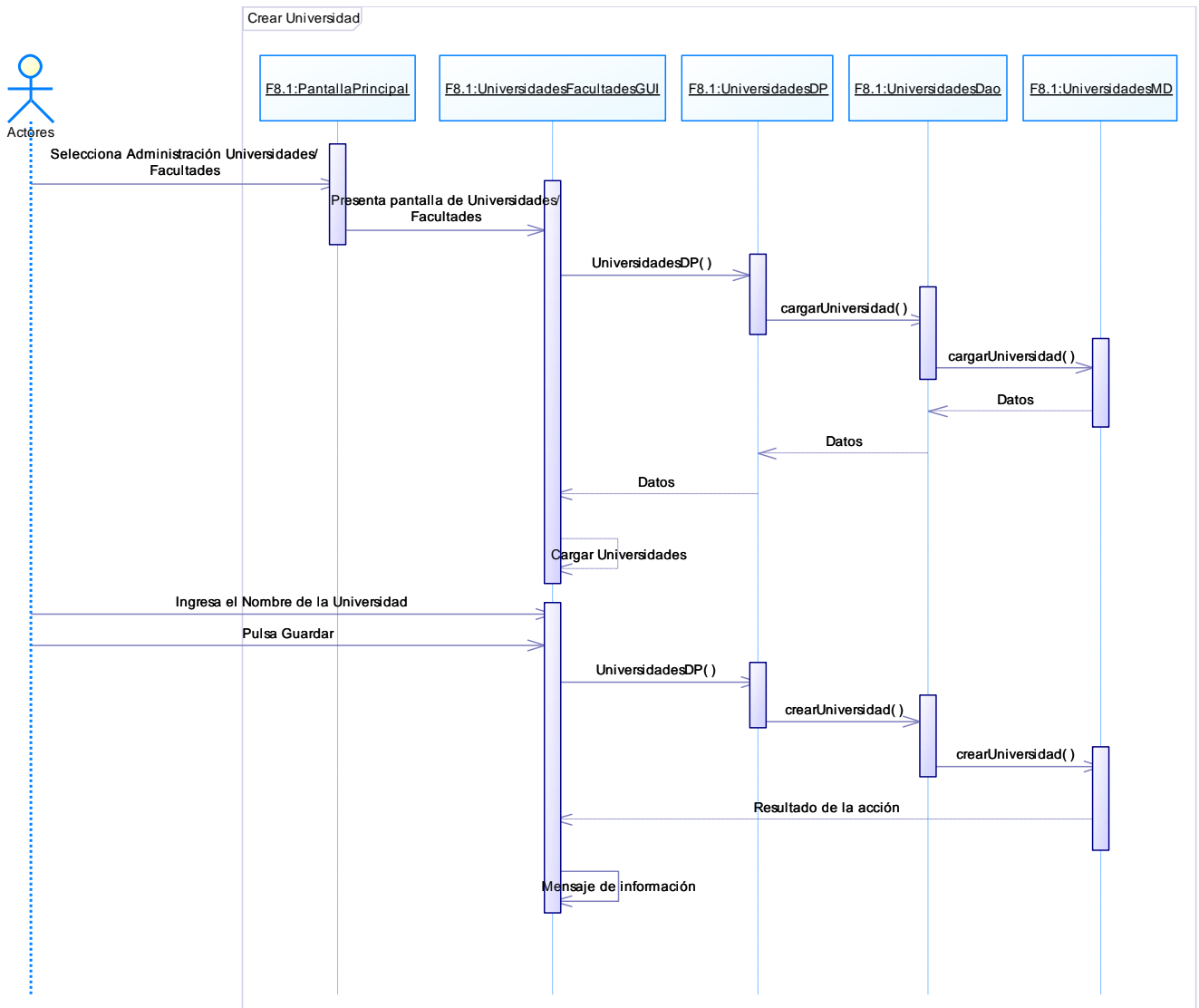


Figura 3.36: Diagrama de Secuencia Crear universidad

F8.2: Modificar universidad

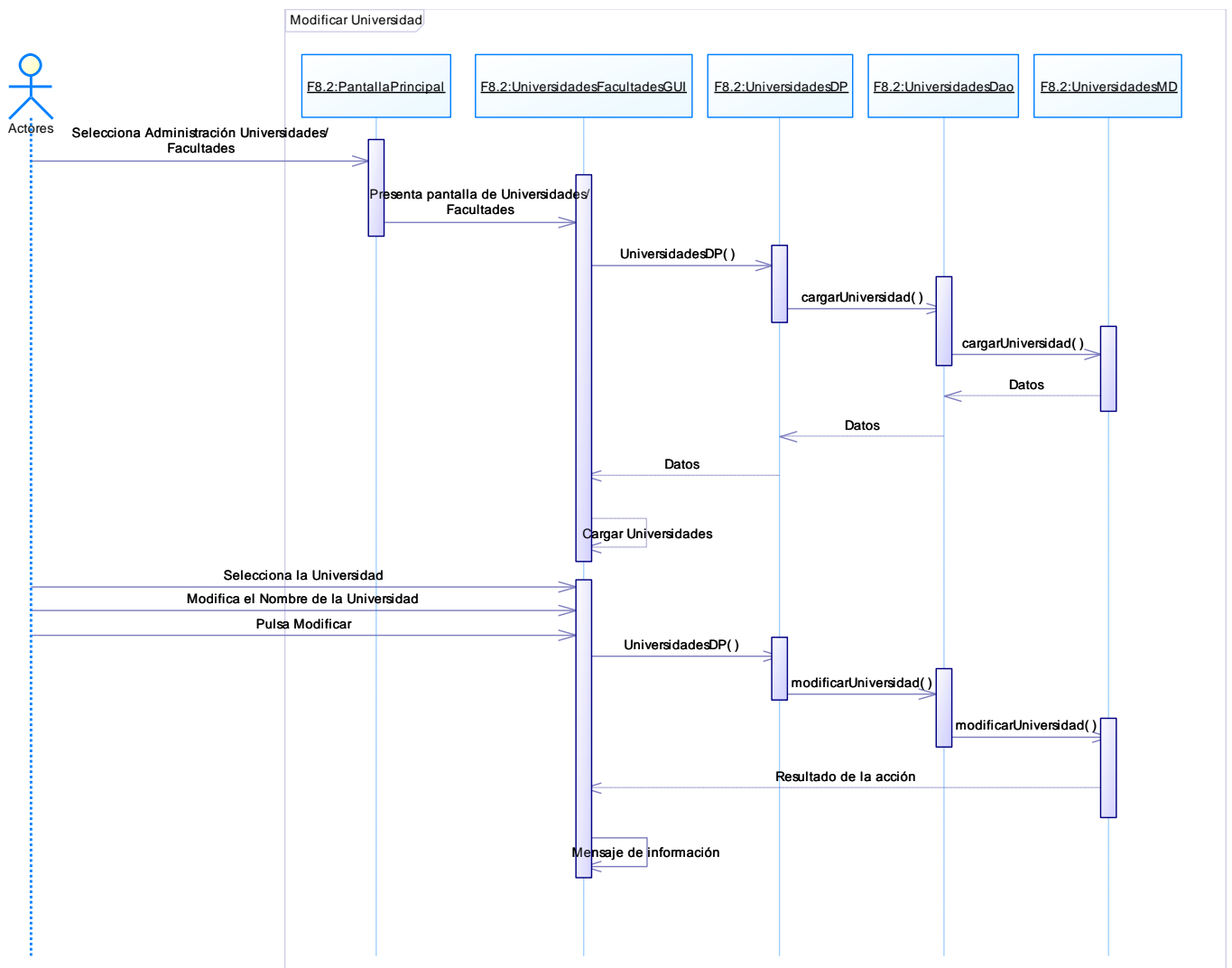


Figura 3.37: Diagrama de Secuencia Modificar universidad

F8.3: Consultar universidad

F8.3.1: Consulta general

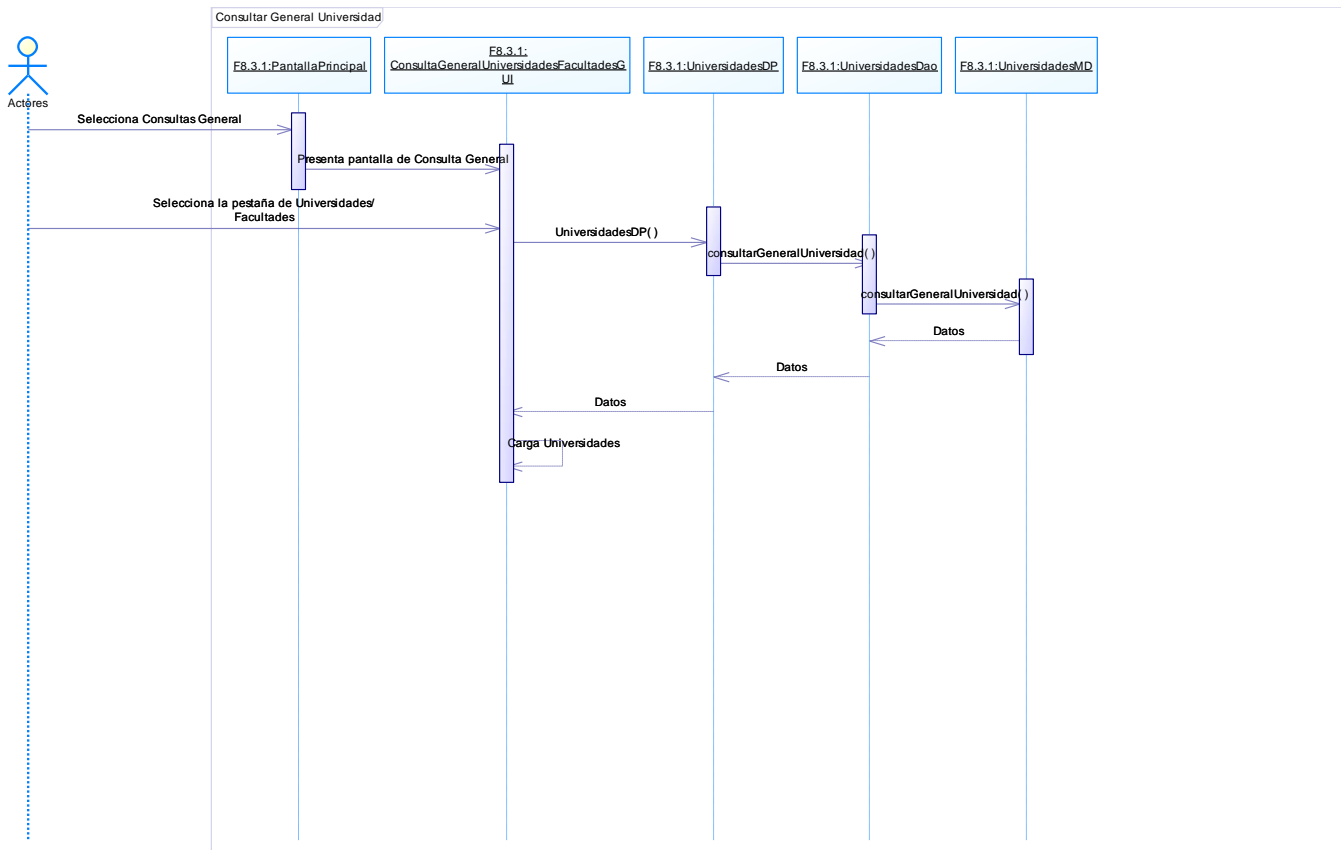


Figura 3.38: Diagrama de Secuencia Consulta general universidad

F8.4: Eliminar universidad

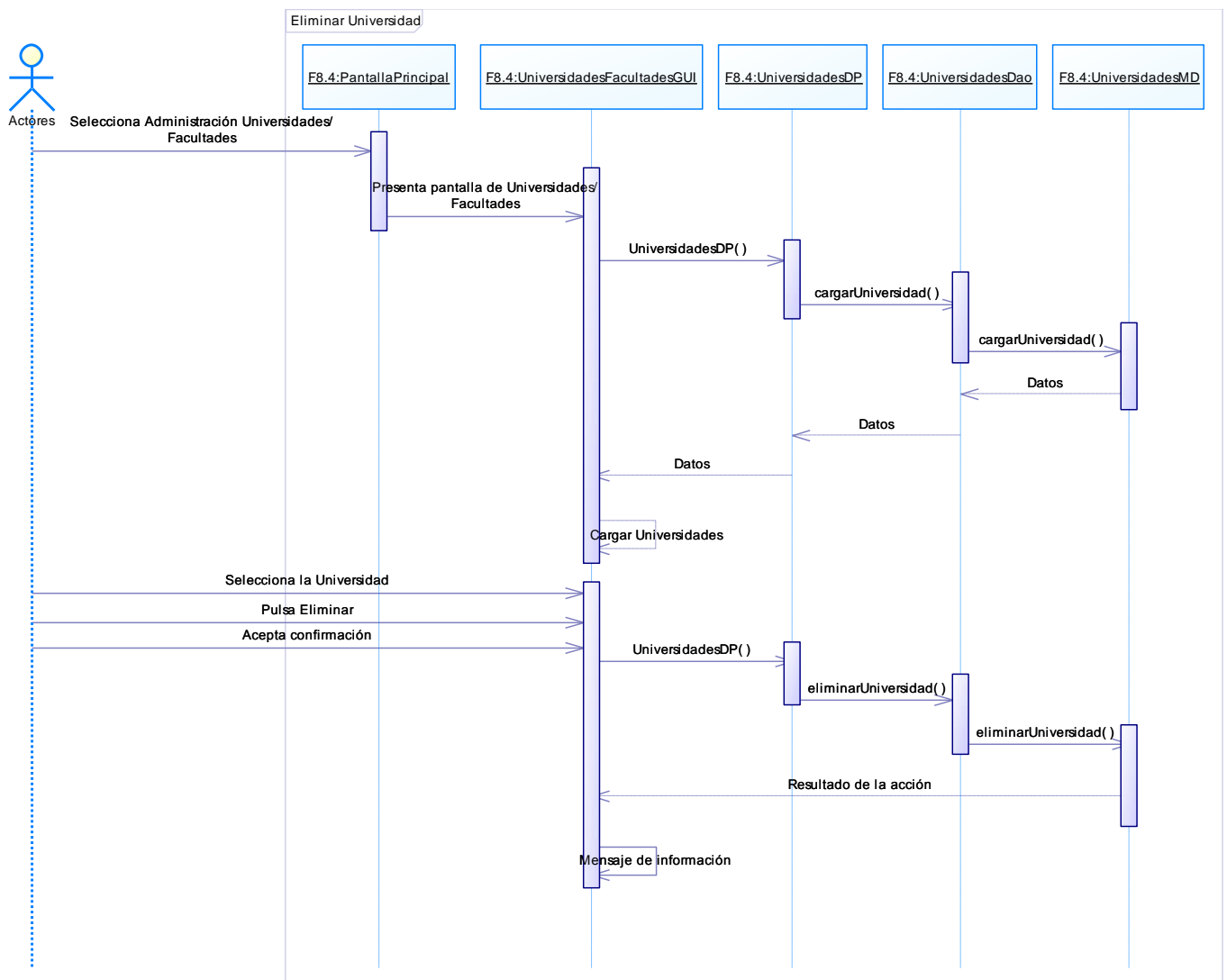


Figura 3.39: Diagrama de Secuencia Eliminar universidad

F9: Administración de facultades

F9.1: Crear facultad

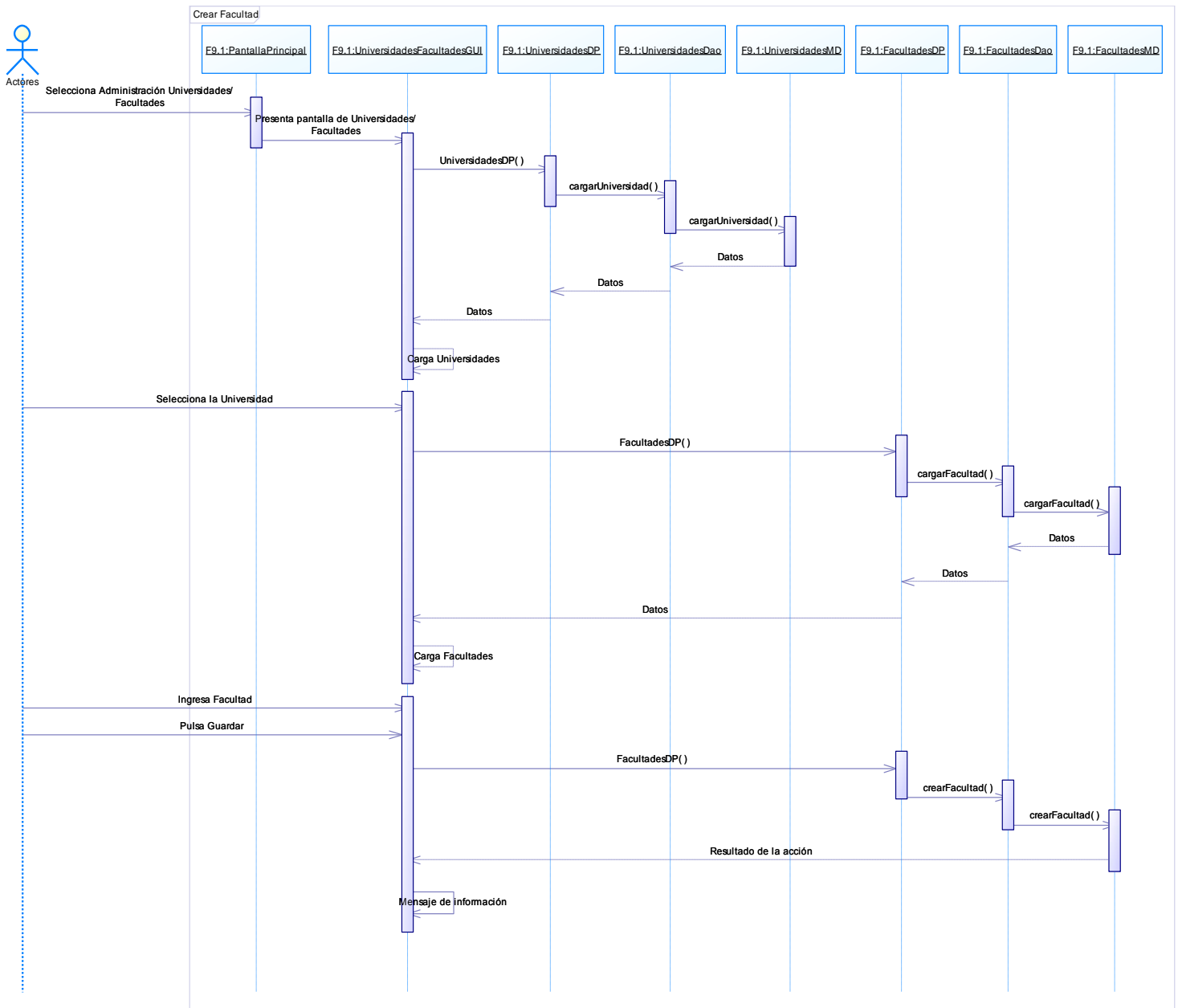


Figura 3.40: Diagrama de Secuencia Crear facultad

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

