

Proyecto Fin de Carrera  
Ingeniería Industrial

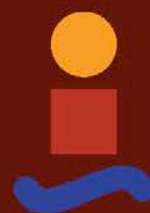
## **Gestión automatizada de voluntariado en centros hospitalarios**

Autor: María Iglesias Caso

Tutor: Jesús Racero Moreno

**Dep. Organización Industrial y Gestión de Empresas I**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2017





Proyecto Fin de Carrera  
Ingeniería Industrial

# **Gestión automatizada de voluntariado en centros hospitalarios**

Autor:

María Iglesias Caso

Tutor:

Jesús Racero Moreno

Profesor titular

Dep. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017



Trabajo Fin de Carrera: Gestión automatizada de voluntariado en centros hospitalarios

Autor: María Iglesias Caso

Tutor: Jesús Racero Moreno

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal



# Resumen

---

En muchos hospitales públicos, voluntarios de diversas asociaciones acompañan a los pacientes de las distintas especialidades para que su estancia sea lo menos traumática y más llevadera posible. Para que ello sea posible, con la calidad y seguridad necesaria, se deben coordinar a los voluntarios, para que asistan los días requeridos, y a los pacientes, pues no todos los pacientes desean o pueden recibir acompañamiento por parte de los voluntarios. La gestión eficiente de los acompañamientos requiere que cada voluntario y cada paciente esté asociado con un programa de asistencia concreto dentro de los que se realizan en el hospital pertinente.

Frente a la necesidad de los hospitales de gestionar este servicio, se propone desarrollar un sistema de información, donde destacar la importancia de la colaboración entre los voluntarios y el personal del hospital, que pueda cubrir las necesidades actuales y futuras.

El presente trabajo se encuentra estructurado en los siguientes seis capítulos:

1. **Introducción y Objetivos.** Se describen los problemas de la organización que existe actualmente en los hospitales y la manera de intentar resolverlos.
2. **Definición y Desarrollo de los Documentos de organización.** Se explica en detalle los documentos necesarios para una buena organización del voluntariado.
3. **Diseño y Desarrollo de un Sistema.** Se introduce a la metodología orientada a objetos, y se diseña y desarrollará el sistema de información colaborativo, detallando los requisitos del sistema, sus interfaces y diseñando la base de datos necesaria.
4. **Manual de Usuario.** Se explica en detalles como debe usarse la aplicación por parte de los usuarios que la utilicen.
5. **Conclusiones y futuras líneas de trabajo.** Se detalla las conclusiones y los objetivos cumplidos, y las futuras mejoras que podría tener la aplicación.





# Índice

---

<b>Resumen</b>	<b>vii</b>
<b>Índice</b>	<b>ix</b>
<b>Índice de Tablas</b>	<b>xi</b>
<b>Índice de Figuras</b>	<b>xiii</b>
<b>Notación</b>	<b>xix</b>
<b>1 Introducción y Objetivos</b>	<b>21</b>
1.1 <i>Objetivos generales</i>	22
1.2 <i>Objetivos específicos</i>	23
<b>2 Metodología de desarrollo de proyecto software</b>	<b>25</b>
2.1 <i>Fases de desarrollo de proyecto</i>	25
2.1.1 Modelo de cascada	26
2.1.2 Modelo repetitivo	26
2.1.3 Modelo en espiral	27
2.1.4 Modelo V	27
2.1.5 Modelo Big Bang	28
2.2 <i>Introducción a la metodología Orientado a Objetos (NDT)</i>	28
<b>3 Definición y Desarrollo de los Documentos de organización</b>	<b>31</b>
3.1 <i>Concepto de Hoja de Voluntario</i>	31
3.2 <i>Concepto de Hoja de Usuarios</i>	31
3.3 <i>Concepto de Hoja de Actividad</i>	31
3.4 <i>Documentos</i>	32
3.4.1 Documento 1: Hoja de inscripción del voluntario	33
3.4.2 Documento 2: Solicitud de acompañamiento al usuario	35
3.4.3 Documento 3: Hoja de actividad de voluntario	36
3.5 <i>Proceso de documentación</i>	38
<b>4 Diseño y Desarrollo de un Sistema</b>	<b>39</b>
4.1 <i>Requisitos del sistema</i>	40
4.1.1 Actores	40
4.1.2 Diagrama de paquetes	42
4.1.3 Requisitos funcionales	43
4.1.4 Diagrama de casos de uso	44
4.1.5 Prototipos de interfaces de la aplicación	82
4.2 <i>Diseño de la Base de datos</i>	99
4.2.1 Modelo relacional	99
4.2.2 Diagrama clases	99
4.2.3 Descripción de las tablas	102
4.2.4 Desarrollo de la Base de datos	106
4.3 <i>Diseño de la interfaz de usuario</i>	119
4.3.1 Arquitectura del sistema	119
4.3.2 HTML	120

4.3.3	PHP	120
4.3.4	CSS	121
4.3.5	Javascript	121
4.3.6	MySQL	121
4.3.7	Librerías Javascript	122
4.4	<i>Herramientas</i>	123
4.5	<i>Sistema de Gestión colaborativo</i>	126
4.5.1	Acceder al sistema [acceso.php]	126
4.5.2	User correcto [userCorrecto.php]	127
4.5.3	Introducir calendario [interfaceAdmAgregarCalendario.php]	128
4.5.4	Introducir Voluntario [interfaceAdmIntroducirVoluntario.php]	131
4.5.5	Modificar Voluntario [interfaceAdmModificarVoluntario.php]	133
4.5.6	Visualizar Usuarios [interfaceAdmVisualizarUsuarios.php]	136
4.5.7	Introducir Usuarios [interfacePerSolicitarAsistencia.php]	138
4.5.8	Modificar Usuarios [interfacePerModificarAsistencia.php]	140
4.5.9	Visualizar Calendario [interfaceVolVisualizarCalendario.php]	142
4.5.10	Visualizar Voluntarios del día [interfaceVolVisualizarVoluntariosDia.php]	144
4.5.11	Visualizar Usuarios del día [intefaceVolVisualizarUsuariosDia.php]	145
4.5.12	Registro Actividad [interfaceVolRegistrarActividad.php]	147
4.6	<i>Implementación</i>	150
<b>5</b>	<b>Manual de Usuario</b>	<b>163</b>
5.1	<i>Instalación y configuración la base de datos</i>	163
5.1.1	Base de datos del sistema	163
5.1.2	Ficheros de la aplicación	164
5.2	<i>Instrucciones de uso</i>	165
5.2.1	Administrador	165
5.2.2	Personal	171
5.2.3	Voluntario	174
<b>6</b>	<b>Conclusiones y futuras líneas de trabajo</b>	<b>179</b>
	<b>Referencias</b>	<b>181</b>

# ÍNDICE DE TABLAS

---

Tabla 3-1 Documento 1: Hoja de inscripción del voluntario. Unidad de Donación del Tiempo Libre.	33
Tabla 3-2 Documento 1: Hoja de inscripción del voluntario. Diver.	34
Tabla 3-3 Documento 2: Solicitud de acompañamiento al usuario	35
Tabla 3-4 Documento 3: Hoja de actividad de voluntario. Unidad de Donación de Tiempo Libre	36
Tabla 3-5 Documento 3: Hoja de actividad de voluntario. Diver	37
Tabla 4-1 Descripción de un caso de uso	44
Tabla 4-2 CU User - Validación de password	45
Tabla 4-3 CU Introducir Calendario – Selecciona introducir calendario	47
Tabla 4-4 CU Introducir Calendario – Introducir actividad	48
Tabla 4-5 CU Introducir Calendario – Modificar actividad	49
Tabla 4-6 CU Introducir Voluntario – Selecciona introducir voluntario	51
Tabla 4-7 CU Introducir voluntario – Introducir nuevo voluntario	52
Tabla 4-8 CU Introducir Voluntario – Modificar nuevo voluntario	53
Tabla 4-9 CU Modificar Voluntario – Selecciona modificar voluntario	55
Tabla 4-10 CU Modifica voluntario – Busca voluntario	56
Tabla 4-11 CU Modificar Voluntario – Modificar voluntario	57
Tabla 4-12 CU Visualizar Usuario – Selecciona visualizar usuario	59
Tabla 4-13 CU Visualizar Usuario – Busca usuario	60
Tabla 4-14 CU Visualizar Usuario – Visualiza detalle usuario	60
Tabla 4-15 CU Introducir Usuario – Selecciona introducir usuario	63
Tabla 4-16 CU Introducir Usuario – Introducir nuevo usuario	64
Tabla 4-17 CU Introducir Usuario – Modificar nuevo usuario	65
Tabla 4-18 CU Modificar Usuario – Selecciona modificar usuario	67
Tabla 4-19 CU Modificar Usuario – Modificar usuario	68
Tabla 4-20 CU Visualizar Calendario – Selecciona visualizar calendario	70
Tabla 4-21 CU Visualizar Calendario – Visualiza actividad	71
Tabla 4-22 CU Visualizar Voluntarios del día – Selecciona visualizar voluntarios del día	73
Tabla 4-23 CU Visualizar Usuarios del día – Selecciona visualiza usuario día	75
Tabla 4-24 CU Visualizar Usuarios del día – Visualiza detalle usuario día	76
Tabla 4-25 CU Registro Actividad – Selecciona registro actividad	78
Tabla 4-26 CU Registro Actividad – Registro voluntario	79
Tabla 4-27 CU Registro Actividad – Registro usuario	80

Tabla 4-28 Estructura de USER	109
Tabla 4-29 Estructura de TIPO_USER	109
Tabla 4-30 Estructura de ESTADO_USER	110
Tabla 4-31 Estructura de ASOCIACION	110
Tabla 4-32 Estructura de PROGRAMA	110
Tabla 4-33 Estructura de DIA_HORA	110
Tabla 4-34 Estructura de USUARIO	110
Tabla 4-35 Estructura de ESTADO_USUARIO	111
Tabla 4-36 Estructura de ASISTENCIA_VOLUNTARIOS	111
Tabla 4-37 Estructura de COMENTARIOS_USUARIOS	111
Tabla 4-38 Estructura de ACTIVIDAD	111
Tabla 4-39 Volcado parcial de USER	113
Tabla 4-40 Volcado parcial de USUARIO	114
Tabla 4-41 Volcado completo de TIPO_USER	115
Tabla 4-42 Volcado completo de ESTADO_USER	115
Tabla 4-43 Volcado completo de ASOCIACION	115
Tabla 4-44 Volcado completo de PROGRAMA	115
Tabla 4-45 Volcado completo de DIA_HORA	116
Tabla 4-46 Volcado completo de ESTADO_USUARIO	116
Tabla 4-47 Volcado completo de ASISTENCIA_VOLUNTARIOS	117
Tabla 4-48 Volcado completo de COMENTARIOS_USUARIOS	117
Tabla 4-49 Volcado completo de ACTIVIDAD	118