F9.2: Modificar facultad

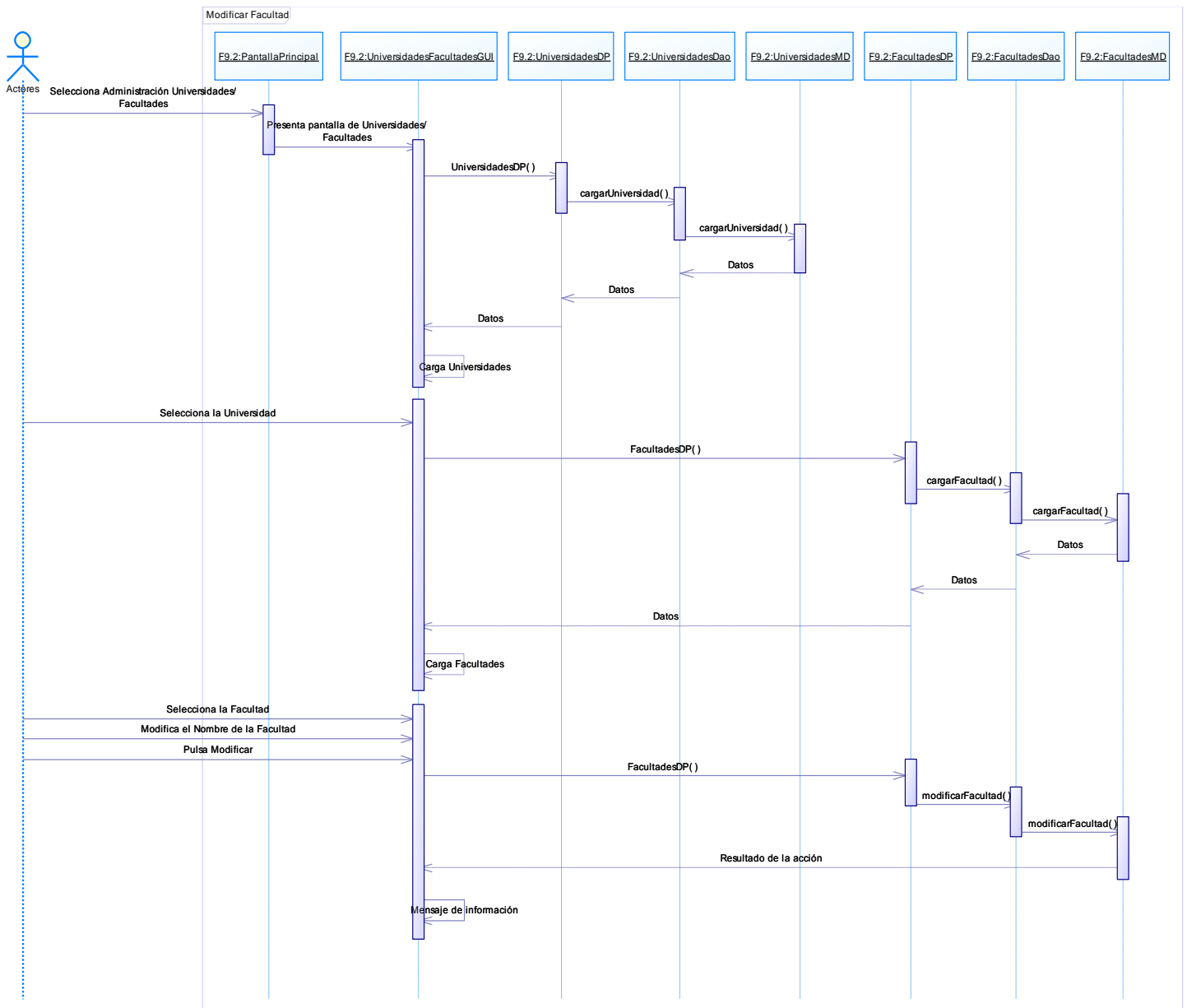


Figura 3.41: Diagrama de Secuencia Modificar facultad

F9.3: Consultar facultad

F9.3.1: Consulta general

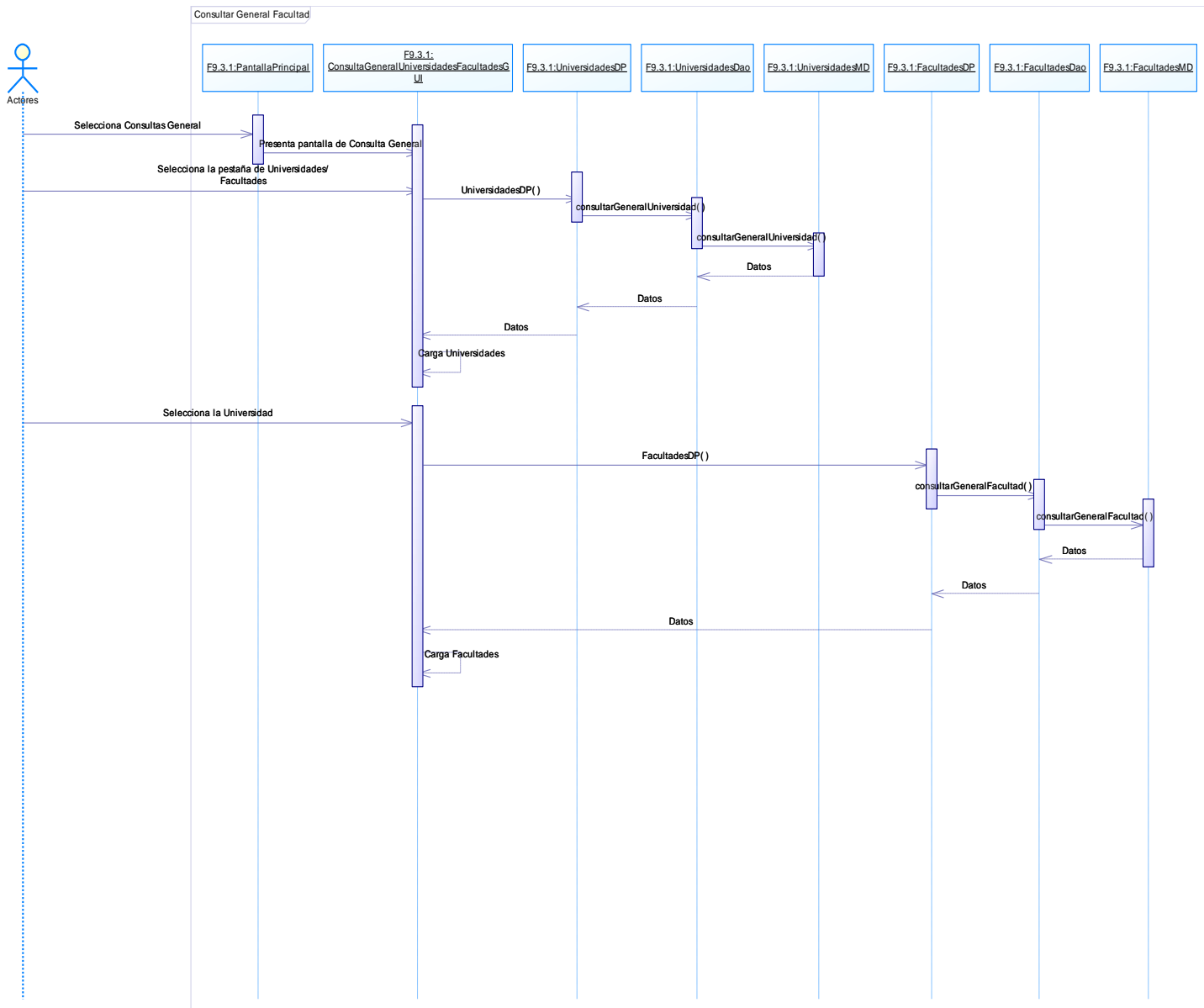


Figura 3.42: Diagrama de Secuencia Consulta general facultad

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

F9.4: Eliminar facultad

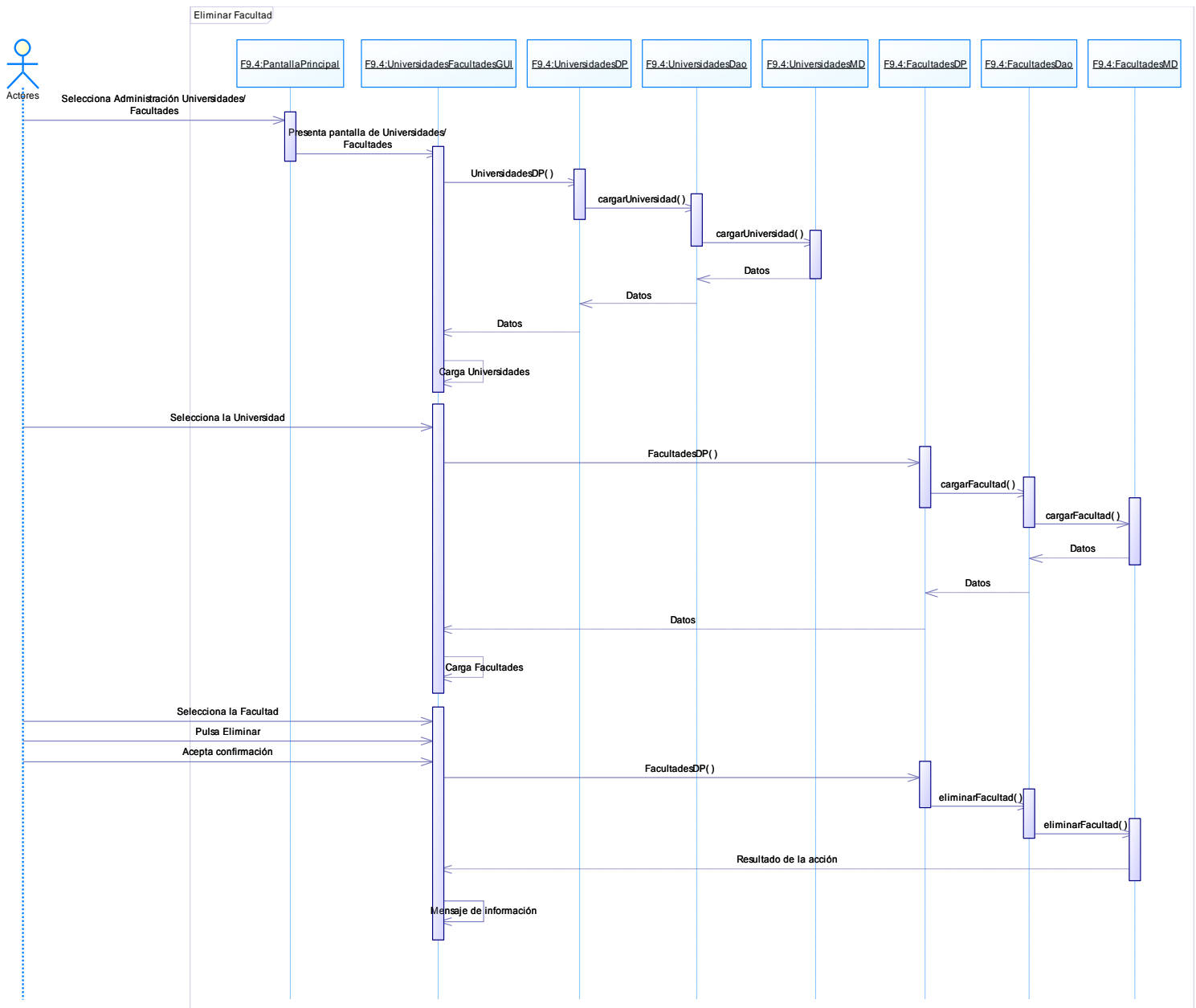
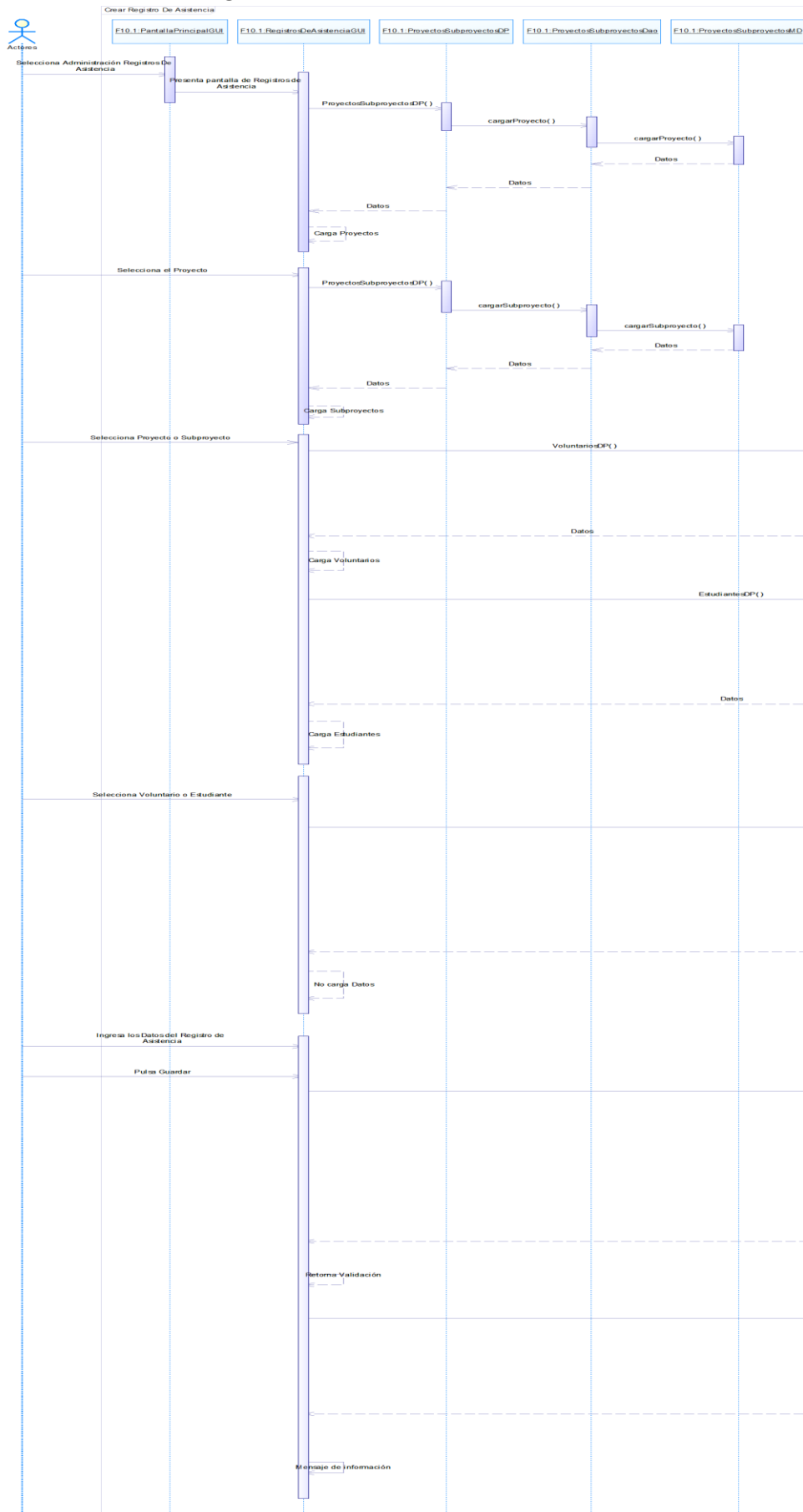


Figura 3.43: Diagrama de Secuencia Eliminar facultad

F10: Administración de registros de asistencia

F10.1: Crear registro de asistencia



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

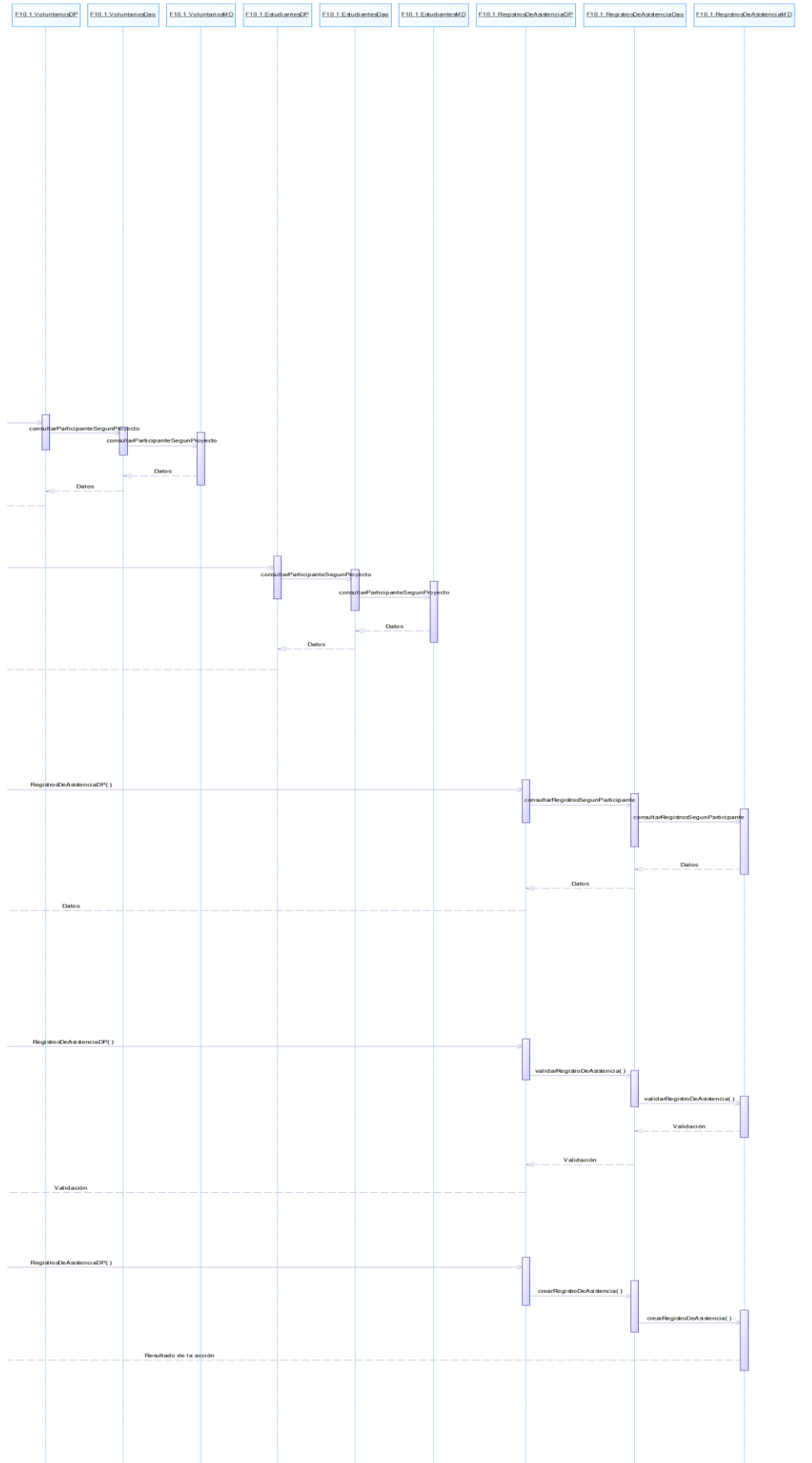
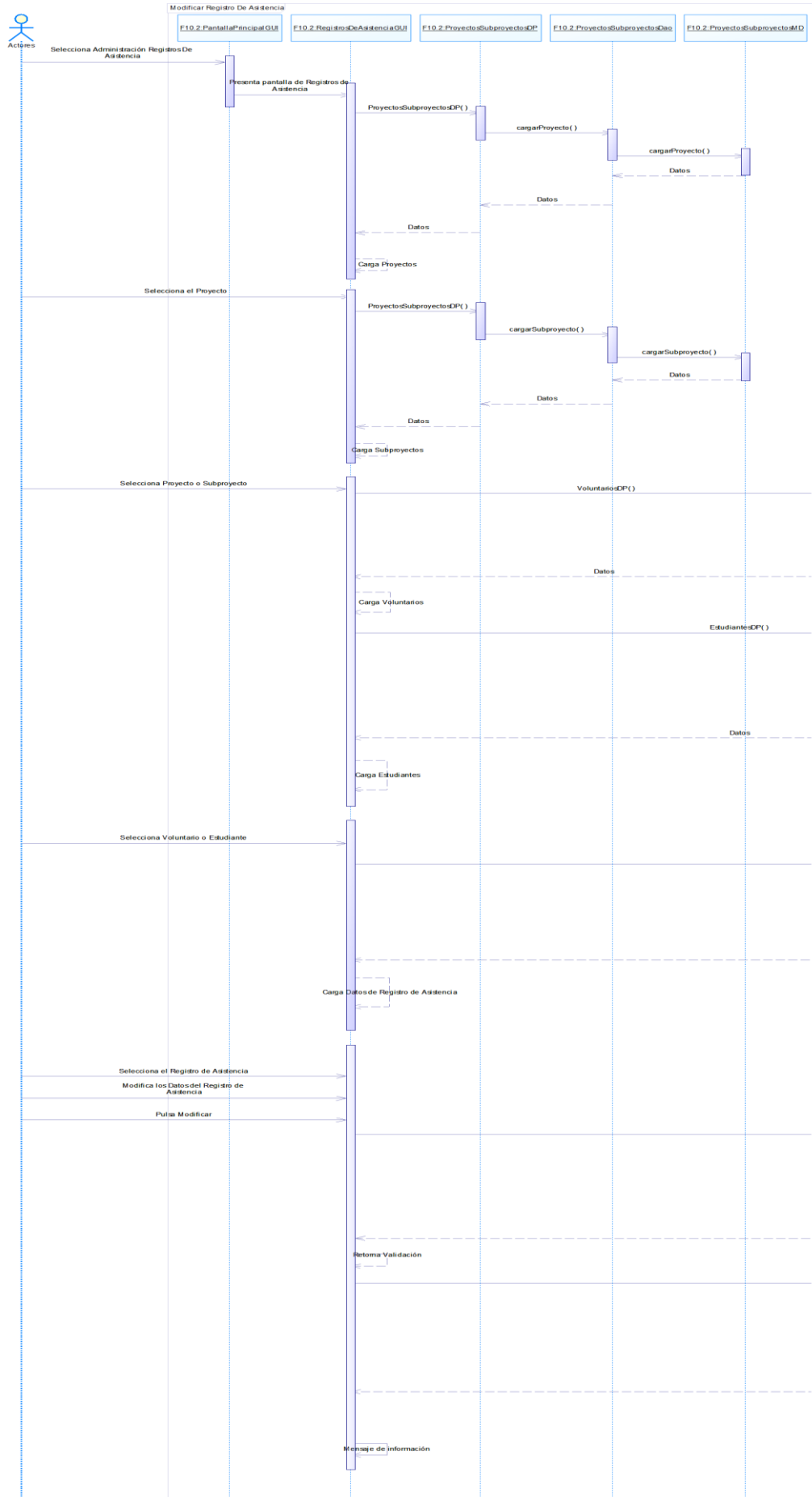


Figura 3.44: Diagrama de Secuencia Crear registro de asistencia

F10.2: Modificar registro de asistencia



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

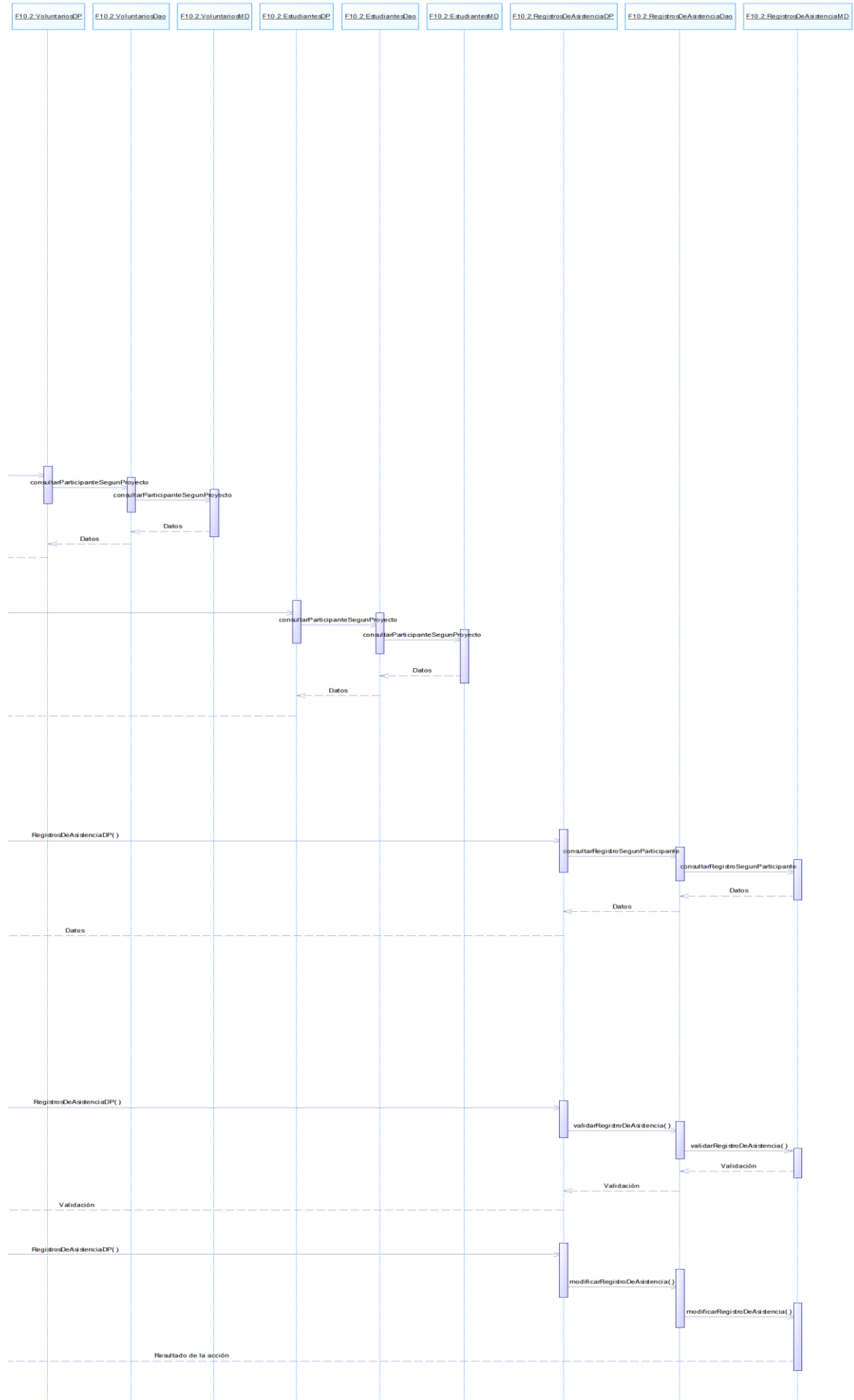


Figura 3.45: Diagrama de Secuencia Modificar registro de asistencia

F10.3: Consultar registro de asistencia

F10.3.1: Consulta general

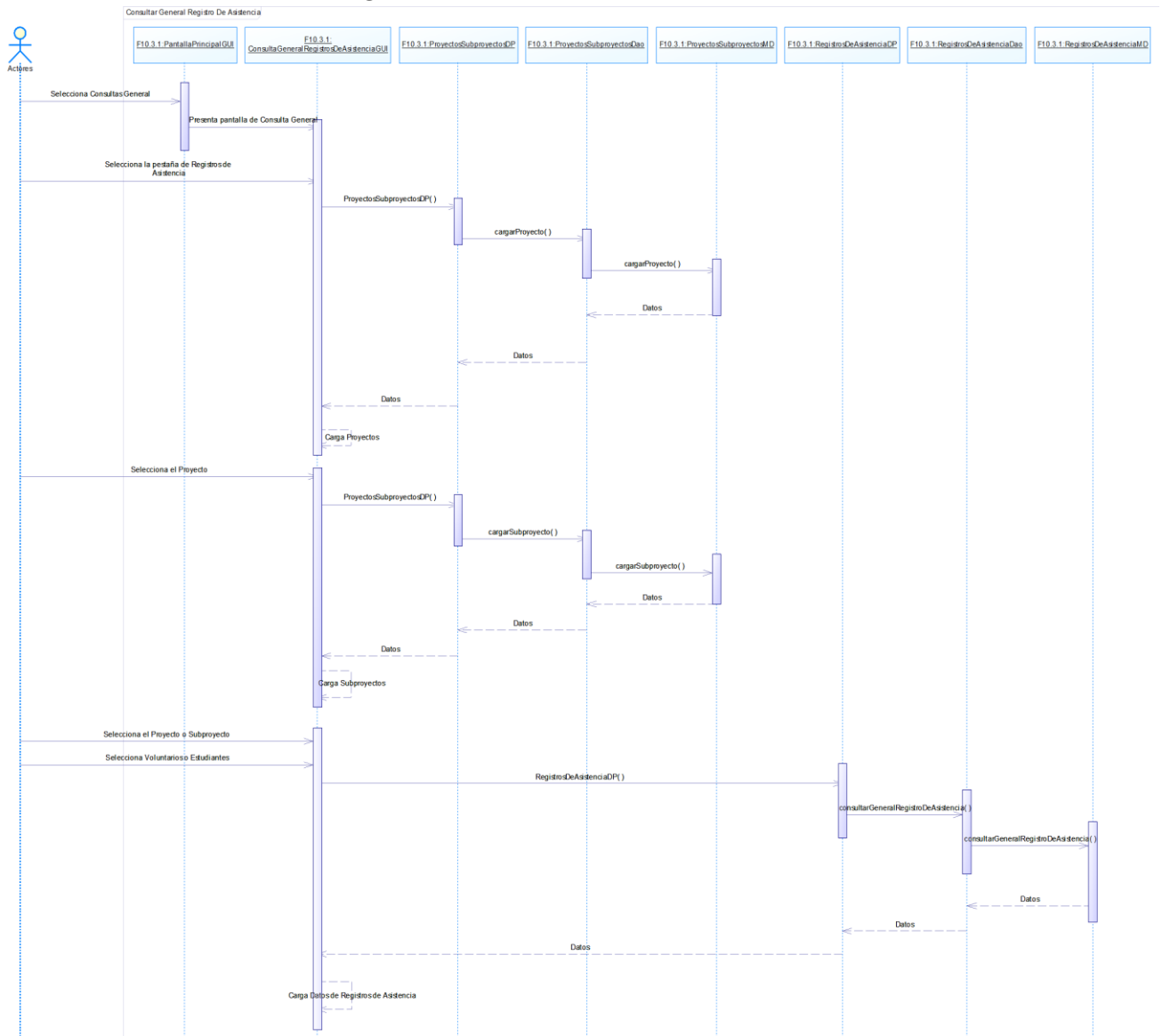
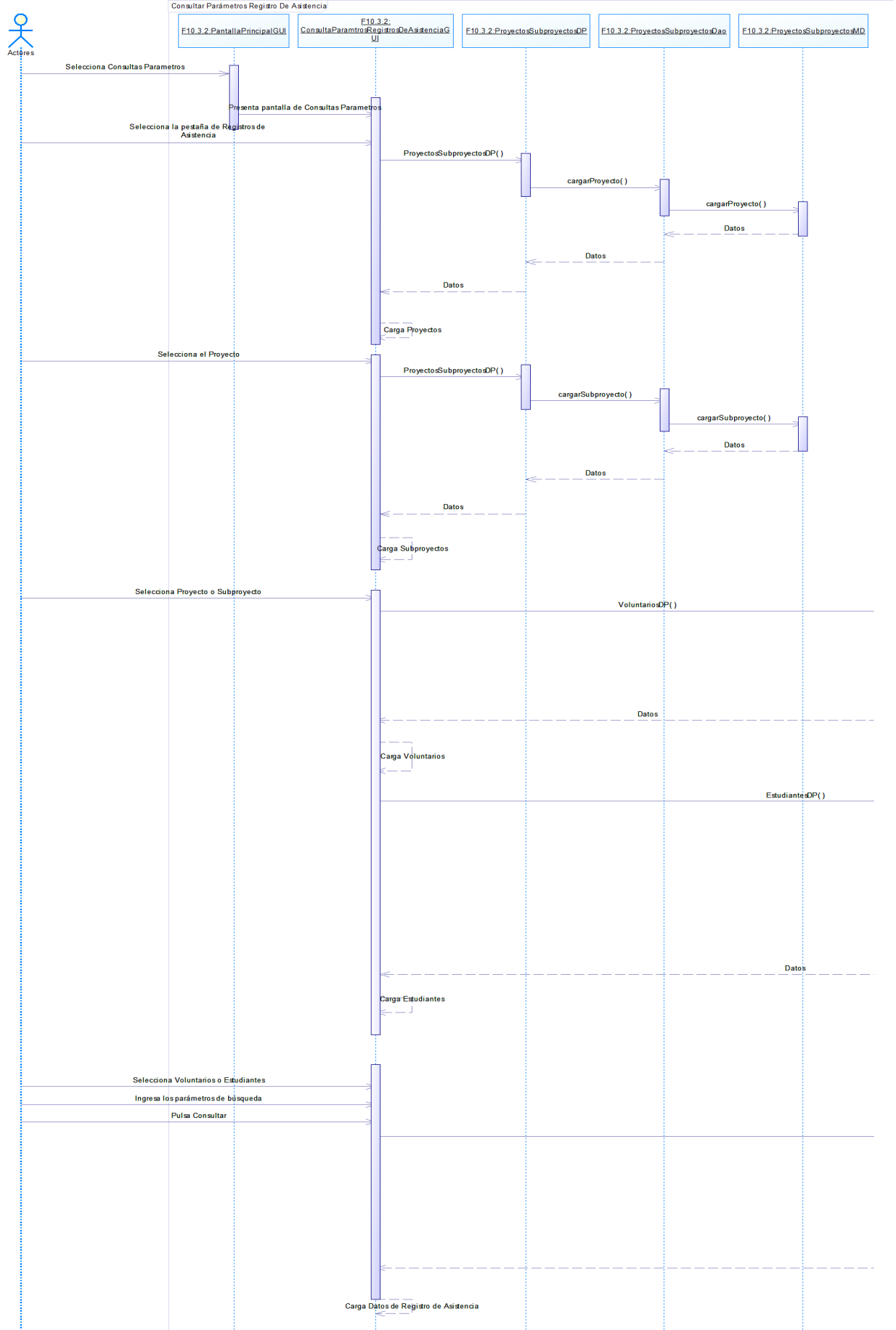


Figura 3.46: Diagrama de Secuencia Consulta general registro de asistencia

F10.3.2: Consulta por parámetros



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

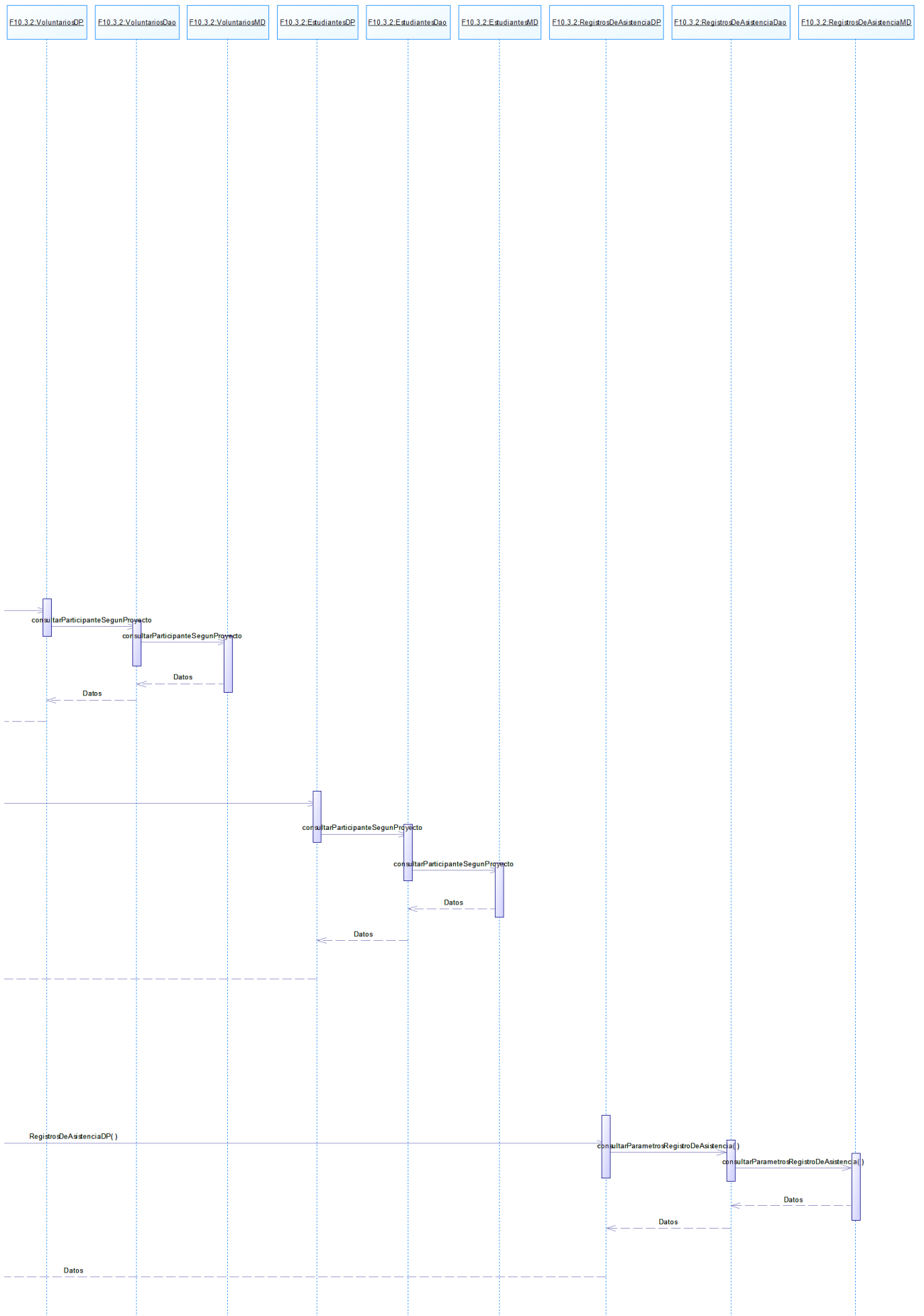
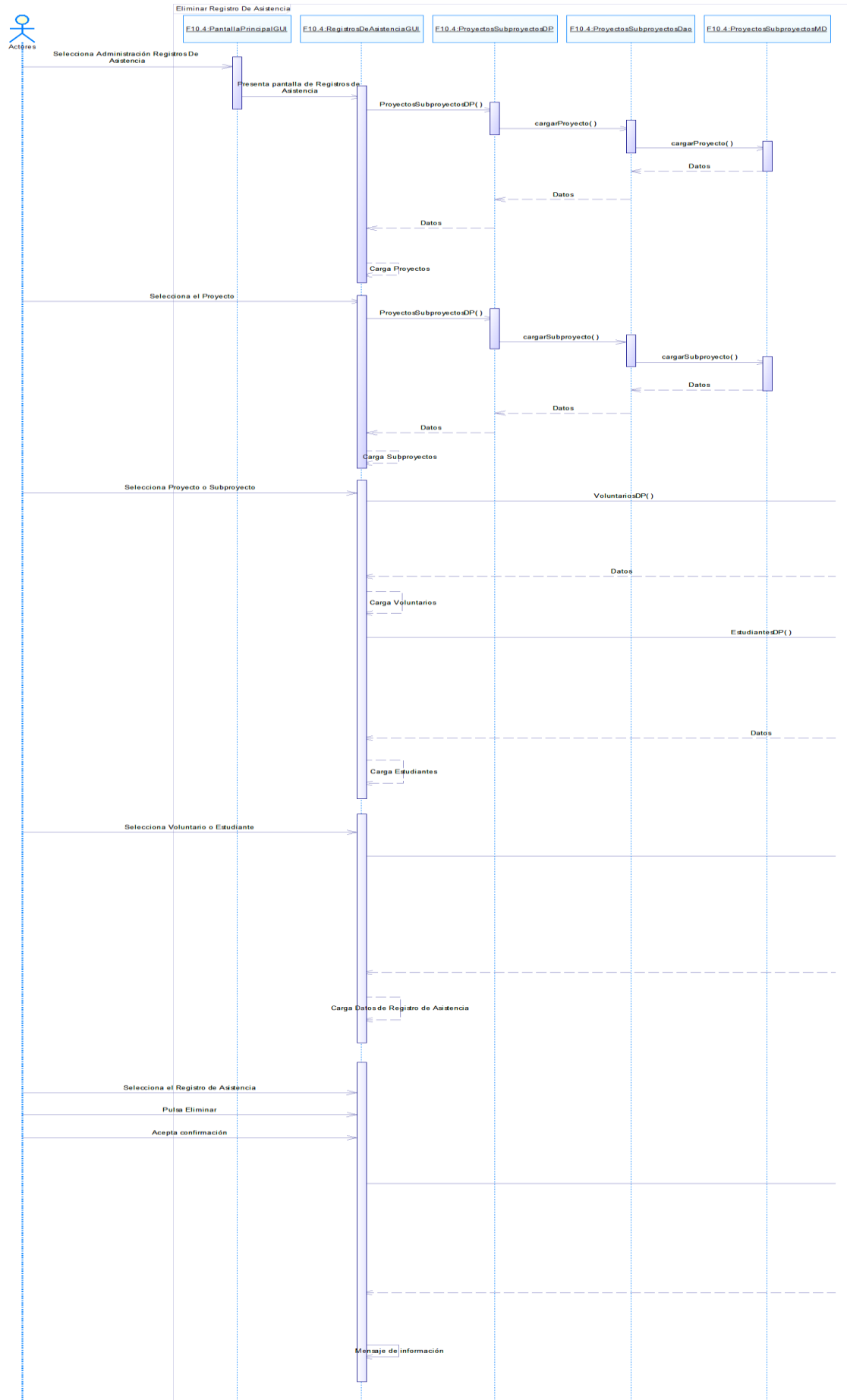