# ÍNDICE DE FIGURAS

---

Figura 1-1 Proceso de recogida de datos de los usuarios	21
Figura 2-1 Modelo de cascada	26
Figura 2-2 Modelo repetitivo	26
Figura 2-3 Modelo en espiral	27
Figura 2-4 Modelo V o Validación-Verificación	27
Figura 2-5 Modelo Big Bang	28
Figura 2-6 Generación de los modelos de análisis de pruebas desde los requisitos	29
Figura 3-1 Uso de la hoja de Usuarios y de Actividad	32
Figura 3-2 Flujo de proceso de documentación	38
Figura 4-1 Diagrama de flujo	39
Figura 4-2 Actores	40
Figura 4-3 Administrador	41
Figura 4-4 Personal	41
Figura 4-5 Voluntario	41
Figura 4-6 Diagrama de paquetes del sistema	42
Figura 4-7 Subsistema User	44
Figura 4-8 DA User - Validación de password	46
Figura 4-9 Subsistema Introducir Calendario	47
Figura 4-10 DA Introducir Calendario – Selecciona introducir calendario	49
Figura 4-11 DA Introducir Calendario – Introducir actividad	50
Figura 4-12 DA Introducir Calendario – Modificar actividad	50
Figura 4-13 Subsistema Introducir Voluntario	51
Figura 4-14 DA Introducir voluntario – Selecciona introducir voluntario	53
Figura 4-15 DA Introducir Voluntario – Introducir nuevo voluntario	54
Figura 4-16 DA Introducir Voluntario – Modifica nuevo voluntario	54
Figura 4-17 Subsistema Modificar Voluntario	55
Figura 4-18 DA Modificar voluntario – Selecciona modificar voluntario	57
Figura 4-19 DA Modificar Voluntario – Busca voluntario	57
Figura 4-20 DA Modificar Voluntario – Modifica voluntario	58
Figura 4-21 Subsistema Visualizar Usuario	59
Figura 4-22 DA Visualizar Usuario – Selecciona visualizar usuario	61
Figura 4-23 DA Visualizar Usuario – Busca usuario	61

Figura 4-24 DA Visualizar Usuario – Visualiza detalle usuario	62
Figura 4-25 Subsistema Introducir Usuario	63
Figura 4-26 DA Introducir Usuario – Selecciona introducir usuario	65
Figura 4-27 DA Introducir Usuario – Introducir nuevo usuario	66
Figura 4-28 DA Introducir Usuario – Modificar nuevo usuario	66
Figura 4-29 Subsistema Modificar Usuario	67
Figura 4-30 DA Modificar Usuario – Selecciona modificar usuario	68
Figura 4-31 DA Modificar Usuario – Modificar usuario	69
Figura 4-32 Subsistema Visualizar Calendario	70
Figura 4-33 DA Visualiza Calendario – Selecciona visualizar calendario	71
Figura 4-34 DA Visualiza Calendario – Visualiza actividad	72
Figura 4-35 Subsistema Visualizar Voluntarios del día	73
Figura 4-36 DA Visualizar Voluntarios del día – Selecciona visualizar voluntarios del día	74
Figura 4-37 Subsistema Visualizar Usuarios del día	75
Figura 4-38 DA Visualizar Usuarios del día – Selecciona visualiza usuario día	76
Figura 4-39 DA Visualizar Usuarios del día – Visualiza detalle usuario día	77
Figura 4-40 Subsistema Registro Actividad	78
Figura 4-41 DA Registro Actividad – Selecciona registro actividad	80
Figura 4-42 DA Registro Actividad – Registro voluntario	81
Figura 4-43 DA Registro Actividad – Registro usuario	81
Figura 4-44 Diagrama de navegación – Aplicación web	82
Figura 4-45 Prototipo Login	84
Figura 4-46 Prototipo Login – Error pantalla inicial	84
Figura 4-47 Prototipo Administrador	85
Figura 4-48 Prototipo Administrador – Introducir Calendario	86
Figura 4-49 Prototipo Administrador – Introducir Calendario – Detalle Introducir Calendario	86
Figura 4-50 Prototipo Administrador – Introducir Voluntario	87
Figura 4-51 Prototipo Administrador – Modificar Voluntario	88
Figura 4-52 Prototipo Administrador – Modificar Voluntario – Detalle Modificar Voluntario	88
Figura 4-53 Prototipo Administrador – Visualizar Usuario	89
Figura 4-54 Prototipo Administrador – Visualizar Usuario – Detalle Visualizar Usuario	89
Figura 4-55 Prototipo Personal	90
Figura 4-56 Prototipo Personal – Introducir Usuario	91
Figura 4-57 Prototipo Personal – Modificar Usuario	92
Figura 4-58 Prototipo Personal – Modificar Usuario – Detalle Modificar Usuario	92

Figura 4-59 Prototipo Voluntario	93
Figura 4-60 Prototipo Voluntario – Visualizar Calendario	94
Figura 4-61 Prototipo Voluntario – Visualizar Calendario – Detalle Visualizar Calendario	94
Figura 4-62 Prototipo Voluntario – Visualizar Voluntario del día	95
Figura 4-63 Prototipo Voluntario – Visualizar Usuario del día	96
Figura 4-64 Prototipo Voluntario – Visualizar Usuario del día – Detalle Visualizar Usuario del día	96
Figura 4-65 Prototipo Voluntario – Registrar actividad	97
Figura 4-66 Prototipo Voluntario – Registrar actividad – Registro voluntario	98
Figura 4-67 Prototipo Voluntario – Registrar actividad – Registro usuario	98
Figura 4-68 Modelo relacional	99
Figura 4-69 Carpetas de la aplicación web	100
Figura 4-70 Relación entre archivos	100
Figura 4-71 Diagrama de clase	101
Figura 4-72 Diagrama de clases –Base Datos	101
Figura 4-73 phpMyAdmin – Ventana principal	106
Figura 4-74 phpMyAdmin – Crear Base de Datos	106
Figura 4-75 phpMyAdmin – Crear tabla	107
Figura 4-76 phpMyAdmin – Estructura tabla	107
Figura 4-77 phpMyAdmin – Nueva tabla	107
Figura 4-78 phpMyAdmin – Modificar tabla	108
Figura 4-79 phpMyAdmin – Introducir fila	112
Figura 4-80 phpMyAdmin – Modificar fila	112
Figura 4-81 Arquitectura Cliente-Servidor	119
Figura 4-82 Ejemplo de Notepad ++	123
Figura 4-83 Ejemplo de KompoZer	124
Figura 4-84 Ejemplo de Dreamweaver CS6	125
Figura 4-85 Ejemplo de phpMyAdmin	125
Figura 4-86 Interfaz Acceso	126
Figura 4-87 Acceso.php	126
Figura 4-88 Interfaz Administrador	127
Figura 4-89 userCorrecto.php - Administrador	127
Figura 4-90 userCorrecto.php - Personal	127
Figura 4-91 userCorrecto.php - Voluntario	128
Figura 4-92 Interfaz Agregar Calendario	128
Figura 4-93 interfaceAdmAgregarCalendario.php – 1	129

Figura 4-94 interfaceAdmAgregarCalendario.php - 2	129
Figura 4-95 Interfaz Agregar Calendario - Detalle	130
Figura 4-96 interfaceAdmAgregarCalendario.php - 3	130
Figura 4-97 Interfaz Introducir Voluntario	131
Figura 4-98 interfaceAdmIntroducirVoluntario.php – 1	131
Figura 4-99 interfaceAdmIntroducirVoluntario.php – 2	132
Figura 4-100 Interfaz Modificar Voluntario	133
Figura 4-101 interfaceAdmModificarVoluntario.php - 1	133
Figura 4-102 interfaceAdmModificarVoluntario.php - 2	134
Figura 4-103 Interfaz Modificar Voluntario - Detalle	134
Figura 4-104 interfaceAdmModificarVoluntario.php - 3	135
Figura 4-105 Interfaz Visualizar Usuarios	136
Figura 4-106 interfaceAdmVisualizarUsuarios.php - 1	136
Figura 4-107 interfaceAdmVisualizarUsuarios.php - 2	137
Figura 4-108 Interfaz Visualizar Usuarios - Detalle	137
Figura 4-109 interfaceAdmVisualizarUsuarios.php - 3	138
Figura 4-110 Interfaz Introducir Usuarios	138
Figura 4-111 interfacePerSolicitarAsistencia.php – 1	139
Figura 4-112 interfacePerSolicitarAsistencia.php - 2	139
Figura 4-113 Interfaz Modificar Usuarios	140
Figura 4-114 interfacePerModificarAsistencia.php - 1	140
Figura 4-115 Interfaz Modificar Usuarios – Detalle	141
Figura 4-116 interfacePerModificarAsistencia.php - 2	141
Figura 4-117 Interfaz Visualizar Calendario	142
Figura 4-118 interfaceVolVisualizarCalendario.php - 1	142
Figura 4-119 interfaceVolVisualizarCalendario.php - 2	143
Figura 4-120 Interfaz Visualizar Calendario – Detalle	143
Figura 4-121 interfaceVolVisualizarCalendario.php - 3	143
Figura 4-122 Interfaz Visualizar Voluntarios del día	144
Figura 4-123 interfaceVolVisualizarVoluntariosDia.php	144
Figura 4-124 Interfaz Visualizar Usuarios del día	145
Figura 4-125 intefaceVolVisualizarUsuariosDia.php - 1	145
Figura 4-126 Interfaz Visualizar Usuarios del día – Detalle	146
Figura 4-127 intefaceVolVisualizarUsuariosDia.php - 2	146
Figura 4-128 Interfaz Registro Actividad	147



Figura 4-129 interfaceVolRegistrarActividad.php - 1	147
Figura 4-130 interfaceVolRegistrarActividad.php – 2	148
Figura 4-131 Interfaz Registro Actividad – Detalle Voluntarios	148
Figura 4-132 interfaceVolRegistrarActividad.php - 3	148
Figura 4-133 Interfaz Registro Actividad – Detalle Usuarios	149
Figura 4-134 interfaceVolRegistrarActividad.php - 4	149
Figura 4-135 Diagrama de clases –Base Datos	150
Figura 4-136 Diagrama de clase – userBD	152
Figura 4-137 Diagrama de clase – usuarioBD	154
Figura 4-138 Diagrama de clase – tipo_userBD	155
Figura 4-139 Diagrama de clase – estado_userBD	155
Figura 4-140 Diagrama de clase – asociacionBD	156
Figura 4-141 Diagrama de clase – programaBD	157
Figura 4-142 Diagrama de clase – dia_horaBD	158
Figura 4-143 Diagrama de clase – estado_usuarioBD	158
Figura 4-144 Diagrama de clase – actividadBD	160
Figura 4-145 Diagrama de clase – asistencia_voluntariosBD	161
Figura 4-146 Diagrama de clase – comentarios_usuariosBD	162
Figura 5-1 Instalación XAMPP	163
Figura 5-2 Archivos de la aplicación	164
Figura 5-3 Página acceso al sistema	165
Figura 5-4 Página principal de Administrador	165
Figura 5-5 Página para gestionar las actividades	166
Figura 5-6 Página para seleccionar la actividad	166
Figura 5-7 Página para gestionar a los voluntarios	168
Figura 5-8 Página para buscar y seleccionar voluntarios	168
Figura 5-9 Página para modificar a los voluntarios	169
Figura 5-10 Página para buscar y seleccionar usuarios	170
Figura 5-11 Página para buscar usuarios	170
Figura 5-12 Página principal de Personal	171
Figura 5-13 Página para solicitar un nuevo acompañamiento	172
Figura 5-14 Página para seleccionar usuarios	172
Figura 5-15 Página para modificar a los usuarios	173
Figura 5-16 Página principal de Voluntario	174
Figura 5-17 Página para visualizar las actividades	175

Figura 5-18 Página para seleccionar la actividad	175
Figura 5-19 Página para visualizar los voluntarios del día	175
Figura 5-20 Página para seleccionar usuarios	176
Figura 5-21 Página para buscar usuarios	176
Figura 5-22 Página para registrar la actividad realizada	177
Figura 5-23 Página para registrar las actividades realizadas por los voluntarios	177
Figura 5-24 Página para registrar los acompañamientos a los usuarios	178

# Notación

---

EA	Enterprise Architect
SDLC	Systems Development Life Cycle
NDT	Navigational Development Techniques
MDE	Model-Driven Engineering
QVT	Query View Transformations
DRS	Documento de Requisitos del Sistema
RA	Requisito de Almacenamiento de Información
NA	Nuevas Naturalezas
AC	Requisitos de Actores
RF	Requisitos funcionales
RNF	Requisitos no funcionales
UML	Unified Modeling Language
EA	Enterprise Architect
XAMPP	X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB, PHP,
Perl.	
SQL	Structured Query Language
PHP	Hypertext Preprocessor
HTML	HyperText Markup Language
W3C	World Wide Web Consortium
CSS	Cascading Style Sheets
XML	eXtensible Markup Language
DOM	Document Object Model
ASP	Active Server Pages
MVW	Model View WhatEver



# 1 INTRODUCCIÓN Y OBJETIVOS

En los voluntariados, especialmente en los hospitales, donde se realizan actividades que involucran a diversas asociaciones independientes, con un alto número de voluntarios, que atienden a un elevado número de usuarios en distintos programas, una buena gestión de los recursos es imprescindible. El elevado volumen de información que hay que manejar, unido a la variabilidad de requerimientos, hace que dicha gestión requiera mucho tiempo y esfuerzo.

Es necesario organizar tanto a los voluntarios como a los usuarios para que las actividades se desarrollen adecuadamente. La gestión de los voluntarios consiste principalmente en asignarles a un programa determinado, asociándolos al horario en que se llevará a cabo la actividad en la que se hayan comprometido. A su vez, la gestión de los usuarios consiste en determinar el programa del que puedan beneficiarse cuando lo soliciten.

La gestión de los voluntarios se realiza mediante formularios con los datos personales y de contacto. Al no estar informatizados, la modificación de los datos de contacto y de interés, rellenados el día de la inscripción por los voluntarios, es un trabajo engorroso debido al alto número de voluntarios que hay apuntados en los distintos programas de voluntariado presentes en el centro hospitalario.

La gestión de los usuarios se lleva a cabo sin el uso de sistemas de gestión asistidos por ordenador, mediante una lista, con la información ofrecida por el personal sanitario. Dicha información la recopila un trabajador del centro hospitalario, o un voluntario al que se le asigna dicha tarea, a medida que va visitando las diferentes plantas y alas de las especialidades del centro. Los usuarios del voluntariado pueden ingresar en cualquier momento del día y de la semana, por ello, los usuarios cuyo ingreso sea posterior a la recogida de datos por parte del trabajador o de los voluntarios quedarán fuera del sistema, no pudiendo disfrutar del beneficio que supone el acompañamiento por parte de los voluntarios.

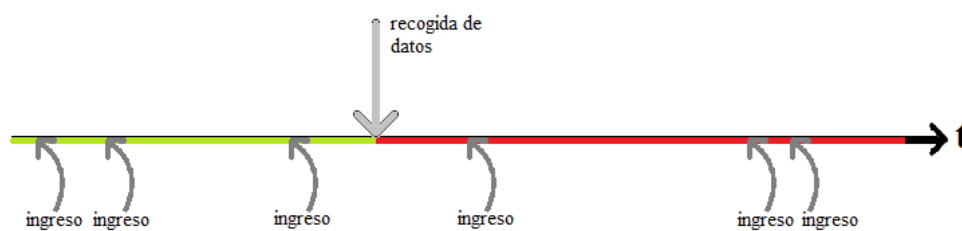


Figura 1-1 Proceso de recogida de datos de los usuarios

Actualmente, las ventajas que se pueden obtener del uso de las nuevas tecnologías, presentes en prácticamente todos los sectores y aspectos de la vida cotidiana, son indiscutibles y su incorporación debe ser una mejora en la calidad y eficiencia de la organización del voluntariado a tener muy en cuenta.

Con relación a ello y frente al número cada vez más elevado de voluntarios y usuarios que gestionar, se propone realizar un sistema de información colaborativo, entre voluntarios y personal del hospital, con el que destacar la importancia de la organización para un buen desarrollo de las actividades programadas.

Al mismo tiempo se pretende optimizar la recogida de datos de interés, tanto de los usuarios del voluntariado como de los voluntarios, y la informatización de dicha información recogida hasta el momento sólo en papel, lo cual conlleva la posibilidad de que se dupliquen los registros de usuarios y provoca una falta de flexibilidad para modificar los datos cambiantes de los voluntarios.

## 1.1 Objetivos generales

El objetivo principal de este proyecto es definir, diseñar y desarrollar un sistema de gestión colaborativa para coordinar las actividades voluntarias realizadas en los hospitales para que el trabajo realizado sea más útil y eficiente y los usuarios tengan el mejor acompañamiento posible. Actualmente para la inscripción de los voluntarios no se usa ningún tipo de sistema de gestión asistido por ordenador, sino que rellenan una hoja con sus datos personales, el programa al que quieren adscribirse y la disponibilidad de horario que tengan. El registro de usuarios se realiza manualmente recorriendo el hospital y entregando después a los voluntarios de cada programa una hoja con los datos recogidos por un trabajador del centro, los datos de los usuarios del voluntariado también pueden recopilarlos los voluntarios asignados, según el sistema que utilicen en el centro hospitalario referido. El registro de usuarios no se realiza continuamente por lo que no todos los usuarios reciben acompañamiento, debido a que los usuarios ingresan a lo largo de todo el día, todos los días. Para la difusión entre los voluntarios de las actividades no rutinarias que se realicen en los distintos programas se manda un correo informativo que puede no obtener la acogida necesaria para la buena realización de dichas actividades.

Para cumplir este objetivo principal nos centraremos en la generación automática de código, usando para ello el programa Enterprise Architect (EA). Puesto que queremos que el proyecto sea lo más versátil posible realizaremos nuestro proyecto siguiendo una metodología estándar y semiautomática de desarrollo de software que nos permita adaptar la aplicación a los distintos centros hospitalarios en función de las necesidades de cada uno.

El diseño comienza con la definición y la estructuración de un modelo que contiene la definición de los diferentes actores y la creación de un sistema para poder gestionarlo. La solución que se debe presentar consiste en un sistema de información colaborativo para poder coordinar las actividades realizadas en los distintos programas. El sistema destaca por estructurar el proceso de gestión de la manera más sencilla e intuitiva posible para que no requiera un elevado nivel de conocimiento informático para poder hacer uso de dicho sistema de información.

Se pretende poner a disposición del personal del hospital y los voluntarios un sistema de información que permita gestionar las actividades de la mejor manera posible para poder:

- **Agilizar la inscripción de voluntarios por parte de la persona encargada de la participación del voluntariado** que actualmente se realiza manualmente sin estar informatizados dichos datos.
- **Asegurar la actualización de los datos de los voluntarios** necesaria para poder asimilar los cambios de disponibilidad de horarios de los voluntarios.
- **Agilizar el registro de los usuarios por parte del personal del hospital** que actualmente se realiza manualmente.
- **Asegurar la actualización de los datos de los usuarios** necesaria debido a que la situación de los usuarios cambia continuamente, tanto por los continuos ingresos y altas como por los cambios en los estados de los usuarios.
- **Obtener mayor difusión de las actividades no rutinarias realizadas en el hospital** como formaciones para los voluntarios o actividades especiales diseñadas para los usuarios.

Para ello es necesario analizar la mejor manera de manejar toda la información para que se pueda registrar y consultar fácilmente tanto por parte del personal hospitalario como por parte de los voluntarios, haciendo que los sistemas sean intuitivos y fáciles de manejar para que se disponga de la mejor información a la hora de realizar las actividades previstas.

## 1.2 Objetivos específicos

Los objetivos generales se estructuran en objetivos parciales y específicos, donde lo principal es realizar una plataforma web para poder gestionar de manera más eficiente las actividades del voluntariado en el hospital usando una generación de código automática. Para ello se utilizará una aplicación web desde la que poder gestionar una base de datos compleja, introduciendo nuevos datos de distintos tipos, modificando los datos ya existentes en la base de datos o visualizando los datos existentes para conocer el estado de las actividades a realizar.

Para poder desarrollar esta aplicación es necesario seguir una serie de procesos durante la realización del proyecto que se pueden resumir en las siguientes actividades:

- Estudio exhaustivo del sistema de organización de los voluntarios y las actividades que estos llevan a cabo en los hospitales.
- Análisis de las herramientas software o métodos que desarrollan las empresas.
- Análisis de los diferentes tipos de acciones que se necesitan realizar para poder desarrollar el sistema de información.
- Uso de la metodología estándar y semiautomática para desarrollar el software.
- Realización del diseño de la aplicación integrada, precisando la funcionalidad de las interfaces que le afectan y especificación del comportamiento de los elementos.
- Establecimiento de los requisitos técnicos y funcionales para diseñar y desarrollar una aplicación que pueda realizar las acciones de la gestión.
- Desarrollo de la aplicación para cumplir con los requisitos técnicos y funcionales especificados en el diseño funcional y de requisitos del sistema.
- Redacción del manual necesario, tras la preparación de la aplicación, para instruir a los usuarios de la aplicación sobre los cambios que significa la ejecución del proyecto. Dichos manuales contemplarán las necesidades planteadas en la evaluación del impacto de la ejecución del proyecto.
- Realización de una prueba de la aplicación para comprobar el cumplimiento de los requisitos técnicos y funcionales exigidos.





# 2 METODOLOGÍA DE DESARROLLO DE PROYECTO SOFTWARE

---

En este capítulo se verán las diferentes metodologías que se pueden seguir para desarrollar un proyecto software, desde el planteamiento de la idea hasta su implementación.

## 2.1 Fases de desarrollo de proyecto

Existen distintos modelos para desarrollar los proyectos de software basándose en los ciclos de vida del desarrollo software (SDLC). Los diferentes ciclos de vida siguen una secuenciación estructurada y bien definida de cada etapa, aportando una serie de pasos para poder diseñar y desarrollar el producto software eficientemente.