Figura 3.47: Diagrama de Secuencia Consulta por parámetros registro de asistencia

F10.4: Eliminar registro de asistencia



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

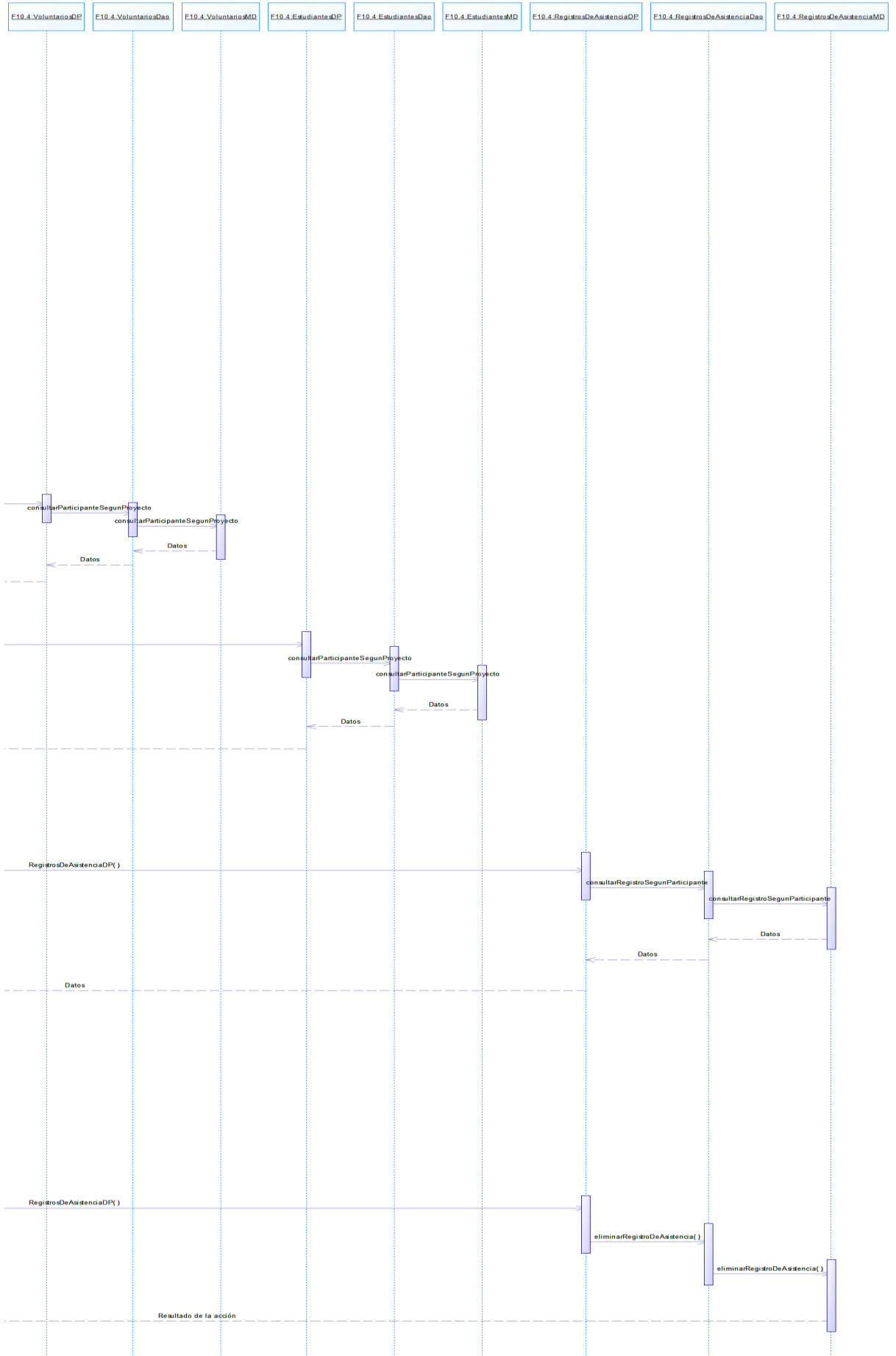
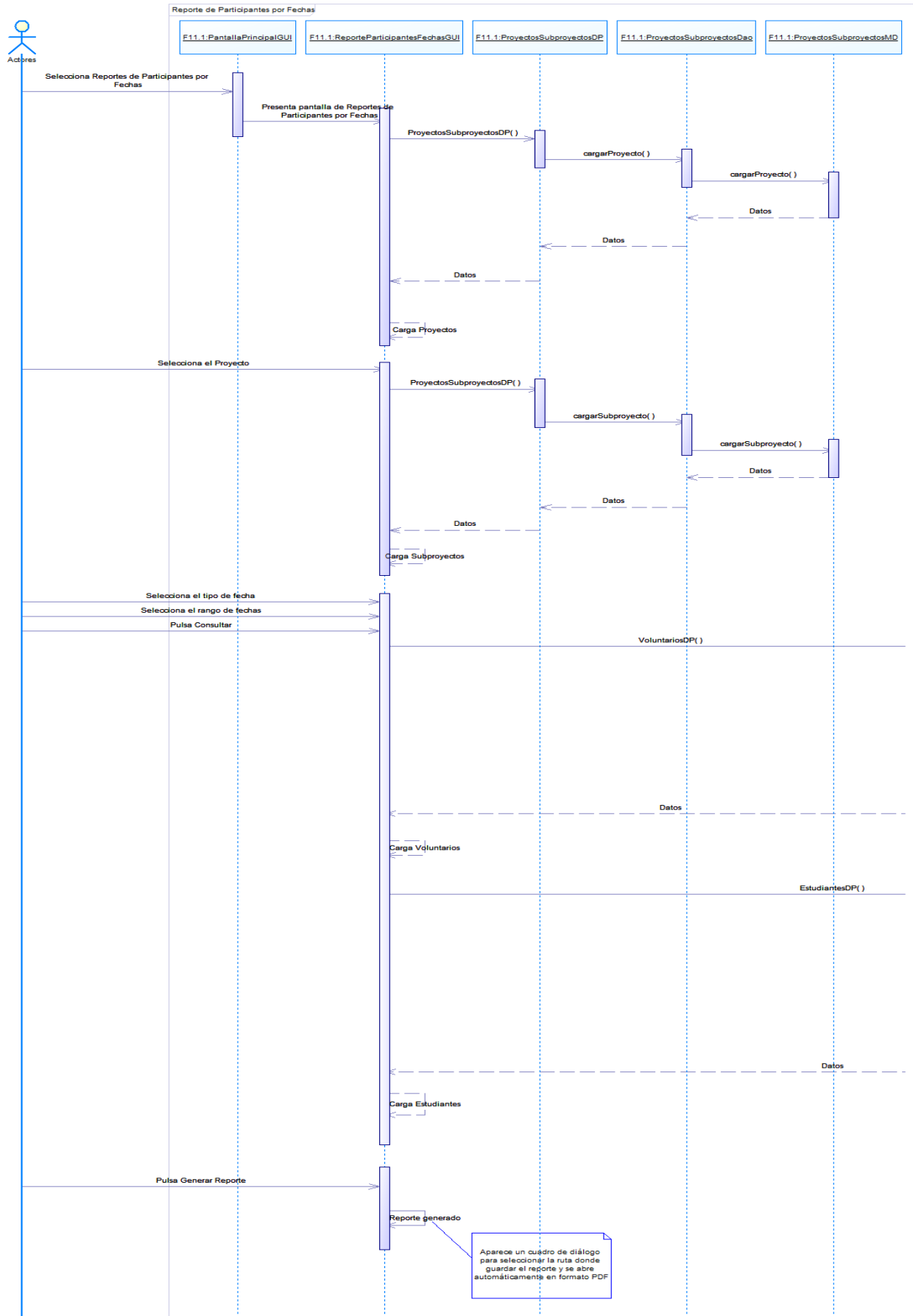


Figura 3.48: Diagrama de Secuencia Eliminar registro de asistencia

F11: Generación de reportes

F11.1: Reporte de participantes por fechas



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

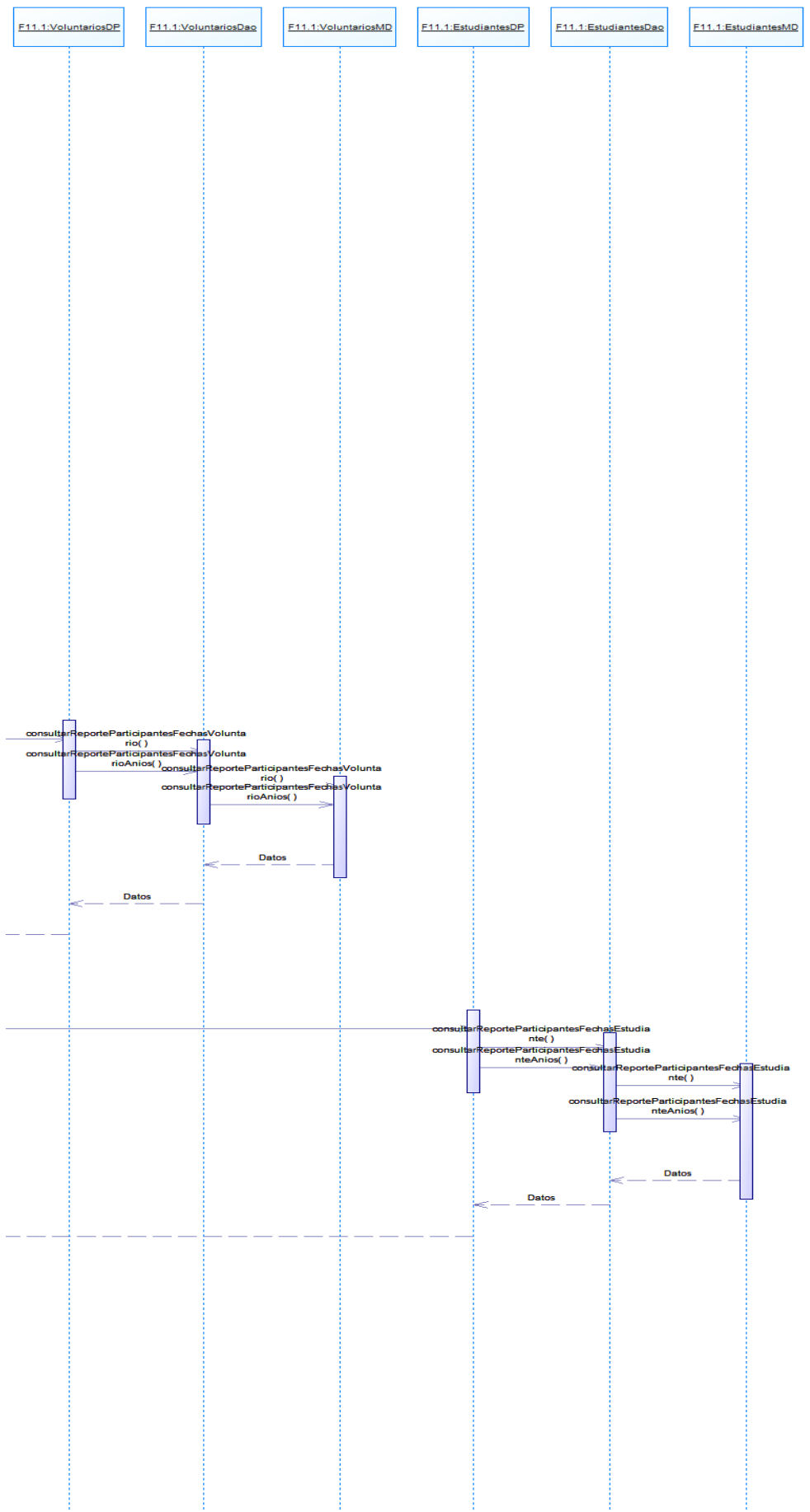
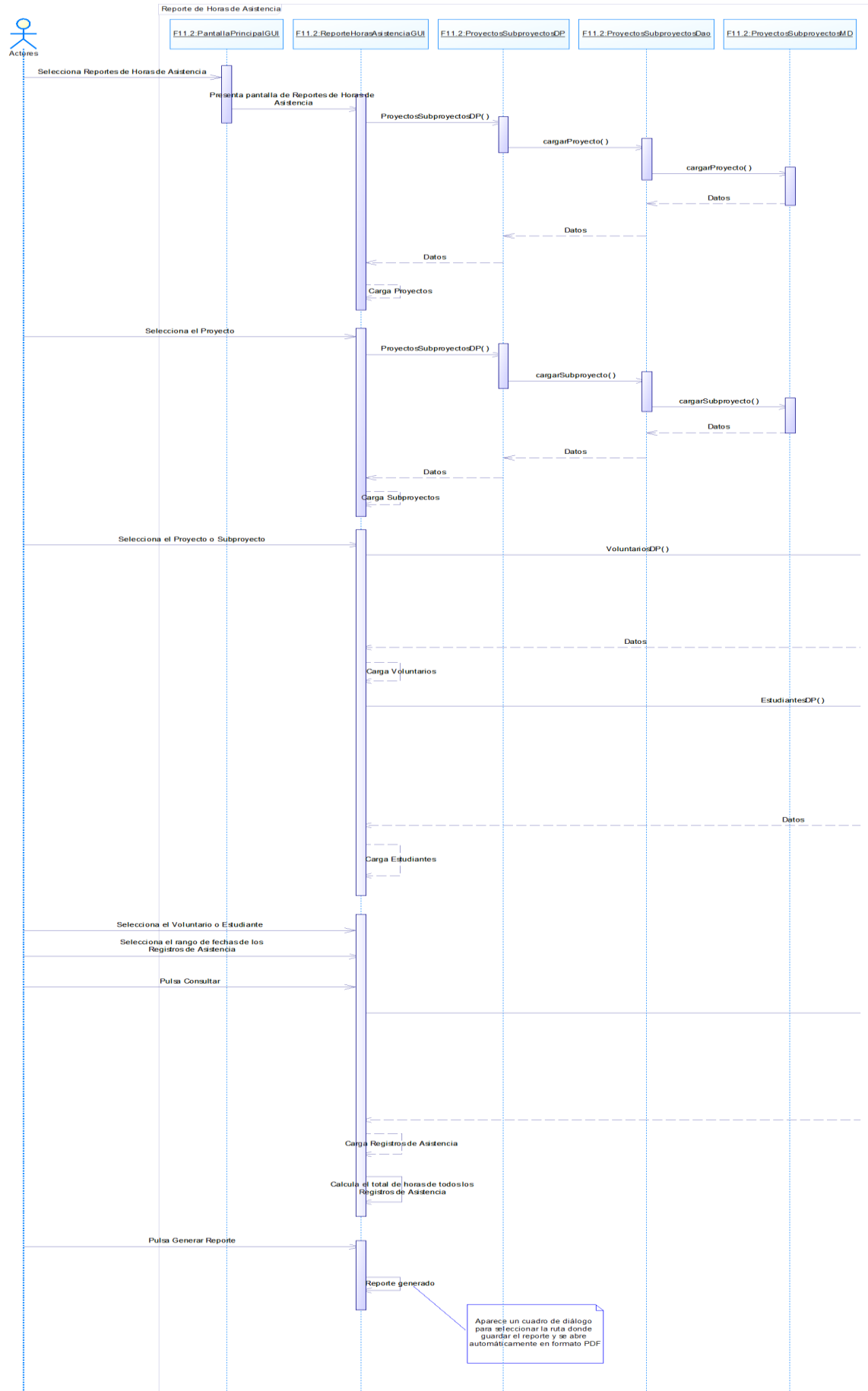