Los distintos ciclos se basan en un esquema común que cuenta con varios pasos:

- Comunicación entre el usuario que solicita el software y el desarrollador de dicho software.
- Recogida de la mayor cantidad posible de información sobre los requisitos del sistema, para ello los desarrolladores del software se reunirán con los usuarios y estudiarán los sistemas empleados.
- Estudio de la viabilidad financiera, práctica y tecnológica para comprobar que compensa hacerse.
- Análisis del sistema para comprobar las limitaciones, problemas y cambios a realizar.
- Diseño de software usando la información de los requisitos y el análisis.
- Implementación de la programación del sistema en el lenguaje más conveniente, desarrollando programas ejecutables sin errores lo más eficientemente posible.
- Pruebas realizadas mientras se programa, evaluando los módulos, programa y producto, tanto interna como externamente para subsanar los posibles errores y obtener un producto fiable.
- Integración de bibliotecas, bases de datos y otros programas.
- Implementación mediante la instalación del software en la máquina del cliente.
- Mantenimiento y funcionamiento del software, puede requerir formación al usuario del software.
- Cierre del sistema si el software se queda obsoleto o falla en la ejecución.

A continuación, se detallan algunos de los ciclos de vida del desarrollo de software más comunes [1]:

### 2.1.1 Modelo de cascada

Es el modelo más simple en desarrollo de software, las fases se realizan de forma lineal, no pudiendo empezar una hasta que no se haya completado la anterior. Se asume que todo se lleva a cabo y tiene lugar tal y como se ha planeado anteriormente, si se deja algo de lado en la fase previa no funcionará correctamente puesto que no permite volver atrás para rehacer acciones.

Es recomendable si se han desarrollado softwares similares con anterioridad.



Figura 2-1 Modelo de cascada

### 2.1.2 Modelo repetitivo

Este modelo se basa en repetir cíclicamente cada paso después de cada ciclo del proceso. Se comienza desarrollando a menor escala, siguiendo todos los pasos, y en cada repetición se le van añadiendo más módulos y características. Con cada ciclo se obtiene un software completo con más características y capacidad que el previo.

Como el ciclo incluye pequeñas porciones de la totalidad del proceso software, es más fácil gestionar el proceso de desarrollo, pero a la vez se consumen más recursos



Figura 2-2 Modelo repetitivo

### 2.1.3 Modelo en espiral

El modelo en espiral es una combinación de los modelos repetitivo y en cascada. Comienza determinando los objetivos y limitaciones del software al inicio de cada repetición, después se crean los modelos de prototipo incluyendo el análisis de riesgos, tras esto se usa un modelo estándar de SDLC para construir el software, por último, se prepara el plan de la siguiente repetición.

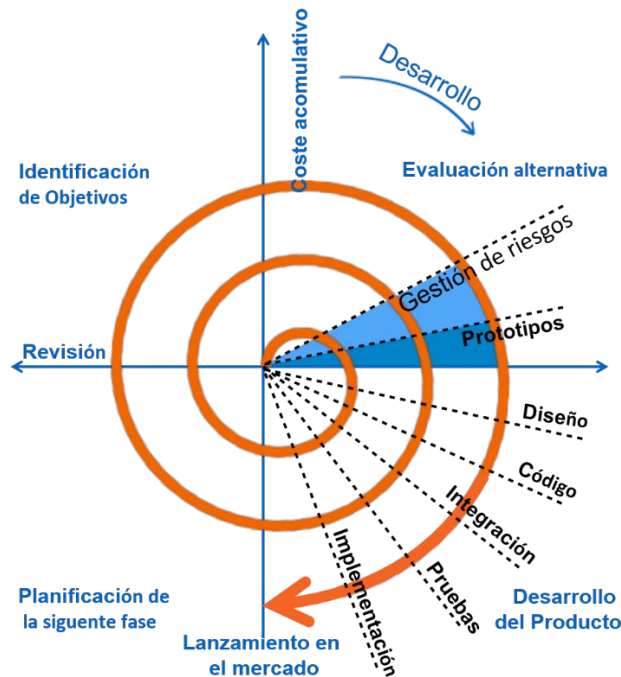


Figura 2-3 Modelo en espiral

### 2.1.4 Modelo V

El Modelo V aporta opciones de evaluación del software en cada etapa de manera inversa. En cada etapa, se planifican las pruebas y los casos de pruebas para verificar y validar el producto según los requisitos de la etapa. Más tarde, cuando el producto se desarrolla y está preparado para ser evaluado, las pruebas de caso en esta etapa verifican el software y su validez según sus requisitos. Haciendo que tanto la verificación como la validación vayan en paralelo. Este modelo también se conoce como modelo de validación y verificación.

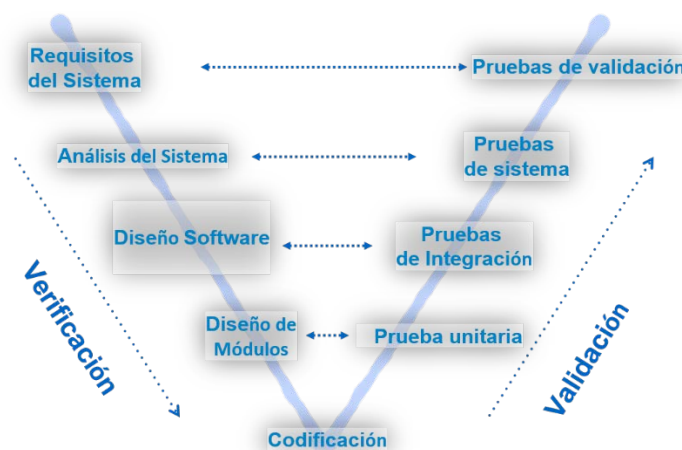


Figura 2-4 Modelo V o Validación-Verificación

### 2.1.5 Modelo Big Bang

Es el modelo con la forma más simple. Requiere poca planificación, mucha programación y muchos fondos. se conceptualiza alrededor de la teoría del 'Big Bang', si reunimos muchos fondos y programación, quizá podemos conseguir el mejor producto de software. No sigue ningún proceso concreto, y a veces el cliente no está seguro de las futuras necesidades y requisitos. Por tanto, la entrada o input respecto a los requisitos es arbitraria.

No es recomendable para grandes proyectos de software, pero es bueno para aprender y experimentar.

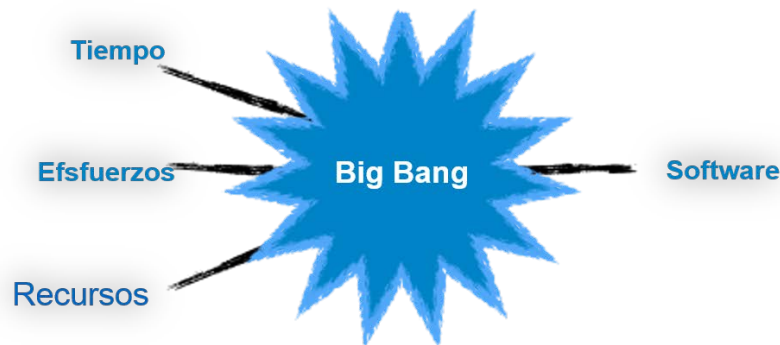


Figura 2-5 Modelo Big Bang

## 2.2 Introducción a la metodología Orientado a Objetos (NDT)

NDT (Navigational Development Techniques) es una propuesta metodológica que se engloba dentro del paradigma MDE (Model-Driven Engineering, ingeniería Guiada por Modelos) y que surgió para cubrir las necesidades navegacionales de los sistemas de información orientados a la Web. Sin embargo, actualmente la metodología NDT puede ser aplicada para especificar cualquier tipo de sistema software.

Inicialmente contemplaba únicamente un conjunto de metamodelos para el tratamiento formal de las fases de ingeniería de requisitos y análisis de un proyecto de desarrollo de software. Asimismo, también definía un conjunto de reglas de derivación (especificadas con el lenguaje formal de transformaciones QVT, Query View Transformations) que posibilitaba generar de manera sistemática todos los modelos de la fase de análisis a partir de los modelos de requisitos. Actualmente, la metodología ha evolucionado y proporciona soporte completo a todas las fases del ciclo de vida software: estudio de viabilidad, requisitos, análisis, diseño, implementación, mantenimiento y pruebas. Además, establece nuevas reglas de transformación entre cada una de estas fases.

Las principales ventajas que representa actualmente son la reutilización masiva que permiten la creación de las clases, la calidad que aporta el diseño a partir de componentes ya probados con anterioridad y pulidos varias veces, la rapidez de creación debido al uso de componente ya existentes, y la sencillez de mantenimiento por la compartimentación de las clases [2].

La siguiente figura esquematiza esta idea en la que se puede apreciar cómo se generan los modelos de la fase de análisis desde la fase de requisitos. Las reglas de transformación de NDT están representadas mediante el estereotipo «QVTTransformations», y una vez que han sido aplicadas, el equipo de analistas puede realizar transformaciones controladas con el objetivo de completar y enriquecer los modelos para obtener el modelo definitivo. Puesto que este paso no es automático requiere la experiencia del analista, están representadas mediante el estereotipo «NDTSupport» [3].

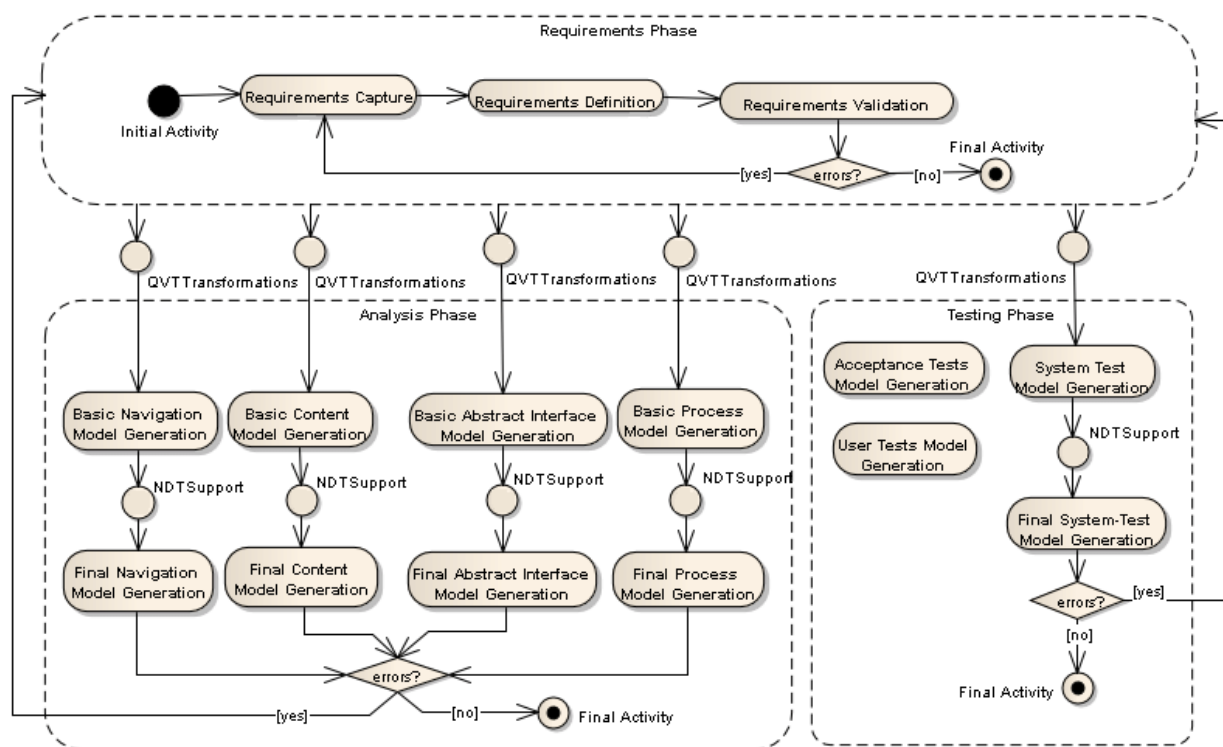


Figura 2-6 Generación de los modelos de análisis de pruebas desde los requisitos

NDT contempla la fase de requisitos como aquella fase que engloba el proceso de averiguar, normalmente en circunstancias difíciles, lo que se debe construir y que tiene por objetivo guiar el desarrollo hacia el sistema correcto mediante una descripción adecuada de los requisitos del sistema. NDT divide la fase de requisitos en tres actividades:

- **Captura:** se utilizan diferentes técnicas de captura de requisitos, como entrevistas o brainstorming, para averiguar los objetivos y requisitos del sistema.
- **Definición:** se formaliza la información capturada a partir de las entrevistas y reuniones con el cliente.
- **Validación:** se contempla la utilización de técnicas de evaluación y conciliación de requisitos.

Como resultado de la fase de requisitos, se obtiene el catálogo o Documento de Requisitos del Sistema (DRS) que debe reflejar de la manera más completa posible cada uno de los requisitos del sistema a desarrollar.

NDT propone la siguiente tipología de requisitos [4] [5]:

a) **Requisitos de información.** Este tipo de requisitos definen qué información debe almacenar y gestionar el sistema, así como las relaciones que se establecen entre dicha información. Se establecen dos tipos de requisitos de información: Requisitos de Almacenamiento de Información (RA) y Nuevas Naturalezas (NA). El concepto de NA hace referencia a las necesidades externas de información que requiere el sistema que se está especificando.

b) **Requisitos de actores (AC).** Definen los perfiles de usuario que pueden interactuar con las funciones del sistema y las relaciones que se establecen entre dichos perfiles.

c) **Requisitos Funcionales (RF).** Estos requisitos describen las necesidades de funcionalidad que debe ofrecer el sistema.

d) **Requisitos de interacción.** Estos requisitos definen las necesidades de interacción del usuario con el sistema, así como las posibilidades de navegación a través del mismo.

e) **Requisitos no funcionales(RNF).** Este tipo de requisitos se utilizan para catalogar cualquier otra necesidad del sistema que no puede ser clasificada de acuerdo con los requisitos anteriores. Pueden ser, por ejemplo, aspectos técnicos, requisitos de fiabilidad, requisitos software, requisitos de accesibilidad, etc.

Toda esta definición teórica de la metodología era impracticable en contextos empresariales debido al uso de la terminología demasiado abstracta (metamodelos, transformaciones, conceptos, etc.). Por ello, se desarrollaron herramientas software que ocultasen esta terminología teórica para mejorar la aplicabilidad de NDT en estos entornos y proyectos reales, el punto fuerte de NDT.

El grado de automatización de la metodología NDT es una de sus cualidades más relevantes gracias a que dispone de un amplio conjunto de herramientas distribuidas bajo el nombre de NDT-Suite y cuyas herramientas principales son: NDT-Profile, que implementa cada uno de los metamodelo de NDT usando perfiles UML definidos en Enterprise Architect (EA); NDT-Driver, que implementa un conjunto de procedimientos automáticos basados en las reglas de transformación entre modelos definidas de forma teórica por la metodología; y NDT-Quality, que automatiza la revisión metodológica y de trazabilidad de todos los elementos contemplados por la metodología [6].

Aplicaremos esta metodología, estructurada en diagramas y en lo que se denomina artefactos. En los siguientes apartados, conforme vayan apareciendo los conceptos asociados a esta metodología y se vayan aplicando, se irán describiendo para facilitar la lectura del proyecto [2].

# 3 DEFINICIÓN Y DESARROLLO DE LOS DOCUMENTOS DE ORGANIZACIÓN

---

En este capítulo se explicará de forma detallada los documentos necesarios para la organización, como las hojas de voluntarios, las hojas de usuarios y las hojas de actividad, y las estructuras que presentan dichos documentos.

## 3.1 Concepto de Hoja de Voluntario

Una hoja de voluntario es un documento que el voluntario que desea inscribirse en el programa de voluntariado cumplimenta con sus datos personales y de contacto y su motivación y preferencias sobre el programa y día de asistencia.

Las hojas de voluntarios son realizadas por el encargado de participación que gestiona los distintos programas de voluntariados que se realizan en el hospital según su experiencia de los datos que se requieren de los voluntarios para que puedan realizar la actividad en el programa que más se adecue con sus preferencias.

El número de datos a cumplimentar en la hoja de voluntario puede ser modificado y ampliado según las necesidades del momento.

## 3.2 Concepto de Hoja de Usuarios

Una hoja de usuarios es un documento que un trabajador del hospital o un voluntario asignado cumplimenta para que los voluntarios de los distintos programas realicen la actividad con la mejor calidad posible.

Las hojas de usuarios son realizadas por el encargado de participación que gestiona los distintos programas de voluntariados que se realizan en el hospital según su experiencia de los datos que se requieren de los usuarios para que los voluntarios puedan realizar la actividad correctamente.

El número de datos a cumplimentar en la hoja de usuarios puede ser modificado y ampliado según las necesidades del momento.

## 3.3 Concepto de Hoja de Actividad

Una hoja de actividad es un documento que el voluntario rellena una vez que ha terminado la actividad del día, según los usuarios acompañados y la actividad realizada.

Las hojas de voluntarios son realizadas por el encargado de participación que gestiona los distintos programas de voluntariados que se realizan en el hospital según su experiencia de los datos que se requieren de las actividades realizadas para que se pueda llevar un registro que ayude a mejorar las actividades realizadas en los distintos programas.

El número de datos a cumplimentar en la hoja de actividad puede ser modificado y ampliado según las necesidades del momento.

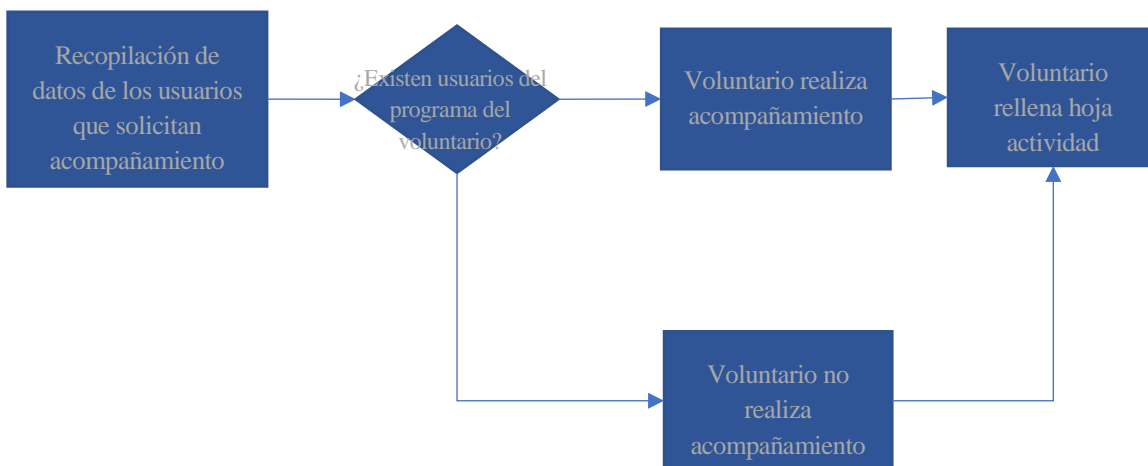


Figura 3-1 Uso de la hoja de Usuarios y de Actividad

### 3.4 Documentos

Los documentos indicados a continuación se usan para poder gestionar los distintos programas de voluntariado presentes en el hospital. Su cumplimentación se llevará a cabo por los voluntarios o el personal, según el tipo de documento a rellenar. Los datos requeridos para dichos documentos los especificarán la persona responsable de participación del hospital.

- Documento 1: Hoja de inscripción del voluntario.
- Documento 2: Solicitud de acompañamiento al usuario.
- Documento 3: Hoja de actividad de voluntario.

Según el programa de voluntariado los documentos pueden tener distintas versiones solicitando datos del voluntario, usuario o actividad distintos.

Los distintos campos de los documentos son identificados como:

- **Obligatorio:** campo obligatorio a cumplimentar en el documento, mostrado en negrita o con un asterisco a pie de página.
- **Opcional:** campo incluido para facilitar más información en el documento que puede ser dejado en blanco, mostrado con una fuente estándar.



### 3.4.1 Documento 1: Hoja de inscripción del voluntario

El documento 1 es usado para la toma de datos en la inscripción de los voluntarios que llegan al hospital para realizar la actividad. Este documento es cumplimentado por el voluntario que ha solicitado la inscripción en el programa con sus datos personales y de interés, pudiendo ser modificado o complementado por el responsable de participación del hospital.

La hoja de inscripción del voluntario cuenta con diversas variedades según el programa al que quieran apuntarse los voluntarios según sus preferencias o disponibilidad de tiempo. A continuación, se muestran dos ejemplos de hojas de inscripción:

- Hoja de inscripción en el programa Diver.
- Hoja de inscripción en el programa de Unidad de Donación de Tiempo Libre.

<b>DATOS PERSONALES:</b>			
<b>Nombre:</b>		<b>Teléfono:</b>	
<b>Apellidos:</b>		<b>Correo electrónico:</b>	
<b>Dirección:</b>		Edad:	
<b>Población:</b>		<b>Provincia:</b>	
<b>DNI:</b>		<b>CP:</b>	
<b>Fecha de nacimiento:</b>			
Fecha de inscripción:		Asociación:	
<b>DATOS ACADÉMICOS:</b>			
<b>Titulación:</b>			
Formación complementaria:			
<b>PRACTIVAS Y/O EXPETIENCIA:</b>			
<b>DISPONIBILIDAD:</b>			
<b>PREFERENCIAS DE COLABORACIÓN:</b>			

Tabla 3-1 Documento 1: Hoja de inscripción del voluntario. Unidad de Donación del Tiempo Libre.

<b>DATOS PERSONALES:</b>			
<b>Nombre:</b>		<b>Teléfono:</b>	
<b>Apellidos:</b>		<b>E-mail:</b>	
<b>Dirección:</b>		Asociación:	
<b>DNI:</b>		<b>Provincia:</b>	
Edad:		<b>CP:</b>	
	Fecha de inscripción:		
<b>¿Porque quieres hacer este voluntariado?</b>			
Prácticas y /o experiencia de voluntariado			
<b>¿Cómo nos has conocido?</b>			
<b>DISPONIBILIDAD:</b>			
Comentarios:			

Tabla 3-2 Documento 1: Hoja de inscripción del voluntario. Diver.

### 3.4.2 Documento 2: Solicitud de acompañamiento al usuario

El documento 2 es usado para la toma de datos en la solicitud de acompañamiento a los usuarios que están ingresados en el hospital. Este documento es cumplimentado por un trabajador del hospital o por un voluntario asignado que recorre las distintas especialidades del hospital apuntando los datos personales y de interés de los usuarios que han solicitado el acompañamiento en el programa.

La hoja de solicitud de acompañamiento al usuario cuenta con diversas variedades dependiendo del programa al que correspondan los usuarios. A continuación, se muestran un ejemplo de hojas de solicitud de acompañamiento:

- Hoja de solicitud de acompañamiento en el programa de Unidad de Donación de Tiempo Libre.

<b>Programa:</b>		
<b>Fecha:</b>		
<b>Nombre:</b>	<b>Observaciones:</b>	<b>Cama:</b>

Tabla 3-3 Documento 2: Solicitud de acompañamiento al usuario

### 3.4.3 Documento 3: Hoja de actividad de voluntario

El documento 3 es usado para registrar la actividad que han realizado los voluntarios de los distintos programas. Este documento es cumplimentado por los mismos voluntarios que han realizado la actividad con las observaciones que sean necesarias según el programa al que pertenezcan.

Al igual que en los otros documentos, la hoja de actividad del voluntario cuenta con diversidad variedades dependiendo del programa al que pertenezcan. A continuación, se muestran dos ejemplos de hojas de actividad de voluntario:

- Hoja de actividad de voluntario en el programa de Unidad de Donación de Tiempo Libre.
- Hoja de actividad de voluntario en el programa Diver.

<b>Programa:</b>		<b>Fecha:</b>	
<b>VOLUNTARIOS</b>			
<b>Nombre:</b>		<b>Asociación:</b>	
<b>USUARIOS</b>			
<b>Nombre:</b>		<b>Cama:</b>	<b>Comentarios:</b>

Tabla 3-4 Documento 3: Hoja de actividad de voluntario. Unidad de Donación de Tiempo Libre

FECHA	VOLUNTARIOS/AS		ENTIDAD		
NIÑOS/AS		SEXO		EDAD	HABITACION
		Hombre	Mujer		
OBSERVACIONES					

Tabla 3-5 Documento 3: Hoja de actividad de voluntario. Diver

### 3.5 Proceso de documentación

Actualmente, en los hospitales donde se realizan voluntariados, los procesos necesarios para obtener los documentos son:

1. Definición del documento.
2. Creación documento y almacenamiento en la herramienta.
3. Realización del documento.
4. Almacenamiento del documento.

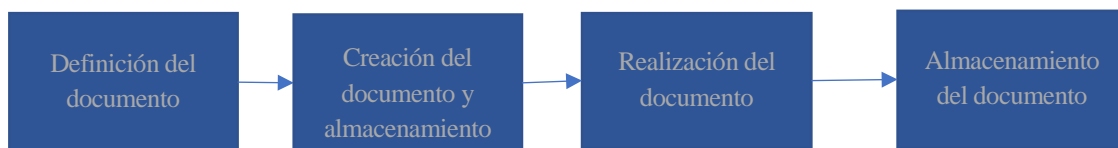


Figura 3-2 Flujo de proceso de documentación

En el siguiente apartado se detallará de manera más detallada cada una de estas etapas de del proceso de documentación:

1. El encargado de participación define el documento.

El encargado de participación del hospital, en base a su experiencia, define cuales deben ser los datos que deben de incluir los documentos para que el registro sea correcto y la actividad se pueda hacer con la calidad adecuada. Para ello se puede basar en los documentos que utilizan otros hospitales o voluntariados y realizar un documento Word donde se incluirán las tablas necesarias para que se cumplimenten con los datos requeridos.

2. El encargado de participación crea el documento y lo añade al registro.

Tras realizar el documento, se almacena en la herramienta de organización con el nombre asociado, que referenciará dicho documento.

Al realizar la ruta de organización se define una operación específica para reflejar el proceso de documentación en la que se rellenar el documento correspondiente.

3. El personal o voluntario cumplimenta el documento.

El orden de actividad refleja el documento que es necesario rellenar en cada momento. Dependiendo del tipo de documento, este será rellenado por el personal del hospital o por los voluntarios, con los datos requeridos en cada documento.

4. El encargado de participación archiva el documento.

Al finalizar el registro, el documento es archivado junto con el resto de documentos del voluntariado a la espera de que acabe el periodo de actividad. El conjunto de documentos es necesario para realizar el informe de actividad solicitado por la entidad correspondiente para justificar que la actividad se está realizando.

# 4 DISEÑO Y DESARROLLO DE UN SISTEMA

En este capítulo se definirán las hojas de datos, y se diseñará y desarrollará una aplicación que permitirá que la gestión de los voluntarios y usuarios sea más directa y flexible por la actuación de todas las partes implicadas en el sistema de gestión, consiguiendo con ello que la actividad realizada por los voluntarios se pueda cumplir más eficientemente.

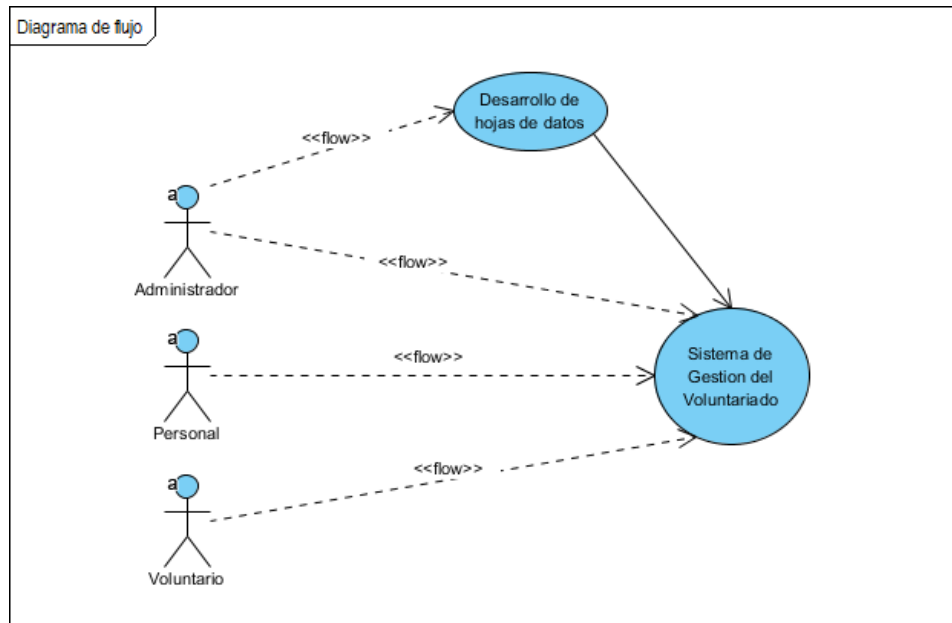


Figura 4-1 Diagrama de flujo

1. Definición del documento por el encargado de participación.

Los documentos necesarios para el registro y organización de la actividad voluntaria son estandarizados para que el sistema de gestión informático puede ser usado por todo el personal del hospital y los voluntarios, independientemente del programa al que pertenezcan. Dicha acción será llevada a cabo por el encargado de participación, incluyendo todos los datos necesarios en los documentos.

2. Realización de la actividad y cumplimentación de documentos.

Los documentos son cumplimentados por la persona pertinente tras haber realizado la acción requerida para cada documento. En el caso de las hojas de voluntario y usuario, se rellenarán con los datos necesarios para la inscripción y en el caso de la hoja de actividad, rellenada por los voluntarios de los distintos programas adscritos al hospital, se completará con los datos pertinentes tras la realización de la actividad en cuestión.

Esta aplicación permitirá al encargado de participación del hospital consultar los datos de los voluntarios y usuarios, y las actividades que han realizado, de forma sencilla e intuitiva para poder gestionar los recursos humanos y materiales de la mejor manera posible.

3. Archivo del documento.

Los documentos son archivados en una base de datos para su consulta mediante la aplicación a la que está conectada.

## 4.1 Requisitos del sistema

### 4.1.1 Actores

Los actores son empleados para especificar los roles que desempeñarán los usuarios de la aplicación o cualquier otro sistema que interactúe con el sujeto. Los actores se relacionan con el sistema mediante flujos de información, representados con líneas continuas en los diagramas de casos de usos [7].

Los tipos definidos para nuestra aplicación en el sistema son los siguientes:

- **Administrador:** Persona o grupo de personas encargadas de gestionar la configuración y administración del sistema, y de gestionar las actividades y a los voluntarios.
- **Personal:** Personas o grupo de personas encargadas de gestionar a los usuarios que reciben el acompañamiento de los voluntarios.
- **Voluntario:** Persona o grupo de personas encargadas de registrar las actividades de acompañamiento que realizan en el hospital, dichos registros se harán a posteriori.

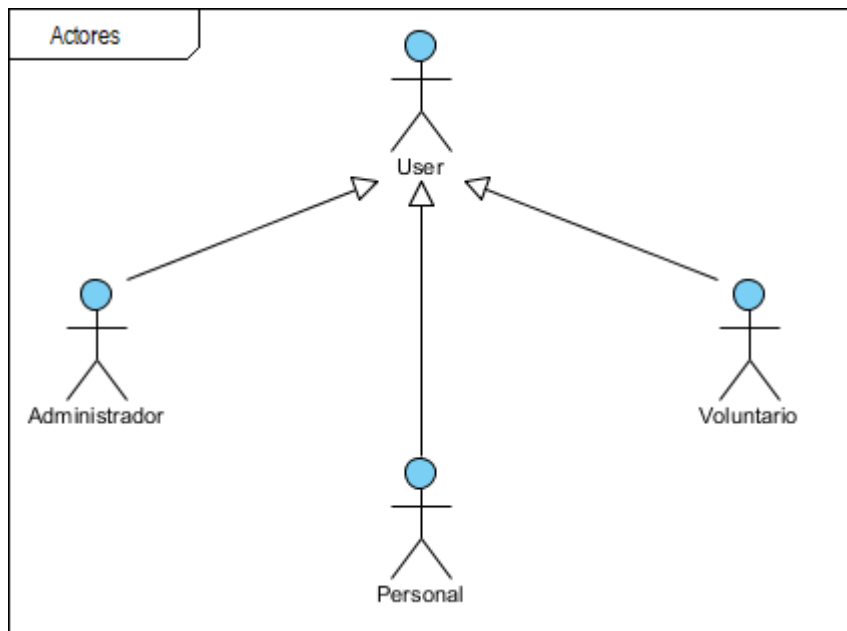


Figura 4-2 Actores

#### 4.1.1.1 Administrador

El administrador puede realizar cuatro acciones:

- Introducir actividades en el calendario, o modificar las actividades ya existentes.
- Introducir nuevos voluntarios.
- Modificar los datos de los voluntarios registrados con anterioridad.
- Visualizar los usuarios existentes que han solicitado acompañamiento.