Figura 3.49: Diagrama de Secuencia Reporte de participantes por fecha

F11.2: Reporte de horas de asistencia



DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

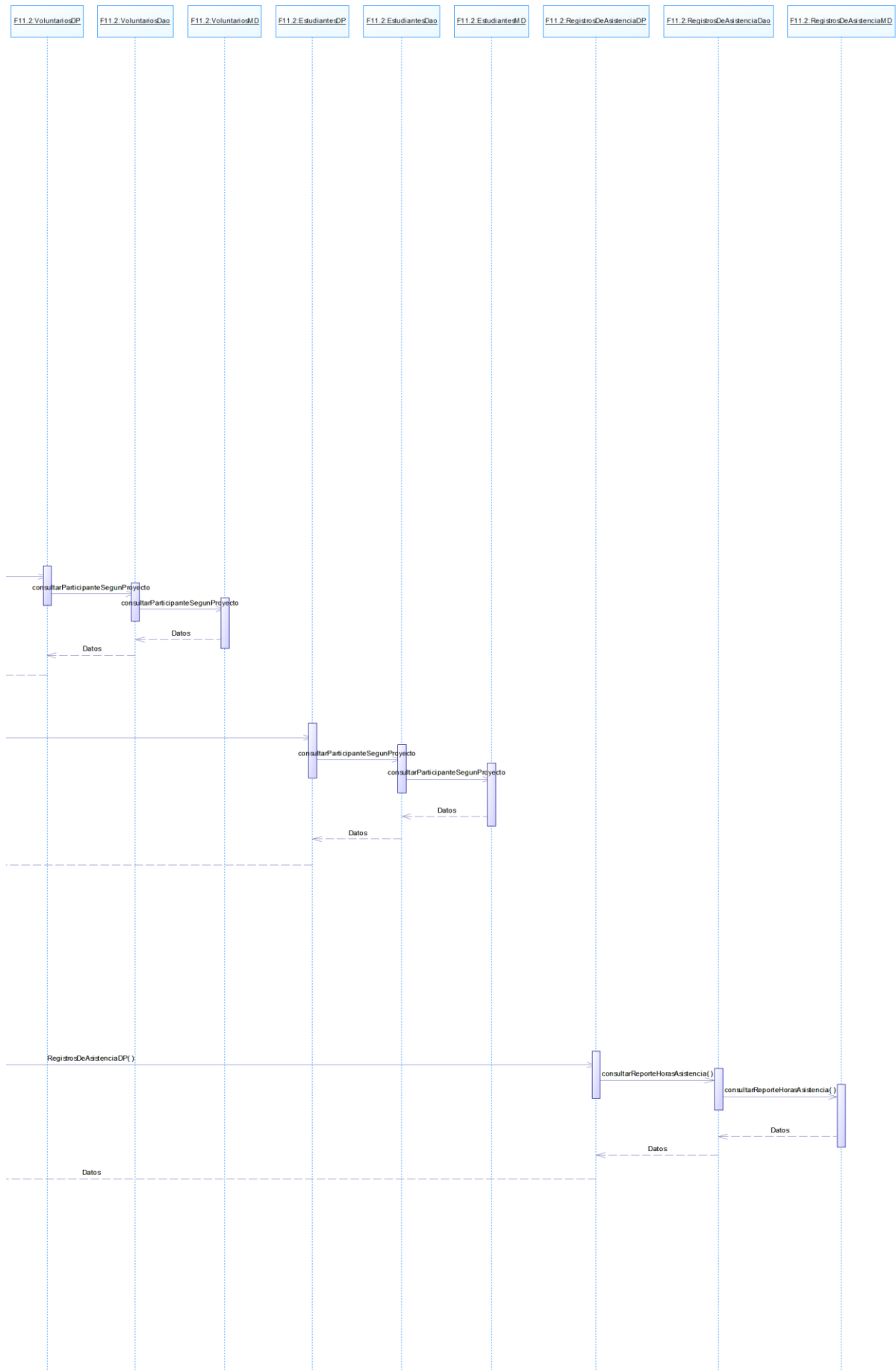


Figura 3.50: Diagrama de Secuencia Reporte de horas de asistencia

3.4. Diagramas de actividades

Se realizó los siguientes diagramas para los procesos principales del sistema:

INGRESO DE VOLUNTARIO

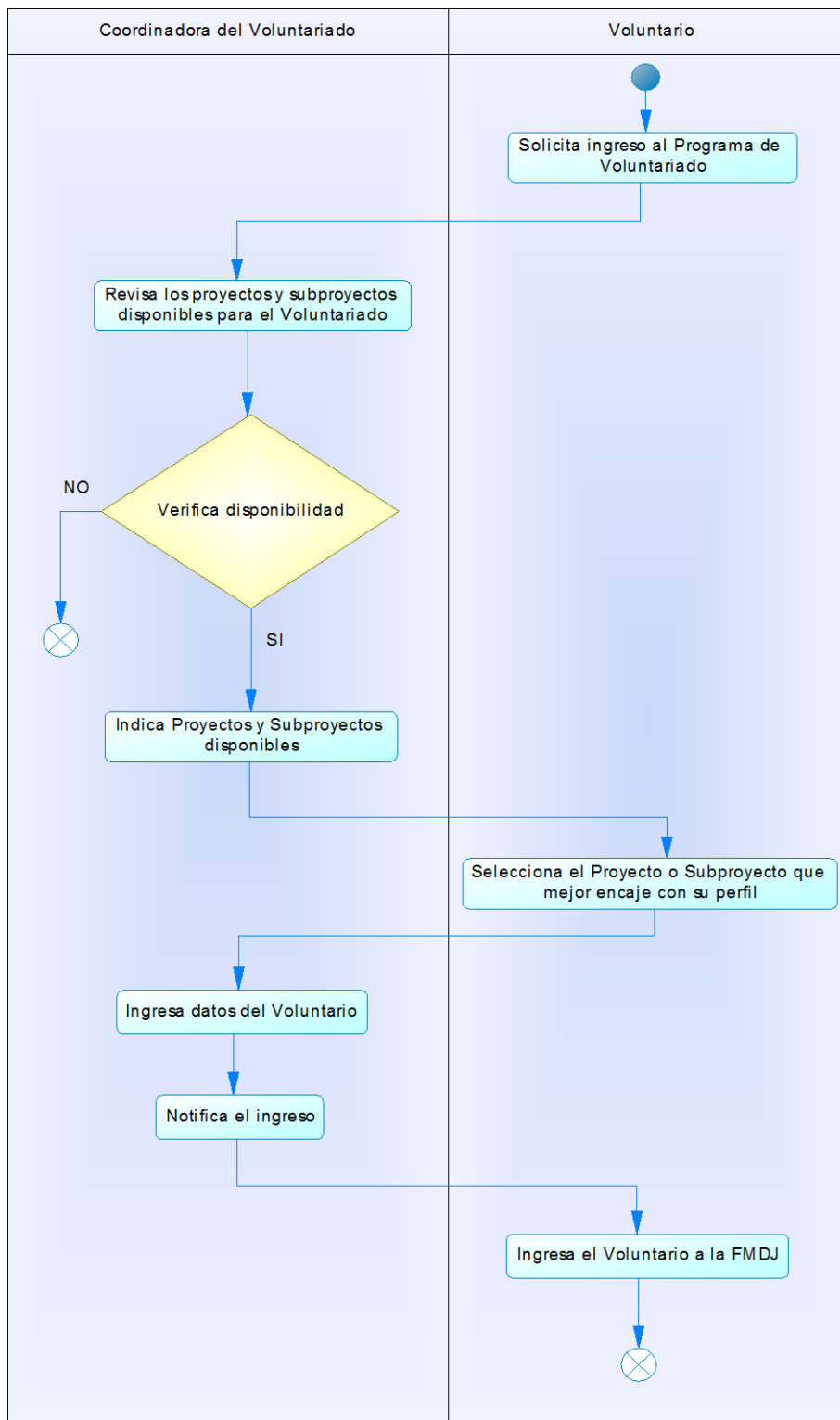


Figura 3.51: Diagrama de Actividades Ingreso de Voluntario

INGRESO DE ESTUDIANTE

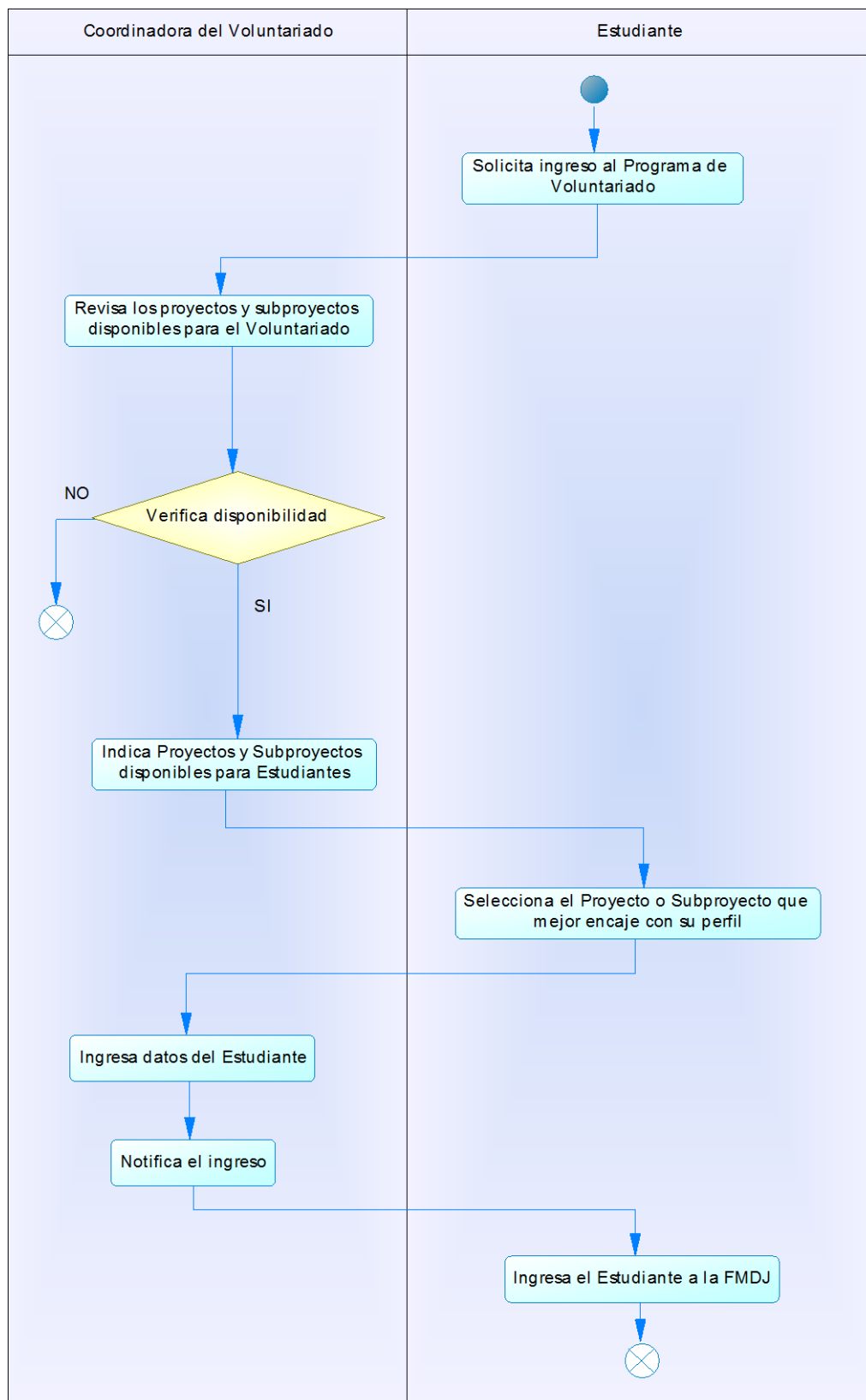


Figura 3.52: Diagrama de Actividades Ingreso de Estudiante

EVALUACION DE LAS HORAS DE ASISTENCIA

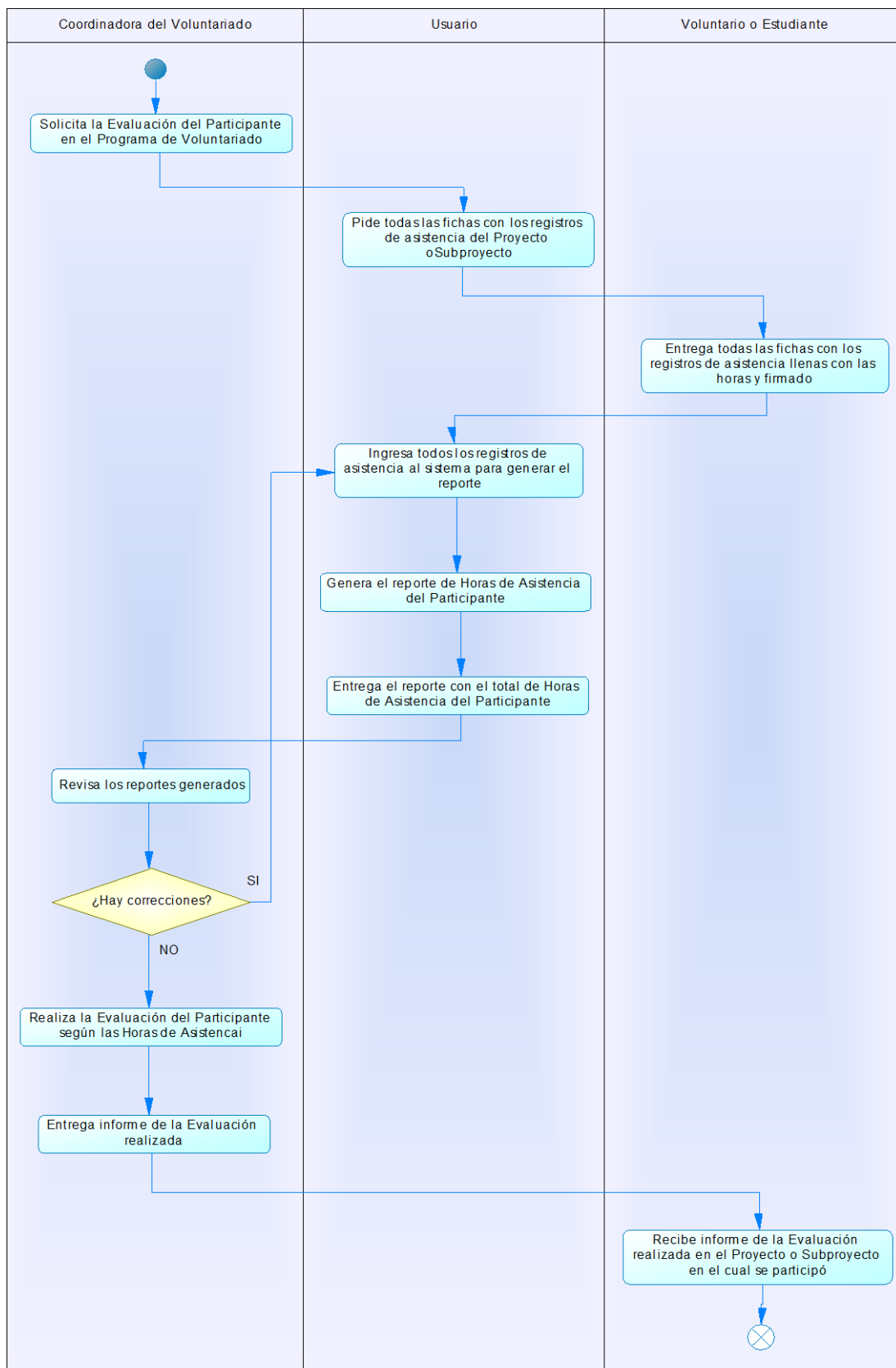


Figura 3.53: Diagrama de Actividades Evaluación de las horas de asistencia

3.5. Modelo Entidad Relación

3.5.1. Modelo conceptual

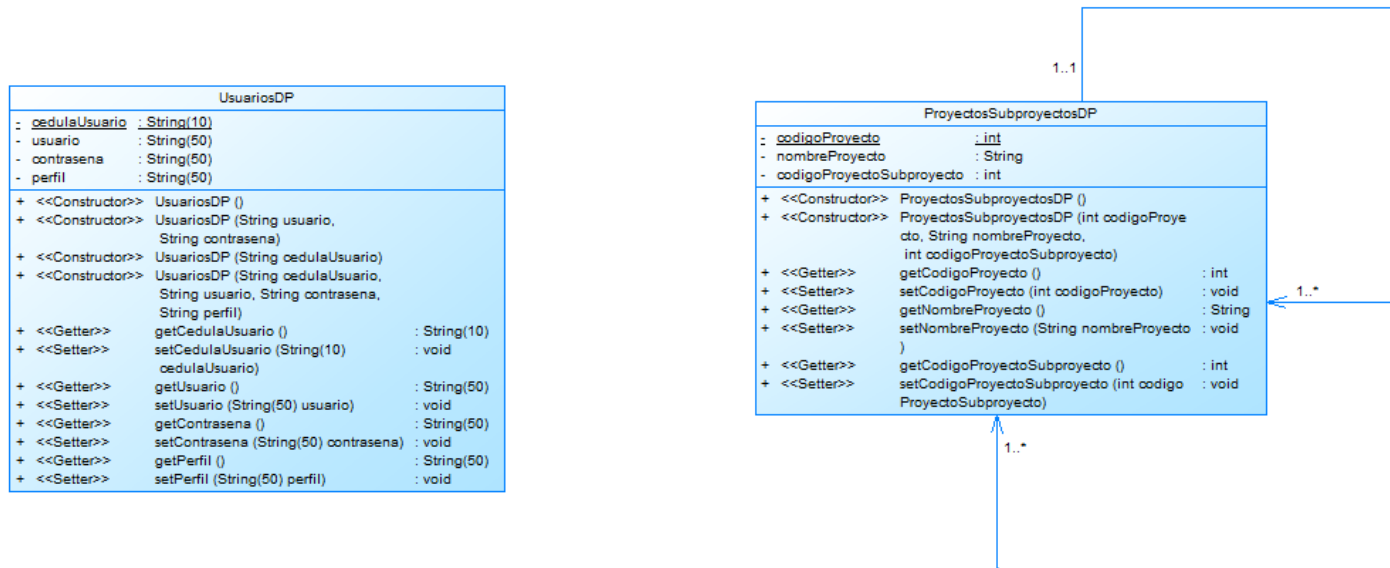


Figura 3.54: Diagrama de Modelo Conceptual

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

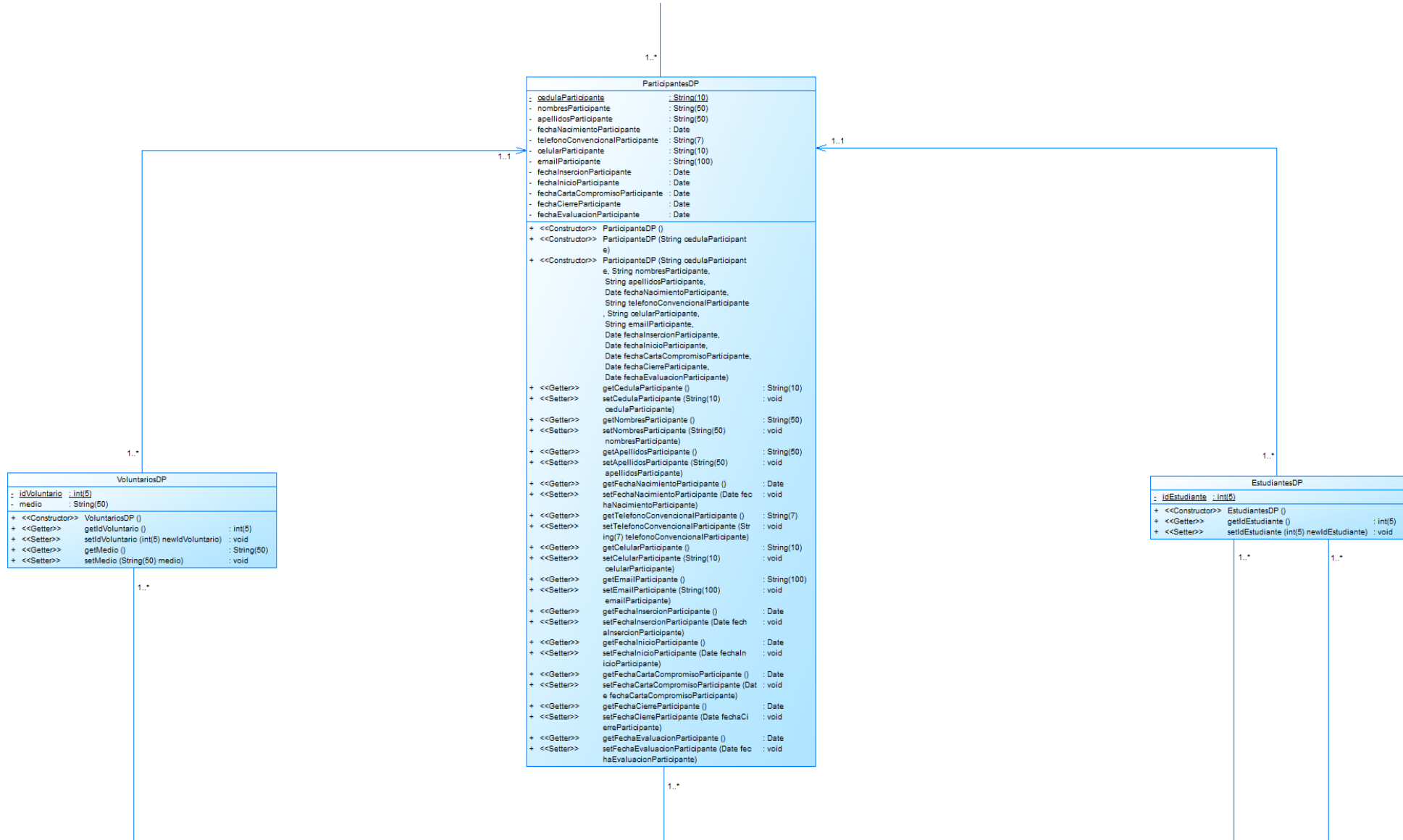


Figura 3.54: Diagrama de Modelo Conceptual

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

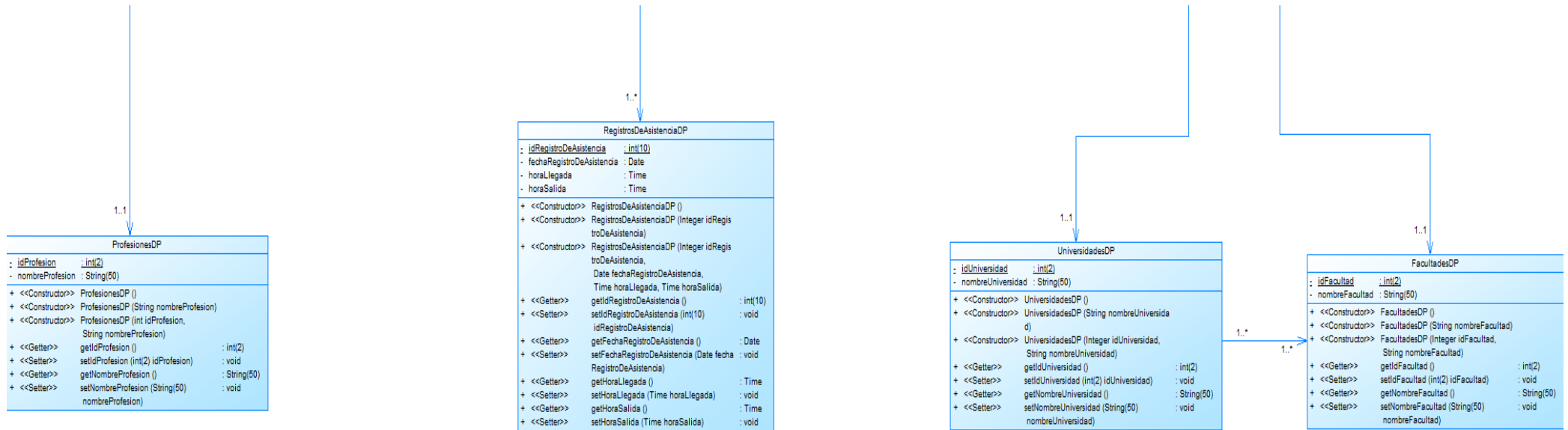


Figura 3.54: Diagrama de Modelo Conceptual

3.5.2. Modelo físico

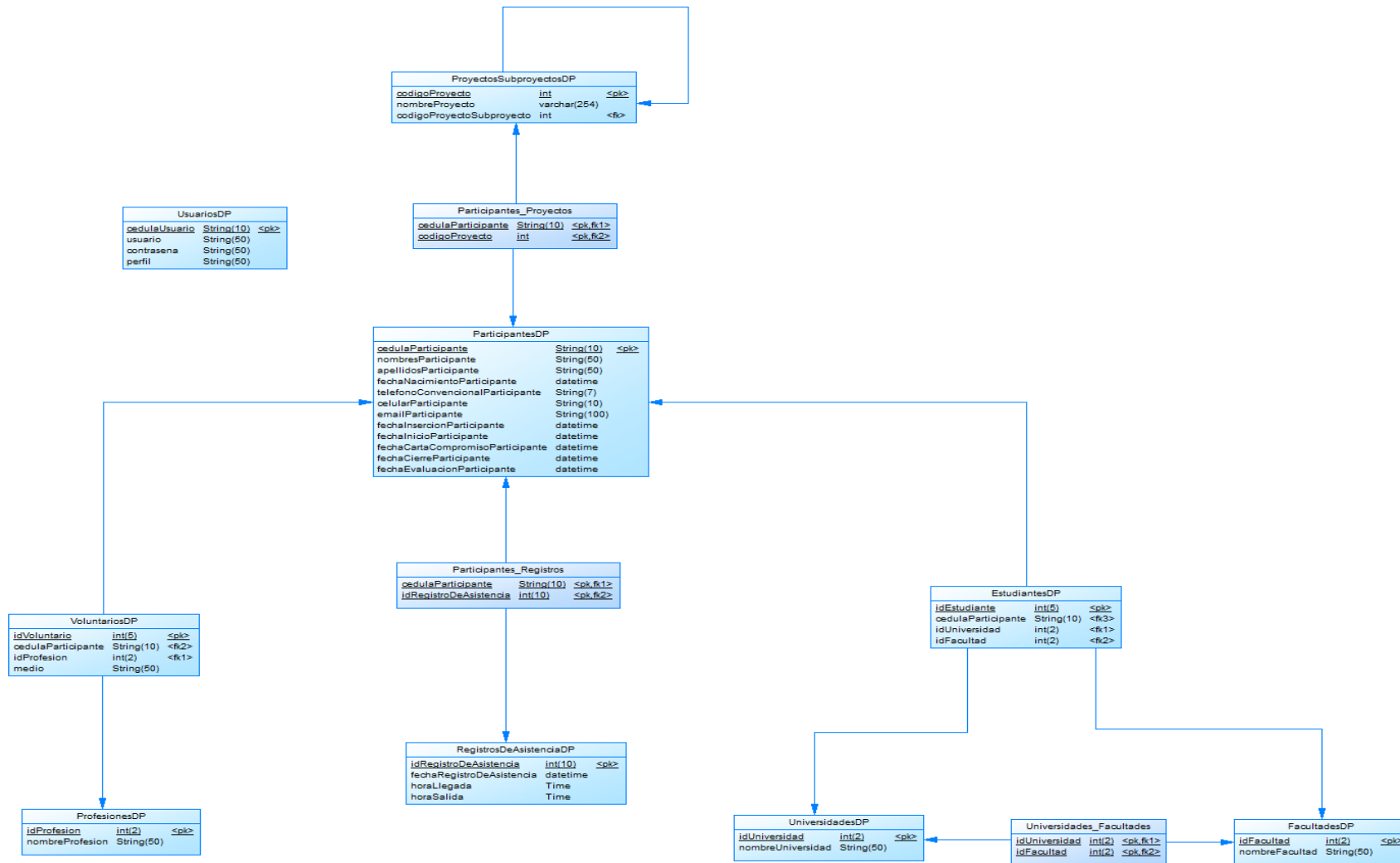


Figura 3.55: Diagrama de Modelo Físico

3.6. Definición de estándares de diseño y desarrollo

A continuación la definición de estándares asignado para el desarrollo de este sistema:

Validaciones en formularios

Validación de campos obligatorios: todos los campos de texto en donde sea obligatorio poner datos deben tener un asterisco de color rojo. Además al pulsar el botón de acción del formulario, se debe indicar un cuadro de diálogo indicando que se debe llenar los campos obligatorios. En cada formulario se deberá poner un mensaje indicando que el asterisco es para campos obligatorios.

Validación del formato de tipo de datos: todos los campos de texto donde se deba introducir solo texto no deberá permitir introducir números, así como en donde se deba introducir solo números no deberá permitir texto. En donde se introduzca texto se deberá controlar que solo sean mayúsculas exceptuando usuarios y contraseñas de inicio de sesión.

Validación de ingreso de datos correcto: se deberá usar listas desplegables para los datos que estén definidos por omisión en la base de datos.

Validaciones de búsqueda: Se deberá validar que los tipos de datos del formulario correspondan a los definidos en las columnas de las tablas, que no vengan valores en cero en los campos numéricos y vacíos en los campos alfanuméricos y que exista integridad de datos es, decir si se está buscando por campo llave, esta debe existir en la tabla.

Estándares de diseño de base de datos

Los nombres de las tablas deberán ser palabras en singular, solo en minúsculas y separadas por guiones bajos, ejemplo: registros_de_asistencia

En el caso de tablas que se relacionen específicamente con otras tablas, esta relación debe quedar expresada en el nombre, ejemplo: estudiantes_proyectos

Los campos de las tablas deberán ser palabras en singular, solo en mayúsculas y separadas por guiones bajos, ejemplo: CEDULA_ESTUDIANTE

Todas las tablas deben tener sus claves primarias¹³.

Toda relación entre tablas debe implementarse mediante claves foráneas¹⁴ con integridad referencial¹⁵. Esta deberá actualizar en cascada y restringir el borrado.

¹³ Clave primaria es un campo o una combinación de campos...

Las claves foráneas deben ser en mayúsculas de esta forma:

FK_VOLUNTARIO_PROFESION

Se deberá trabajar con procedimientos almacenados, definiéndolos en mayúsculas de esta manera: SP_CARGAR_USUARIO

Estándares de codificación

Se utilizará los estándares establecidos de JAVA:

Fichero fuente Java (.java): cada fichero debe coincidir con el nombre de la clase.

Debe tener estas secciones:

- ✓ Sentencias de paquete
- ✓ Sentencias de importación
- ✓ Comentarios de la clase o interfaz seguidos del nombre del autor y la versión
- ✓ Declaraciones de clases e interfaces

Clases: los nombres de las clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúscula, solo las últimas letras serán todas en mayúsculas, dependiendo de la capa de la arquitectura de programación a la que pertenezca, ejemplo: UsuariosGUI

Paquetes: Se escribirán siempre en letras minúsculas para evitar que entren en conflicto con los nombres de clases o interfaces. El prefijo del paquete siempre corresponderá a la capa de la arquitectura de programación a la que pertenezca, ejemplo: gui.administraciones

Métodos: Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera en minúsculas, seguida de la siguiente palabra con letra en mayúsculas, ejemplo: btnGuardarActionPerformed

Variables: Las variables se escribirán siempre en minúsculas. Las variables compuestas tendrán la primera letra de cada palabra componente en mayúsculas, ejemplo: usuarioDP

Nombre de las funciones y procedimientos: El nombre de una función debe iniciar en minúscula e inicia con un verbo, ejemplo: cargarUsuario()

¹⁴ Clave foránea es una limitación referencial...

¹⁵ Integridad referencial es una propiedad deseable...

Esquema de identificación para cada ítem

Controles Visuales

Los controles visuales deben poseer un nombre estándar, el cual se identificara con un prefijo todo en minúscula, seguido por el nombre del control, los elementos visuales como cajas de texto, listas o etiquetas deberán ser nombrados de acuerdo a la información que se va a mostrar/capturar, es decir:

Nombre	Prefijo
JLabel	lbl
JTextField	txt
JComboBox	cbbx
JXTaskPaneContainer	tskpc
JScrollPane	scrp
JCheckBox	chbx
JDateChooser	dtch
JRadioButton	rdbtn
JButton	btn
JXTable	tbl

Estándares de interfaz gráfica

Se utilizaran componentes Swing¹⁶ propios de Java que viene en la versión de NetBeansIDE 8.0.2, pero además también se utilizarán componentes de nuevas librerías de Java como son:

- ✓ swingx-all-1.6.4.jar
- ✓ jcalendar-1.3.3.jar

Para las pantallas principales se utilizará JFrames¹⁷ con las siguientes condiciones:

- ✓ La pantalla principal será del tamaño de la pantalla de la máquina
- ✓ No se podrán maximizar
- ✓ Se abrirán centradas en la pantalla de la máquina

Para las Administraciones y Reportes se utilizaran JPanels¹⁷ que irán añadidos al JFrame de Pantalla Principal a excepción de las Administraciones de Proyectos/Subproyectos, Universidades/Facultades y Profesiones, los cuales serán JFrames independientes para poder acceder a ellos de diferentes partes del sistema.

Para las Consultas se utilizarán JPanels divididos en Pestañas según las Administraciones.

¹⁶ Swing es una biblioteca gráfica de Java...

¹⁷ JFrames y JPanels son clases utilizadas en Swing para...

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

Para los reportes se generarán en formato PDF que podrán ser almacenados en la máquina.

Manejo de errores y mensajes

Se utilizarán tres tipos de cuadros de diálogo: Cuadro de diálogo de Error, Cuadro de Diálogo de Información, Cuadro de Diálogo de Confirmación.

Cuadro de Diálogo de Error: manejará todos los mensajes de errores en el sistema.

Cuadro de Diálogo de Información: manejará todos los mensajes de información de que una acción fue realizada con éxito en el sistema.

Cuadro de Diálogo de Confirmación: manejará todas las confirmaciones para los métodos de eliminación de objetos en el sistema.

Como dato adicional, se implementará un manejo de logs¹⁸ por medio de una librería de JAVA llamada Log4J. Esto permitirá escribir logs en un archivo externo dejando constancia de las determinadas excepciones que sucedan en tiempo de ejecución, para luego poder verificar si existe algún problema y el lugar exacto en donde se está desarrollando el problema.

¹⁸ Logs son registros oficiales de eventos durante...

CAPÍTULO 4.- PRUEBAS DEL SISTEMA

4.1. Plan de pruebas de integración

El principal objetivo de las pruebas de integración es el detectar las fallas de interacción entre las distintas clases que componen la aplicación. Al realizar estas pruebas, estamos utilizando un mecanismo para comprobar el correcto ensamblaje del sistema completo, ya que cada clase es probada unitariamente insertándola de manera progresiva dentro de la estructura siguiendo los lineamientos dictados por el diseño.

Cuando realizamos la integración de los módulos, se debe concentrar la búsqueda de defectos como los que puedan provocar las excepciones arrojadas por los métodos, el empleo de operaciones equivocadas o la invocación incorrecta de los métodos al pasar parámetros equivocados, entre otras.

Para el desarrollo de esta aplicación se ha elegido la integración por dependencias ya que es un modelo que nos indica el orden de integración de cada clase hasta armar totalmente el sistema. Por lo tanto, se ha elegido la estrategia de pruebas de integración por dependencia de la clase.

En esta estrategia se comienza probando las clases llamadas independientes, las cuales son las que usan muy poca o ninguna otra clase del sistema. Después, se continúa probando las clases dependientes, las cuales son las que usan a las independientes. Esta secuencia de pruebas por capas de clases dependientes continúa hasta armar el sistema entero.

Las Clases Independientes del Sistema serían:

- Usuarios
- Proyectos
- Subproyectos
- Profesiones
- Universidades
- Facultades

Las Clases Dependientes del Sistema serían:

- Voluntarios
- Estudiantes
- Registros de Asistencia

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

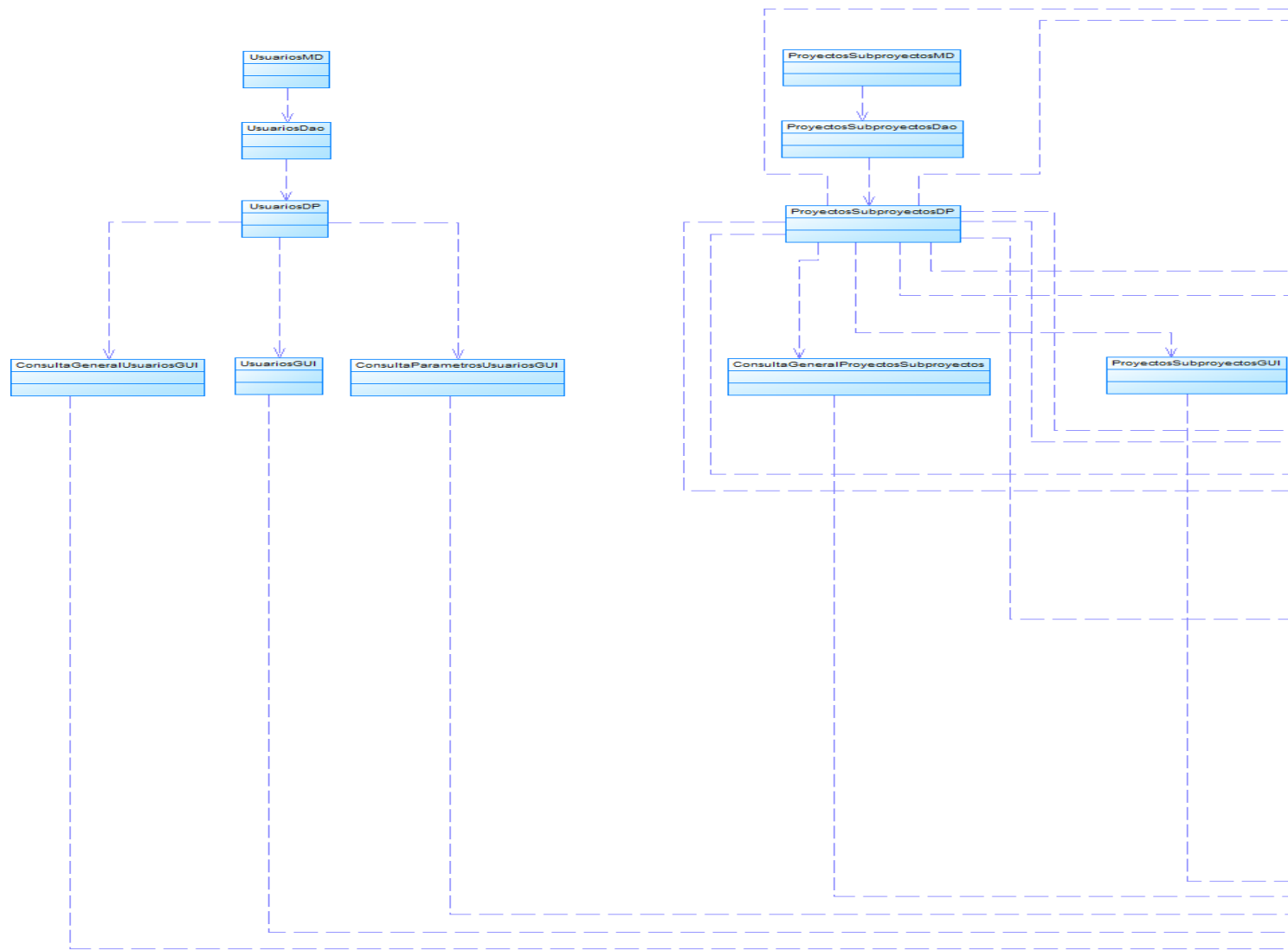


Figura 4.1: Diagrama de Integración por dependencias

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

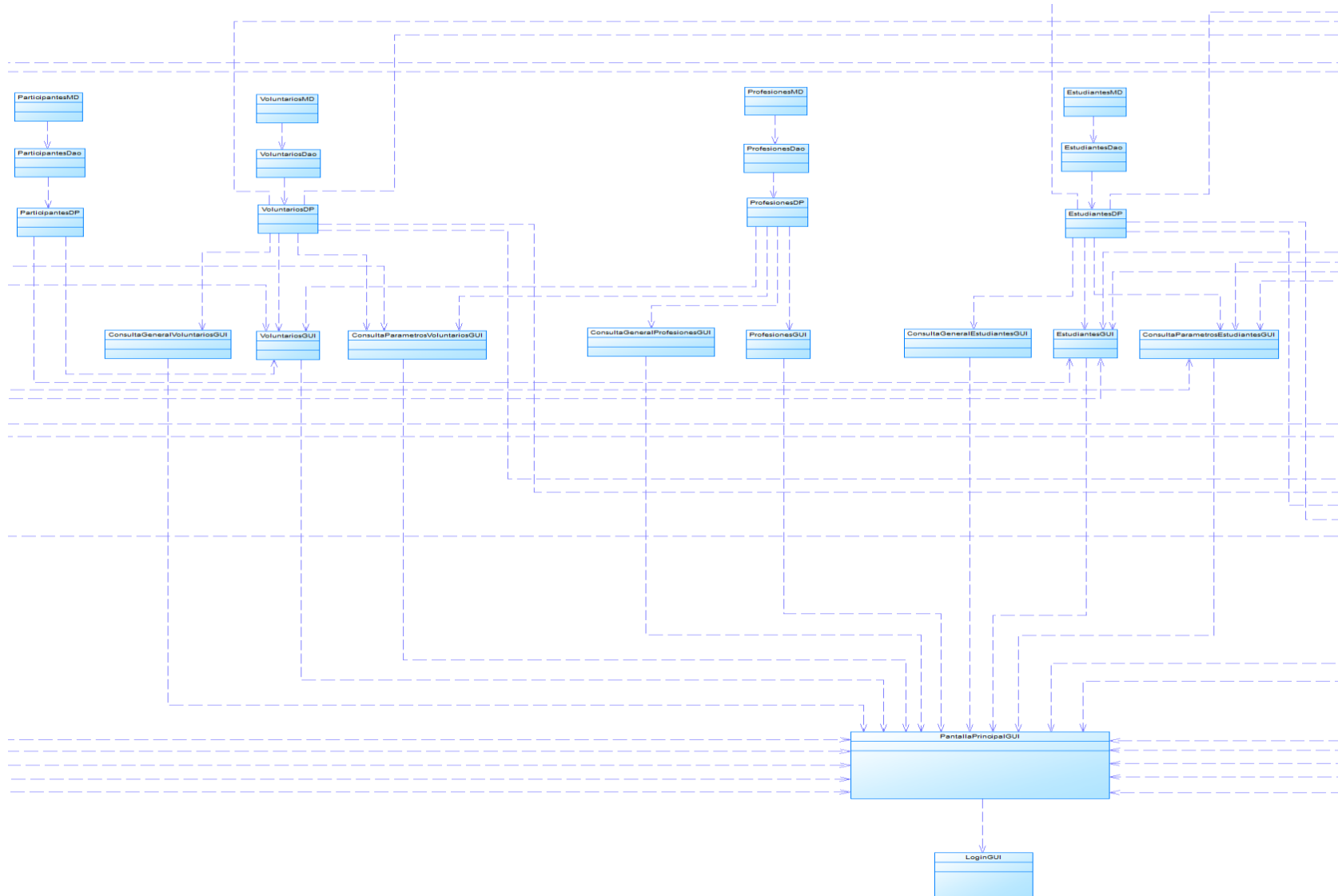


Figura 4.1: Diagrama de Integración por dependencias

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

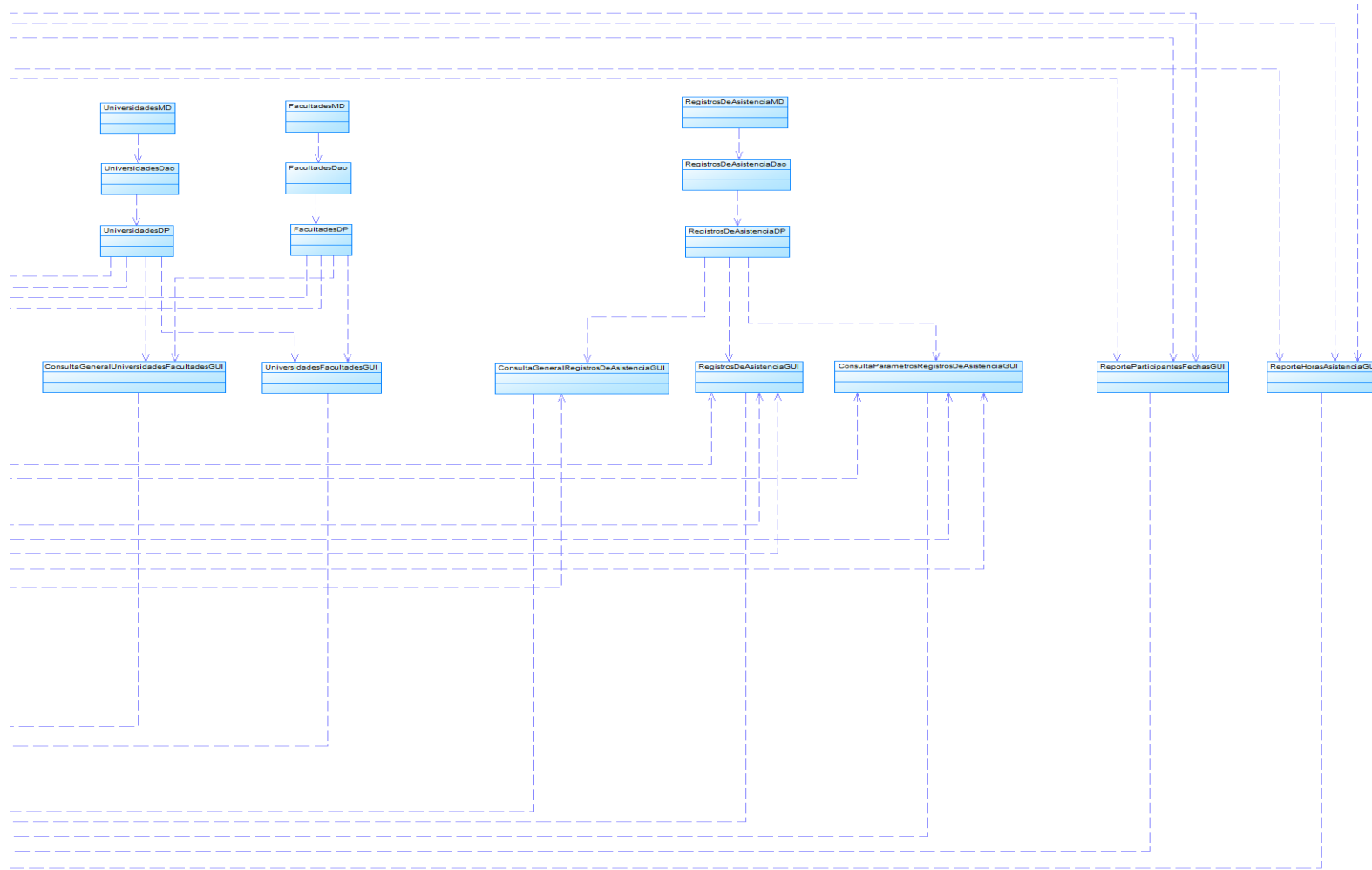


Figura 4.1: Diagrama de Integración por dependencias

4.2. Pruebas de los casos de uso

Para las pruebas de los casos de uso se utilizará como referencia los flujos generados anteriormente en los casos de uso a detalle. Son los siguientes:

- Flujos principales: los cuales se dan en el caso de que no existan errores.
- Flujos alternos: los cuales se dan cuando el flujo principal puede desviarse por un flujo alternativo.
- Excepciones: las cuales se dan en caso de existir errores.

Con estos flujos se generarán escenarios posibles para crear matrices de casos de prueba que se los aplicará a los casos de uso.

Es importante indicar que solo se presentarán las pruebas de los casos de uso principales ya que existen algunos casos de uso en donde sus flujos son parecidos.

F1: Autenticación de usuario

F1.1: Ingresar al sistema

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Ingreso correcto	Los datos son validados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Datos ingresados incorrectos	Los datos son validados incorrectamente	E2
Error en la base de datos	Al validar existe un error en la base de datos	E3

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Ingreso correcto	Usuario y Contraseña	Usuario validado correctamente y se abre la Pantalla Principal
Datos obligatorios vacíos	Usuario y Contraseña	Mensaje: Debe ingresar Usuario y Contraseña
Datos ingresados incorrectos	Usuario y Contraseña	Mensaje: El Usuario y/o Contraseña no son válidos
Error en la base de datos	Al validar Usuario existe un error en la base de datos	Mensaje de error al intentar consultar Usuario en la base de datos

F2: Administración de usuarios

F2.1: Crear usuario

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Creación correcta	Los datos son creados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Datos ingresados incorrectos	Los datos son validados incorrectamente	E2
Error en la base de datos	Al crear existe un error en la base de datos	E3

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Creación correcta	Cédula, Usuario, Contraseña y Perfil	Mensaje: El nuevo Usuario ha sido creado exitosamente
Datos obligatorios vacíos	Cédula, Usuario, Contraseña y Perfil	Mensaje: Debe llenar todos los campos
Datos ingresados incorrectos	Cédula	Mensaje: Debe ingresarse un número de cédula de 10 dígitos
Error en la base de datos	Al crear Usuario existe un error en la base de datos	Mensaje de error al intentar crear Usuario en la base de datos

F2.2: Modificar usuario

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Modificación correcta	Los datos son modificados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Datos ingresados incorrectos	Los datos son validados incorrectamente	E2
Error en la base de datos	Al modificar existe un error en la base de datos	E3

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Modificación correcta	Cédula, Usuario, Contraseña y Perfil	Mensaje: El Usuario ha sido modificado exitosamente
Datos obligatorios vacíos	Cédula, Usuario,	Mensaje: Debe llenar

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	Contraseña y Perfil	todos los campos
Datos ingresados incorrectos	Cédula	Mensaje: Debe ingresarse un número de cédula de 10 dígitos
Error en la base de datos	Al modificar Usuario existe un error en la base de datos	Mensaje de error al intentar modificar Usuario en la base de datos

F2.3: Consultar usuario

F2.3.1: Consulta general

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Error en la base de datos	Al consultar existe un error en la base de datos	E1

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Error en la base de datos	Al consultar Usuario existe un error en la base de datos	Mensaje de error al intentar consultar Usuario en la base de datos

F2.3.2: Consulta por parámetros

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Parámetros vacíos	No se llena al menos un parámetro	E1
Error en la base de datos	Al consultar existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Parámetros vacíos	Cédula y Perfil	Mensaje: Debe llenar al menos un parámetro
Error en la base de datos	Al consultar Usuario existe un error en la base de datos	Mensaje de error al intentar crear Usuario en la base de datos

F2.4: Eliminar usuario

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Eliminación correcta	Los datos son eliminados	

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	correctamente	
Error en la base de datos	Al eliminar existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Eliminación correcta	Cédula, Usuario, Contraseña y Perfil	Mensaje: El Usuario ha sido eliminado exitosamente
Error en la base de datos	Al eliminar Usuario existe un error en la base de datos	Mensaje de error al intentar eliminar Usuario en la base de datos

F3: Administración de proyectos

F3.1: Crear proyecto

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Creación correcta	Los datos son creados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Error en la base de datos	Al crear existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Creación correcta	Nombre del Proyecto	Mensaje: El nuevo Proyecto ha sido creado exitosamente
Datos obligatorios vacíos	Nombre del Proyecto	Mensaje: Debe llenar todos los campos
Error en la base de datos	Al crear Proyecto existe un error en la base de datos	Mensaje de error al intentar crear Proyecto en la base de datos

F3.2: Modificar proyecto

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Modificación correcta	Los datos son modificados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Error en la base de datos	Al modificar existe un error en la base de datos	E2

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Modificación correcta	Nombre del Proyecto	Mensaje: El Proyecto ha sido modificado exitosamente
Datos obligatorios vacíos	Nombre del Proyecto	Mensaje: Debe llenar todos los campos
Error en la base de datos	Al modificar Proyecto existe un error en la base de datos	Mensaje de error al intentar modificar Proyecto en la base de datos

F3.3: Consultar proyecto

F3.3.1: Consulta general

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Error en la base de datos	Al consultar existe un error en la base de datos	E1

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Error en la base de datos	Al consultar Proyecto existe un error en la base de datos	Mensaje de error al intentar consultar Proyecto en la base de datos

F3.4: Eliminar proyecto

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Eliminación correcta	Los datos son eliminados correctamente	
Error en la base de datos	Al eliminar existe un error en la base de datos	E1

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Eliminación correcta	Nombre del Proyecto	Mensaje: El Proyecto ha sido eliminado exitosamente
Error en la base de datos	Al eliminar Proyecto existe un error en la base de datos	Mensaje de error al intentar eliminar Proyecto en la base de datos

F4: Administración de subproyectos

F4.1: Crear subproyecto

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Creación correcta	Los datos son creados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Error en la base de datos	Al crear existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Creación correcta	Nombre del Subproyecto, Nombre del Proyecto	Mensaje: El nuevo Subproyecto ha sido creado exitosamente
Datos obligatorios vacíos	Nombre del Subproyecto, Nombre del Proyecto	Mensaje: Debe llenar todos los campos
Error en la base de datos	Al crear Subproyecto existe un error en la base de datos	Mensaje de error al intentar crear Subproyecto en la base de datos

F4.2: Modificar subproyecto

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Modificación correcta	Los datos son modificados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Error en la base de datos	Al modificar existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Modificación correcta	Nombre del Subproyecto	Mensaje: El Subproyecto ha sido modificado exitosamente
Datos obligatorios vacíos	Nombre del Subproyecto	Mensaje: Debe llenar todos los campos
Error en la base de datos	Al modificar Subproyecto existe un error en la base de datos	Mensaje de error al intentar modificar Subproyecto en la base

		de datos
--	--	----------

F4.3: Consultar subproyecto

F4.3.1: Consulta general

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Error en la base de datos	Al consultar existe un error en la base de datos	E1

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Error en la base de datos	Al consultar Subproyecto existe un error en la base de datos	Mensaje de error al intentar consultar Subproyecto en la base de datos

F4.4: Eliminar subproyecto

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Eliminación correcta	Los datos son eliminados correctamente	
Error en la base de datos	Al eliminar existe un error en la base de datos	E1

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Eliminación correcta	Nombre del Subproyecto	Mensaje: El Subproyecto ha sido eliminado exitosamente
Error en la base de datos	Al eliminar Subproyecto existe un error en la base de datos	Mensaje de error al intentar eliminar Subproyecto en la base de datos

F5: Administración de voluntarios

F5.1: Crear voluntario

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Creación correcta	Los datos son creados correctamente	
Datos obligatorios vacíos	No se llenan los campos obligatorios	E1
Datos ingresados	Los datos son validados	E2

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

incorrectos	incorrectamente	
Error en la base de datos	Al crear existe un error en la base de datos	E3

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Creación correcta	Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono Convencional, Celular, Email, Fecha de Inserción, Fecha de Inicio, Fecha de Carta de Compromiso, Fecha de Cierre, Fecha de Evaluación, Nombre de Profesión, Medio, Nombre del Proyecto, Nombre del Subproyecto	Mensaje: El nuevo Voluntario ha sido creado exitosamente
Datos obligatorios vacíos	Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono Convencional, Celular, Email, Fecha de Inserción, Fecha de Inicio, Fecha de Carta de Compromiso, Fecha de Cierre, Fecha de Evaluación, Nombre de Profesión, Medio, Nombre del Proyecto, Nombre del Subproyecto	Mensaje: Debe llenar todos los campos
Datos ingresados incorrectos	Cédula	Mensaje: Debe ingresarse un número de cédula de 10 dígitos
Error en la base de datos	Al crear Voluntario existe un error en la base de datos	Mensaje de error al intentar crear Voluntario en la base de datos

F5.2: Modificar voluntario

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Modificación correcta	Los datos son modificados correctamente	
Datos obligatorios vacíos	No se llenan los campos	E1

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	obligatorios	
Datos ingresados incorrectos	Los datos son validados incorrectamente	E2
Error en la base de datos	Al modificar existe un error en la base de datos	E3

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Modificación correcta	Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono Convencional, Celular, Email, Fecha de Inserción, Fecha de Inicio, Fecha de Carta de Compromiso, Fecha de Cierre, Fecha de Evaluación, Nombre de Profesión, Medio, Nombre del Proyecto, Nombre del Subproyecto	Mensaje: El nuevo Voluntario ha sido creado exitosamente
Datos obligatorios vacíos	Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono Convencional, Celular, Email, Fecha de Inserción, Fecha de Inicio, Fecha de Carta de Compromiso, Fecha de Cierre, Fecha de Evaluación, Nombre de Profesión, Medio, Nombre del Proyecto, Nombre del Subproyecto	Mensaje: Debe llenar todos los campos
Datos ingresados incorrectos	Cédula	Mensaje: Debe ingresarse un número de cédula de 10 dígitos
Error en la base de datos	Al modificar Voluntario existe un error en la base de datos	Mensaje de error al intentar modificar Voluntario en la base de datos

F5.3: Consultar voluntario

F5.3.1: Consulta general

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Error en la base de datos	Al consultar existe un error en la base de datos	E1

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Error en la base de datos	Al consultar Voluntario existe un error en la base de datos	Mensaje de error al intentar consultar Voluntario en la base de datos

F5.3.2: Consulta por parámetros

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Parámetros vacíos	No se llena al menos un parámetro	E1
Error en la base de datos	Al consultar existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Parámetros vacíos	Nombre del Proyecto, Nombre del Subproyecto, Cédula, Nombres, Apellidos, Fecha de Inserción, Fecha de Carta de Compromiso, Fecha de Evaluación, Profesión	Mensaje: Debe llenar al menos un parámetro
Error en la base de datos	Al consultar Voluntario existe un error en la base de datos	Mensaje de error al intentar consultar Voluntario en la base de datos

F5.4: Eliminar voluntario

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Eliminación correcta	Los datos son eliminados correctamente	
Error en la base de datos	Al eliminar existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Eliminación correcta	Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono Convencional, Celular, Email, Fecha de Inserción, Fecha de Inicio, Fecha de Carta de Compromiso, Fecha de Cierre, Fecha de Evaluación, Nombre de Profesión, Medio, Nombre del Proyecto, Nombre del Subproyecto	Mensaje: El Voluntario ha sido eliminado exitosamente
Error en la base de datos	Al eliminar Voluntario existe un error en la base de datos	Mensaje de error al intentar eliminar Voluntario en la base de datos

F6: Administración de estudiantes

El flujo de este caso de uso es similar al de Administración de Voluntarios.

F7: Administración de profesiones

El flujo de este caso de uso es similar al de Administración de Proyectos.

F8: Administración de universidades

El flujo de este caso de uso es similar al de Administración de Proyectos.

F9: Administración de facultades

El flujo de este caso de uso es similar al de Administración de Proyectos.

F10: Administración de registros de asistencia

El flujo de este caso de uso es similar al de Administración de Voluntarios.