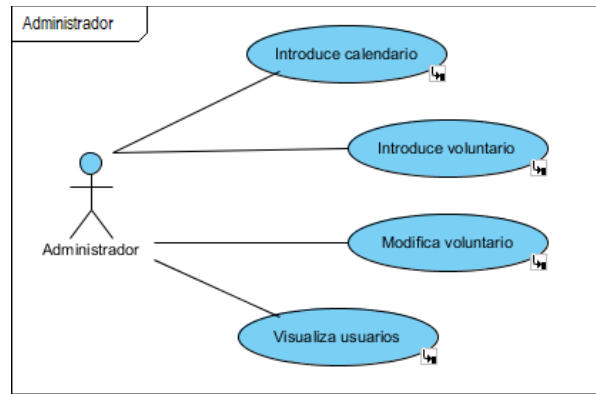


Figura 4-3 Administrador

#### 4.1.1.2 Personal

El personal puede realizar dos acciones:

- Solicitar nuevos acompañamientos a los usuarios del voluntariado.
- Modificar los datos de los acompañamientos de los usuarios ya registrados.

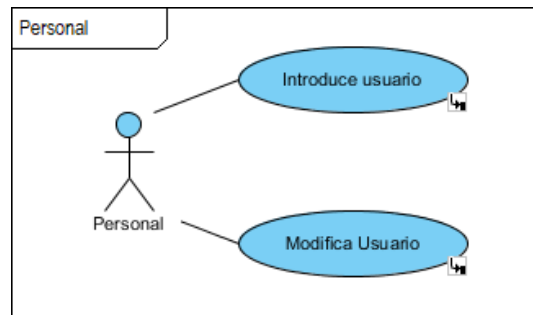


Figura 4-4 Personal

#### 4.1.1.3 Voluntario

El voluntario puede realizar cuatro acciones:

- Visualizar las actividades programadas en el calendario.
- Visualizar los datos de los voluntarios que hacen la actividad conjuntamente.
- Visualizar los datos de los usuarios que han solicitado acompañamiento.
- Registrar los acompañamientos que han realizado.

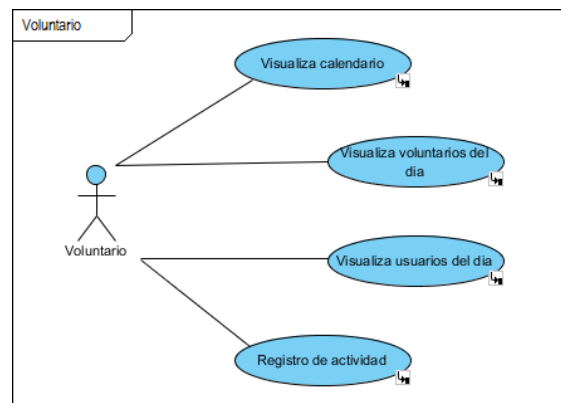


Figura 4-5 Voluntario

### 4.1.2 Diagrama de paquetes

Las relaciones entre los subsistemas se muestran en el diagrama de paquetes, donde cada subsistema tiene funcionalidades diferenciadas [2].

El subsistema o paquete **user** permite a los usuarios acceder a la aplicación, el resto de paquetes que conforman el sistema dependerán de él pues si no se tiene acceso al sistema no se puede realizar ninguna otra acción.

El subsistema **introducir calendario** permite introducir nuevas actividades al calendario o modificar las actividades ya registradas. Una vez introducidas las actividades, mediante el subsistema **visualizar calendario** se podrán ver los detalles sin modificarlos.

Los subsistemas **introducir voluntario** y **modificar voluntario** permitirán la gestión de los voluntarios del sistema, introduciendo un nuevo voluntario o modificando los voluntarios ya existentes, respectivamente.

Análogamente, los subsistemas **introducir usuario** y **modificar usuario** permitirán la gestión de los usuarios del sistema, introduciendo un nuevo usuario o modificando los usuarios ya existentes, respectivamente.

Una vez creados los voluntarios y los usuarios, desde los subsistemas **visualizar usuarios**, **visualizar usuarios del día** y **visualizar voluntarios del día** podrán verse los detalles de los voluntarios y usuarios. En el diagrama mostrado a continuación se pueden ver las relaciones que tienen.

Finalmente, el subsistema **registrar actividad** permitirá introducir en el sistema las actividades realizadas por los voluntarios y los acompañamientos que han recibido los usuarios.

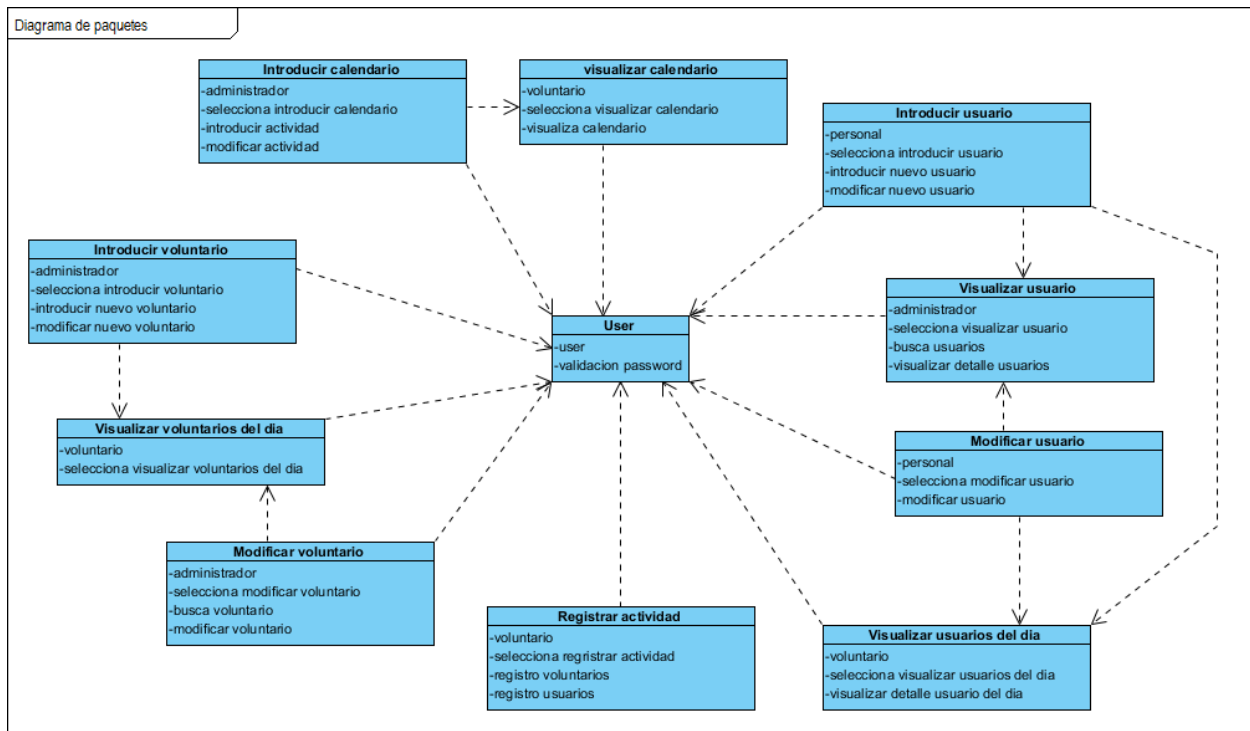


Figura 4-6 Diagrama de paquetes del sistema

### 4.1.3 Requisitos funcionales

A continuación, se establecerán los comportamientos del sistema, que han sido representados mediante los diagramas de casos de uso y el diagrama de paquetes [2].

El diagrama de paquetes muestra cómo está dividido el sistema en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Cada paquete es representado mediante un bloque y puede contener otros subsistemas o un diagrama de casos de uso. Los paquetes de un sistema están relacionados entre sí mediante relaciones de dependencia (representado mediante línea discontinua).

Los diagramas de casos de uso son diagramas de comportamiento que describen los comportamientos del sistema al afrontar una tarea o un requisito. Los principales elementos de los diagramas de caso de uso son:

- **Actores** Representan usuarios y sistemas externos al sistema en estudio. Se representan mediante un “Sticky Man” o muñeco de barras.
- **Caso de uso** Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Un caso de uso se representa mediante un óvalo.
- **Relaciones** Las relaciones son entre casos de uso y actores. Se representan mediante líneas y se clasifican en:
  - **Inclusión (<<include>>)** Es una forma de interacción o creación, un caso de uso dado puede "incluir" otro caso de uso. El primer caso de uso a menudo depende del resultado del caso de uso incluido.
  - **Extensión (<<extend>>)** Es otra forma de interacción, un caso de uso dado (la extensión) puede extender a otro. Esta relación indica que el comportamiento del caso de la extensión se puede utilizar en el caso de uso referenciado. Las relaciones entre un caso de uso y otro caso siempre deben tener extensión o inclusión.

Los diagramas de actividades modelan el flujo de procesos de un caso de uso. En conjunto con la especificación del caso de uso representan a nivel visual la dinámica la aplicación web. Los diagramas de actividades con perfil web representan las acciones que son parte de un caso de uso, así como los datos presentados al usuario y aquellos requeridos como entrada de datos pueden ser modelados con precisión como actividades.

Para describir los casos de usos se utiliza una tabla, donde se describen las principales características del proceso.

- Nombre del caso de uso (primera fila) junto con el identificador del diagrama.
- Relaciones que presenta el caso de uso ya sea con otros casos de uso o con actores (indica el origen de la relación el destino y la forma de relacionarse).
- Breve descripción del caso de uso, especificación de precondiciones previas, si existieran, para la ejecución del caso de uso y postcondiciones que indica cómo queda el sistema.
- Escenarios de ejecución que se realizan, así como las posibles alternativas si es necesario. En cada escenario se expresa un orden indicando si la tarea o acción es realizada por el sistema o bien por el usuario.