F11: Generación de reportes

F11.1: Reporte de participantes por fechas

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Generación de reporte correcta	Los datos son consultados correctamente	
Parámetros vacíos	No se llena al menos un parámetro	E1
Error en la base de datos	Al consultar existe un	E2

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

	error en la base de datos	
--	---------------------------	--

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Generación de reporte correcta	Nombre del Proyecto, Nombre del Subproyecto, Fechas, Días, Meses, Años	Se abre automáticamente el reporte generado en formato PDF
Parámetros vacíos	Nombre del Proyecto, Nombre del Subproyecto, Fechas, Días, Meses, Años	Mensaje: Debe llenar al menos un parámetro
Error en la base de datos	Al consultar Voluntario o Estudiante existe un error en la base de datos	Mensaje de error al intentar consultar Voluntario o Estudiante en la base de datos

F11.2: Reporte de horas de asistencia

Lista de Escenarios

Escenario de caso de uso	Descripción	Excepción
Generación de reporte correcta	Los datos son consultados correctamente	
Parámetros vacíos	No se llena al menos un parámetro	E1
Error en la base de datos	Al consultar existe un error en la base de datos	E2

Matriz de casos de prueba

Escenario	Datos	Resultado Esperado
Generación de reporte correcta	Nombre del Proyecto, Nombre del Subproyecto, Voluntario, Estudiante, Rango de Fechas	Se abre automáticamente el reporte generado en formato PDF
Parámetros vacíos	Nombre del Proyecto, Nombre del Subproyecto, Voluntario, Estudiante, Rango de Fechas	Mensaje: Debe llenar al menos un parámetro
Error en la base de datos	Al consultar Registros de Asistencia existe un error en la base de datos	Mensaje de error al intentar consultar Registros de Asistencia en la base de datos

CAPÍTULO 5.- CONCLUSIONES Y RECOMENDACIONES

Una vez finalizado el proyecto se han podido obtener las siguientes conclusiones y recomendaciones:

5.1. Conclusiones

- Se pudo culminar exitosamente el desarrollo del Sistema de Administración del Programa de Voluntariado de la Fundación Mariana de Jesús, cumpliendo íntegramente todas las necesidades expuestas por la Coordinadora del Programa de Voluntariado.
- Como esta aplicación es un prototipo funcional, se acordó con la Coordinadora del Programa de Voluntariado realizar una aplicación de escritorio, para poder probar las funcionalidades del sistema por un período de tiempo y verificar que los procesos se agilicen.
- Al haber optado por utilizar software libre para implementar este sistema, se escogió herramientas actualizadas que existen hoy en día en el mercado; como son: NetBeans IDE 8.0.2 para la parte de la codificación en el lenguaje de programación JAVA; y MySQL como motor de base de datos.
- La implementación de este sistema por módulos nos garantiza que se pueda implementar nuevas funcionalidades a la aplicación, según como vaya necesitando el Programa de Voluntariado.
- La información que se levantó del Programa de Voluntariado para definir los requerimientos del sistema permitió a la FMDJ analizar la acogida que ha tenido este programa en la comunidad y continuar promoviendo esa ayuda social.
- La aplicación del enfoque metodológico del Modelo en Cascada, permitió realizar un desarrollo rigurosamente ordenado, haciendo más fácil de entender, para la Coordinadora del Programa de Voluntariado, las etapas por las cuales pasará la aplicación. Como se trata solo de un prototipo funcional, la generación de documentos en cada etapa, permitió entender de mejor manera como es el funcionamiento.
- Al utilizar una Arquitectura de Programación en Tres Capas en el desarrollo del sistema, se pudo separar las interfaces de usuario, de los procesos que se deben realizar en la lógica de negocio y de los datos que se almacenan en la base de datos. Esta distribución agilizará el proceso de actualización de funcionalidades del sistema, ya que una vez probada la aplicación, las mejoras podrán implementarse según como sea la demanda del Programa de Voluntariado.
- La etapa de Levantamiento de Requerimientos del sistema fue fundamental y muy importante, ya que al trabajar directamente con la Coordinadora del Programa de Voluntariado, pudimos obtener claramente los requerimientos puntuales y necesarios para añadir a las funcionalidades del sistema.

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

- Elaborar una buena documentación del desarrollo del sistema valió la pena en gran parte porque, al tener todos los diagramas bien detallados y estructurados del sistema en su totalidad, permitió realizar mantenimientos o cambios más fácilmente y sin pérdidas de tiempo innecesarias.
- La definición de estándares de diseño y desarrollo facilita totalmente la codificación del sistema porque proporciona los formatos definidos que debemos seguir para elaborar el código fuente y también para crear la base de datos. Con esto bien definido podemos desarrollar un sistema de buena calidad.
- La aplicación de la librería Log4J de Java para manejar las posibles excepciones generadas por los errores que se presenten en el sistema, permitió tener un archivo externo en donde se guardan los mensajes de estas excepciones, que luego son analizadas para un futuro mantenimiento del sistema.
- La aplicación del Plan de Pruebas de Integración al momento de integrar los módulos del sistema nos indicó que primero debíamos definir las clases independientes, que luego íbamos a usar para armar los módulos de las clases dependientes, en este caso son los módulos de Voluntarios, Estudiantes y Registros de Asistencia. Una vez armado por completo el sistema, se pudo verificar que la interacción entre cada módulo sea la esperada y que su comunicación sea la correcta.
- Las Pruebas de los Casos de Uso nos sirvió para analizar algunos escenarios donde se generarían errores, que luego pudimos corregir para poder realizar un desarrollo adecuado del sistema.
- El Sistema de Administración del Programa de Voluntariado de la FMDJ, actualizó los procesos manuales que actualmente se llevan, y brindó una ayuda tecnológica en el manejo y control de la información.
- El desarrollo de esta aplicación como una aplicación de escritorio permitió que los usuarios, se sientan satisfechos con la gran velocidad de respuesta de los procesos de cada módulo.
- El hecho de haber desarrollado esta aplicación sin dependencia de comunicación externa, permitió a los usuarios trabajar de forma local sin necesidad de depender de conexión a Internet.
- Todo el proceso que se llevó a cabo para el desarrollo de este sistema, resultó costoso por la cantidad de documentación que se obtuvo, pero que a la final servirá como base para nuevos desarrollos de software de este tipo de modelo de negocio.

5.2. Recomendaciones

- Cumplir con el concepto de ciclo de vida de desarrollo de software, sin importar la metodología que se aplica, ya que el desarrollo de este sistema demuestra que al tener los pasos bien definidos, partiendo de una necesidad de resolver un problema, se puede organizar mejor todas las etapas en las que se debe trabajar, para cumplir el objetivo de poner en marcha la solución que se obtuvo.
- Implementar las validaciones de ingreso de datos automáticamente en los formularios, para que el usuario no tenga que preocuparse por considerar el formato correcto que se debe utilizar.
- Probar que los módulos de las clases independientes funcionen correctamente, para luego poder acoplarlos con los módulos de las clases dependientes y así evitar que aparezcan errores de interacción entre ellos.
- Definir muy claramente los modelos de la interfaz gráfica de la aplicación con la Coordinadora del Programa de Voluntariado, para que sea simple de interpretar y muy fácil de utilizarla.
- Como Administrador del Sistema, verificar que la máquina donde va a ser instalada la aplicación y los programas necesarios para que la aplicación funcione, cumplan con los requerimientos de espacio analizados anteriormente, para asegurar que el funcionamiento sea el ideal.
- Los nuevos módulos que se podrían aumentar al sistema son:
 - _ Generación de cartas de compromiso del participante al ingresar a algún proyecto del Programa de Voluntariado.
 - _ Generación de reportes de evaluación del participante, cuando finaliza su participación.
 - _ Programas de Voluntariado Empresarial, para administrar los grupos de gente de diferentes empresas que brindan ayuda social a la FMDJ.
- Al ser esta aplicación un prototipo funcional, la base de datos completa solo la tendrá la máquina de la Coordinadora del Programa de Voluntariado, por lo tanto, al aumentar los usuarios que utilicen la aplicación, la expectativa de la Coordinadora es centralizar una base de datos en donde puedan trabajar otros usuarios de otras máquinas.
- Con esta aplicación se podrá probar los procesos que antes se administraban de forma manual, de una forma automatizada. Se debe impulsar el uso del sistema, primero analizando el tiempo que se ahorra en ingresar datos de los participantes directamente en la aplicación, también en las consultas de información referente a los proyectos y en la generación de reportes que utilizan estos datos. Luego, se debe capacitar a las personas encargadas de administrar esta información, para que se familiaricen con la aplicación, y se den cuenta que agiliza mucho más este proceso. Una vez que se tenga más confianza en el uso del aplicativo, se puede comenzar a aumentar el número de máquinas que tenga instalado el sistema y motivar a la gente a utilizarlas.

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

- Los datos que ya existen archivados en documentos físicos o en plantillas de Excel deberán ser ingresadas directamente en la aplicación, para poder actualizar la base de datos con la que podrán trabajar.
- Con la base de datos actualizada, se debe hacer un respaldo completo principal para poder almacenarlo en caso de pérdida de información; y, además un respaldo secundario en el que se trabajará con la aplicación. Cada semana los viernes se deberá añadir los nuevos cambios de la base de datos al respaldo principal, y se continuará trabajando en el respaldo secundario. Este proceso será realizado por el Administrador del Sistema, utilizando las mismas herramientas que proporciona MySQL.
- Solicitar un mantenimiento del sistema periódicamente por posibles fallas que hayan quedado sin solventar, o por errores puntuales que se los pueda apreciar en el archivo externo de logs que genera el sistema. El mantenimiento se lo realizará una vez cada mes, a menos que existan errores que no permitan seguir utilización la aplicación, en este caso será inmediato. El mantenimiento lo realizará el Administrador del Sistema y consistirá en el análisis del funcionamiento de la aplicación y la depuración de cada uno de los errores que estén registrados en el archivo de logs.
- Si los servicios de voluntariado que ofrece la FMDJ crecen, se debe diseñar este sistema de Administración del Programa de Voluntariado como una aplicación Web. Esta versión 2.0 permitirá la utilización del sistema desde diferentes terminales de cualquier parte del País, y se podrá centralizar una sola base de datos, en donde se administrará la información de cualquier terminal.
- Se recomienda a la Pontificia Universidad Católica del Ecuador, en específico a la Facultad de Ingeniería, Escuela de Sistemas, actualizar y aumentar las materias referentes a Ingeniería de Software y Programación en diferentes lenguajes, ya que en el medio profesional actual donde nos desenvolvemos, la tecnología se actualiza cada vez más rápido y los procesos de las empresas son automatizados en su gran mayoría.

TRABAJOS CITADOS

- 1: Fundación Mariana de Jesús, “Voluntariado”, [en línea]. [2015]. Disponible en la Web: <http://www.fmdj.org/fmdj/index.php/quienes-somos/plan-estrategico>
- 2: Fundación Mariana de Jesús, “Voluntariado”, [en línea]. [2015]. Disponible en la Web: <http://www.fmdj.org/fmdj/index.php/desarrollo-social/programas-fmj>
- 3: El sistema operativo GNU, “¿Qué es el software libre?”, [en línea]. [2015]. Disponible en la Web: <https://www.gnu.org/philosophy/free-sw.es.html>
- 4: gestiweb, “Libertades del Software libre”, [en línea]. [2012]. Disponible en la Web: <http://gestiweb.com/?q=content/33-libertades-del-software-libre>
- 5: Libertades del software libre, “Características, ventajas y desventajas del software libre”, [en línea]. [2015]. Disponible en la Web: <http://softwarelibrelibertades.wordpress.com/2013/03/15/caracteristicasventajas-y-desventajas-del-software-libre/>
- 6: Wikipedia Foundation Inc, “Programación orientada a objetos”, [en línea]. [2015]. Disponible en la Web: http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos
- 7: Ciberaula, “POO (Programación Orientada a objetos)”, [en línea]. [2015]. Disponible en la Web: http://java.ciberaula.com/articulo/tecnologia_orientada_objetos/
- 8: eHow en Español, “Ventajas y desventajas de la programación orientada a objetos”, [en línea]. [2015]. Disponible en la Web: http://www.ehowenespanol.com/ventajas-desventajas-programacion-orientada-objetos-info_184901/
- 9: PRESSMAN, Roger: “Ingeniería del Software: Un enfoque práctico”, 3ª Edición
- 10: RUBBY CASALLAS, Andrés Yie. “Ingeniería de Software: Ciclos de vida y Metodologías. Presentado en la Universidad de los Andes, Facultad de Ingeniería, Departamento de Sistemas y Computación”, [en línea]. [2015]. Disponible en la Web: <https://sistemas.uniandes.edu.co/~isis2603/dokuwiki/lib/exe/fetch.php?media=principal:isis2603-modelosciclosdevida.pdf>
- 11: academia.edu, “METODO EN CASCADA”, [en línea]. [2015]. Disponible en la Web: http://www.academia.edu/6362716/METODO_EN_CASCADA
- 12: LUCAS CHÁVEZ, Geomara Juliana y LOOR ALTAMIRANO, Yandri Víctor. Sistema informático de inventario y facturación de mercadería con entorno web en la Imprenta y Gráficas Chone. Calceta: s.n., 2013. Presentada en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López para obtención del título de Ingeniero en Informática.

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ADMINISTRACIÓN DEL
PROGRAMA DE VOLUNTARIADO DE LA FUNDACIÓN MARIANA DE JESÚS

13: Universo el periódico de los universitarios, “Programación en tres capas”, [en línea]. [2015]. Disponible en la Web: http://www.uv.mx/universo/566/infgral/infgral_13.html

14: Wikipedia Foundation Inc, “Java (lenguaje de programación)”, [en línea]. [2015]. Disponible en la Web:
http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29

15: ORACLE, “Java”, [en línea]. [2015]. Disponible en la Web:
<http://www.oracle.com/es/technologies/java/features/index.html>

16: Wikipedia Foundation Inc, “MySQL”, [en línea]. [2015]. Disponible en la Web:
<http://es.wikipedia.org/wiki/MySQL>

17: Grid Morelos, “MySQL”, [en línea]. [2015]. Disponible en la Web:
<http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/MySQL.pdf>

BIBLIOGRAFÍA

- PRESSMAN, Roger S.: “Ingeniería del software: un enfoque práctico”, Editorial México McGraw-Hill, Séptima Edición, 2010.
- SOMMERVILLE, Ian: “Ingeniería de Software”, Editorial Pearson Education, 2011.
- DEÁN, Jhons: “Introducción a la programación con Java”, Editorial México D.F. McGraw-Hill Interamericana, 2009.
- DUBOIS, Paul: “MySQL”, Editorial Prentice-Hall, Edición Especial, 2005.
- KNIBERG, Hennick: “Desarrollo de software con Metodologías Ágiles”, Editorial Madrid: McGraw-Hill, 2006.
- DEITEL, Paul J.: “JAVA Como Programar”, Editorial Pearson, Novena Edición, 2012.
- LARMAN, C.: “UML y patrones: Introducción al análisis y diseño orientado a objetos”, Editorial Prentice Hall.
- NN/g Nielsen Norman Group, “10 Usability Heuristics for User Interface Design”, [en línea]. [2005]. Disponible en la Web: <http://www.nngroup.com/articles/ten-usability-heuristics/>

ANEXO 1

GLOSARIO DE TÉRMINOS TÉCNICOS Y SIGLAS

A

API

API (siglas de Application Programming Interface) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

C

Claves foráneas

Es una limitación referencial entre dos tablas de una base de datos relacional. La clave foránea identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla maestra o referenciada). Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada.

Claves primarias

Es un campo o una combinación de campos que identifica de forma única a cada fila de una tabla de una base de datos relacional. Una clave primaria comprende de esta manera una columna o conjunto de columnas. No puede haber dos filas en una tabla que tengan la misma clave primaria.

Código fuente

También llamado código base, es un texto que se ha escrito en un lenguaje de programación concreto, y que sólo puede ser leído por un experto o programador. Debe traducirse a lenguaje máquina para que pueda ser ejecutado por la computadora o a bytecode para que pueda ser ejecutado por un intérprete. Este proceso se denomina compilación.

D

Desarrollo de software

Es una estructura aplicada al desarrollo de un producto de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso. Algunos autores consideran un modelo de ciclo de vida un término más general que un determinado proceso para el desarrollo de software.

F

FMDJ

Fundación Mariana de Jesús

I

IDE

IDE (siglas de Integrated Development Environment) es una aplicación de software, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de software. Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDEs tienen auto-completado inteligente de código.

Ingeniería de software

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, la aplicación de la ingeniería al software. Integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.

Integridad referencial

Es una propiedad deseable en las bases de datos. Gracias a la integridad referencial se garantiza que una entidad (fila o registro) siempre se relacione con otras entidades válidas, es decir, que existen en la base de datos. Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas.

J

JFrames y JPanels

Son clases utilizadas en Swing (biblioteca gráfica) para generar ventanas sobre las cuales añadir distintos objetos con los que podrá interactuar o no el usuario. A diferencia de JPanel, JFrame posee algunas nociones típicas de una ventana como minimizar, cerrar, maximizar y poder moverla.

L

Login

Es el proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.

Logs

Son registros oficiales de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación. También se le considera como aquel mensaje que genera el

programador de un sistema operativo, alguna aplicación o algún proceso, en virtud del cual se muestra un evento del sistema.

O

Open Source

Es la expresión con la que se conoce al software o hardware distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

P

POO

Programación Orientada a Objetos

Programación

Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito principal de la programación es crear programas que exhiban un comportamiento deseado. El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal.

Prototipo funcional

Es un sistema que sin tener todas las funcionalidades que debe disponer en su versión final, es lo suficientemente estable como para dar a conocer de forma demasiado simple los requisitos más esenciales.

S

Swing

Es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegados y tablas.

U

UML

UML, (siglas de, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

ANEXO 2

CARTA DE AUSPICIO Y ACEPTACIÓN



Quito, Febrero 3 del 2014

Señores

CONSEJO DE ESCUELA DE SISTEMAS

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

Presente

De mi consideración

Reciba un cordial saludo de quienes hacemos Fundación Mariana de Jesús, somos una institución de derecho privado sin fines de lucro, con finalidad social o pública. Constituida legalmente el 5 de diciembre de 1939 y aprobada mediante Acuerdo Ministerial N°1734 del 20 de junio de 1950 por el Ministerio de Previsión Social y Trabajo, luego modificado mediante Acuerdo Ministerial N° 1009 del 23 de julio de 1998 por el Ministerio de Bienestar Social, que se halla vigente a la fecha. Su objetivo general es mejorar la calidad de vida de la población de menores recursos económicos facilitando su acceso a servicios que contribuyan a dignificar a la persona, la familia y la comunidad.

Actualmente estamos ejecutando el Programa de Educación Básica y Media dentro del cual tenemos el Proyecto Centro de Refuerzo Escolar Arte y Recreación CREAM, el Programa del Adulto Mayor y el Programa de Voluntariado en la zona del Distrito Metropolitano de Quito.

Por lo expuesto y en virtud del interés que manifiesta la PUCE generando soluciones encaminadas al beneficio de las ONG, me permito participarles nuestro compromiso en brindar las facilidades para que el proyecto: **DESARROLLO E IMPLEMENTACION DEL SISTEMA DE ADMINISTRACION DEL PROGRAMA DE VOLUNTARIADO DE LA FUNDACION MARIANA DE JESUS**, sea ejecutado en nuestra institución, mismo que será de gran utilidad al momento de canalizar la información requerida.

Atentamente

Gabriela Parra



Coord. Voluntariado

Fundación Mariana de Jesús