<b>Designación</b>	Nombre del caso de uso		<b>Identificación</b>	XX
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>	
	Actor o Caso de uso	Use, include o extend	Caso de uso	
<b>Descripción</b>	Descripción del caso de uso			
<b>Precondiciones</b>	Condiciones que debe tener el Sistema antes de realizar el caso de uso			
<b>Postcondiciones</b>	Condiciones en las debería quedar el sistema después de haber realizado el caso de uso.			
<b>Escenarios</b>	Escenario básico, donde se muestra los pasos que normalmente se desarrollaría en el caso de uso. Por cada escenario se incluye las tareas que se van a realizar. Por cada tarea, se especifica el orden de ejecución y si es realizada por el sistema o un usuario. A continuación, el nombre de la tarea y finalmente los resultados si hubiera.			
	<b>Pasos</b>	<b>Acción</b>	<b>Resultados</b>	

Tabla 4-1 Descripción de un caso de uso

#### 4.1.4 Diagrama de casos de uso

A continuación, se describen los procesos de los distintos subsistemas con un mayor grado de detalle mediante los diagramas de casos de uso.

##### 4.1.4.1 «Sub-Sistema» 01. Validación password

El presente subsistema especifica el comportamiento de la aplicación con respecto al acceso y seguridad del sistema. Los tipos de usuarios de la aplicación definidos en el sistema son los definidos con anterioridad:

- **Administrador:** Con acceso a los procesos relacionados con la gestión del voluntariado y calendario, y a la visualización de usuarios
- **Personal:** Con acceso a los procesos relacionados con la gestión de los usuarios.
- **Voluntario:** Con acceso a los procesos relacionados con el registro de actividades, y a la visualización de actividades, voluntarios y usuarios.

El sistema proporcionará el acceso a la aplicación a través de la identificación del “user” y “password”. El sistema realizará la validación del usuario de la aplicación o user contra la base de datos y en el caso que todas las validaciones hayan sido correctas, el sistema cargará la pantalla inicial donde aparecerá un menú con las opciones del tipo de user que se ha registrado.

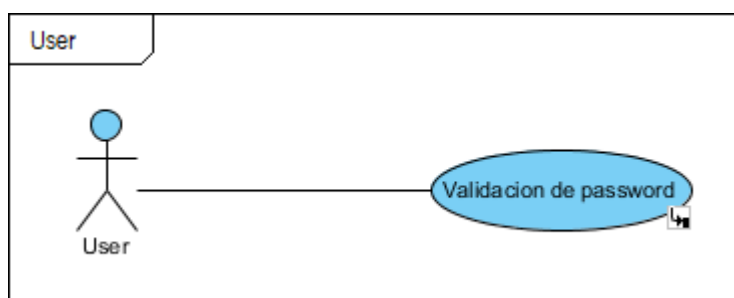


Figura 4-7 Subsistema User

<b>Designación</b>	Validación de password	<b>Identificación</b>	P1.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	User	UseCaseLink	Validación de password
<b>Actores</b>	Administrador – Personal - Voluntario		
<b>Descripción</b>	Comprobar que el user está registrado en la base y poder acceder a la aplicación		
<b>Precondiciones</b>	A. La página de inicio debe estar cargada previamente. B. Para poder acceder a la aplicación el user debe estar dado de alta en el sistema. C. La página del administrador debe tener acceso al servidor web		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(user)	Introduce la dirección de la aplicación	
	2 (sistema)	Crea el formulario de la página de inicio	Obtiene página de inicio
	3-(user)	Introduce datos de acceso (user/password)	
	4-(sistema)	Comprueba datos de acceso	El sistema realiza la validación del user contra la base de datos y accede a la aplicación si es positiva
	5-(sistema)	Presenta pantalla de error	El sistema vuelve a presentar el formulario de inicio (con un mensaje de error) al obtener la validación negativa previamente (paso 2)
	6-(sistema)	Comprueba tipo user	El sistema comprueba el tipo de user para mostrar el menú correspondiente
	7-(sistema)	Presenta pantalla user de inicio	Obtiene página de inicio del user con el menú correspondiente

Tabla 4-2 CU User - Validación de password

## a. Diagrama de actividades

El usuario de la aplicación o user introduce la dirección de la aplicación en el navegador para que el sistema pueda crear el formulario de la página de inicio en el cual meter los datos para iniciar la sesión. Después de que el user haya introducido su user y password en el formulario de acceso, el sistema comprueba dichos datos en la base de datos. Si la validación es negativa, el sistema muestra un mensaje de error en la pantalla de inicio para que vuelva a introducir los datos de acceso en el formulario. Si la validación es positiva, el sistema comprueba el tipo de user que se ha registrado y muestra la pantalla de inicio de user con el menú correspondiente.

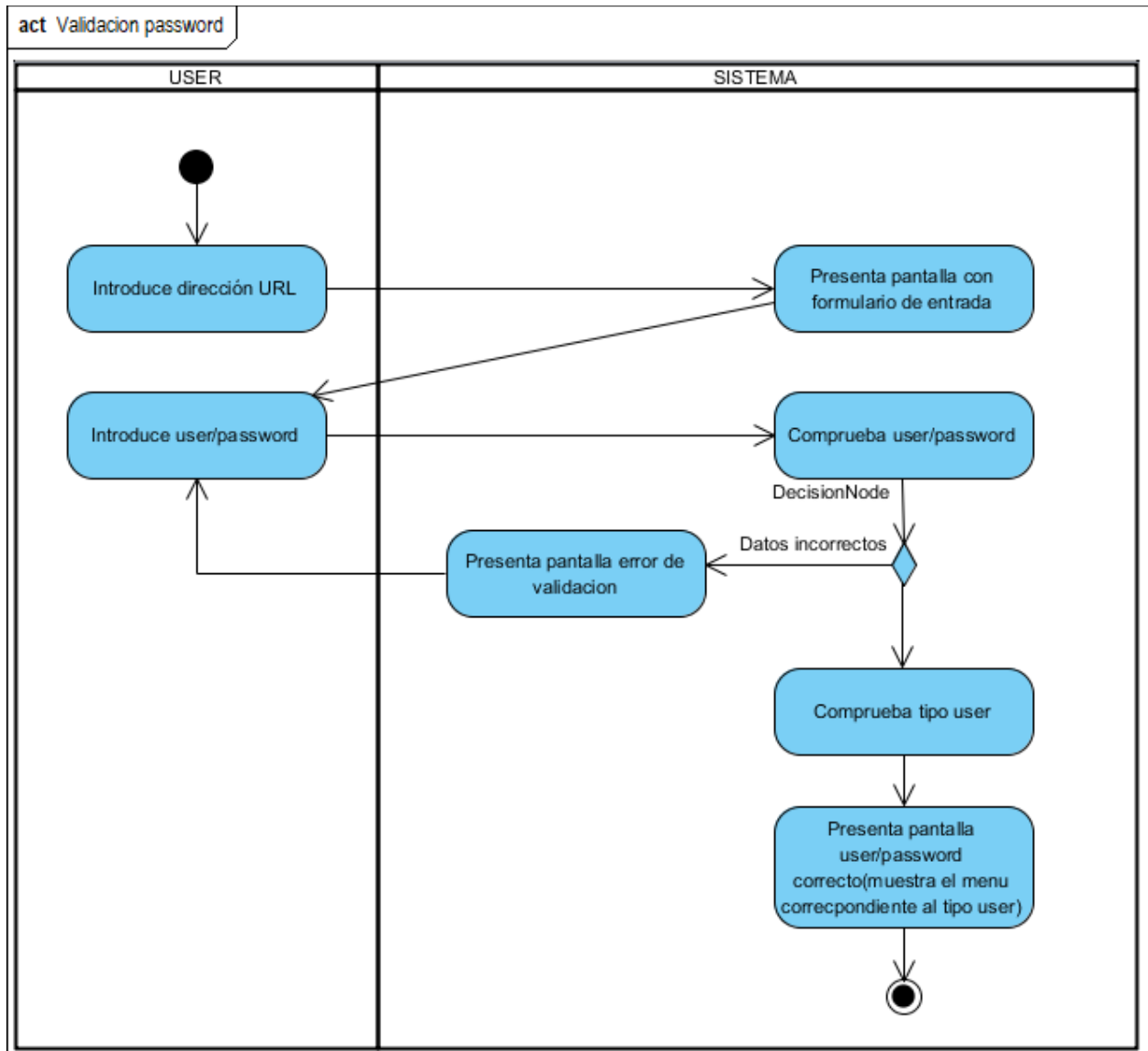


Figura 4-8 DA User - Validación de password

#### 4.1.4.2 «Sub-Sistema» 02. Introducir Calendario

El subsistema **Introducir Calendario** se encarga de gestionar las actividades en el sistema. Los usuarios de la aplicación o user del tipo Administrador serán los únicos del sistema que cuentan en su menú con la opción de gestionar las actividades del calendario. La gestión de las actividades consiste en introducir nuevas actividades al calendario o modificar las actividades ya existentes.

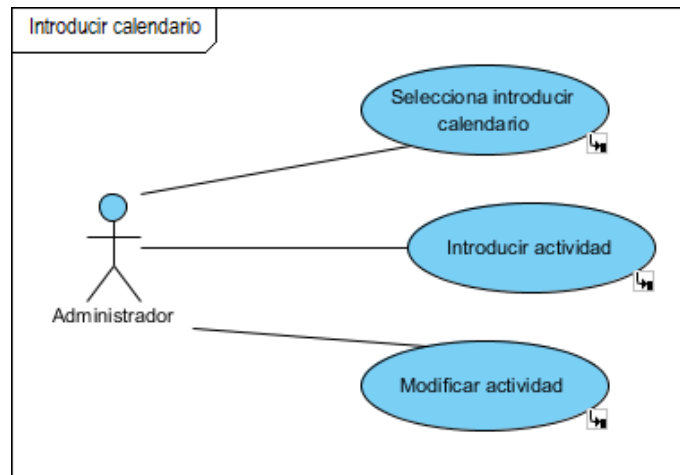


Figura 4-9 Subsistema Introducir Calendario

<b>Designación</b>	Selecciona introducir calendario	<b>Identificación</b>	P2.1.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Selecciona introducir calendario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Cuando el administrador selecciona esta opción, el sistema le permitirá gestionar las actividades del calendario		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona introducir calendario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de introducción de actividades, con una tabla para introducir datos y un calendario

Tabla 4-3 CU Introducir Calendario – Selecciona introducir calendario

<b>Designación</b>	Introducir actividad	<b>Identificación</b>	P2.1.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Introducir actividad
<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite introducir nuevas actividades en el calendario		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Introduce datos actividad	
	2 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	3-(sistema)	Mensaje datos guardados	El sistema muestra un mensaje por pantalla cuando los datos se han guardado correctamente
	4-(sistema)	Actualiza listado	El sistema actualiza el calendario

Tabla 4-4 CU Introducir Calendario – Introducir actividad

<b>Designación</b>	Modificar actividad	<b>Identificación</b>	P2.1.3
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Modificar actividad
<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite modificar las actividades del calendario		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona un día del calendario	
	2 (sistema)	Abre nueva ventana con las actividades	El sistema abre una nueva ventana con un listado de las actividades registradas en el calendario el día seleccionado



	3-(administrador)	Selecciona actividad del listado	
	4-(sistema)	Cierra ventana y rellena tabla de datos	El sistema cierra la ventana con el listado de actividades y rellena la tabla de datos de la página principal
	5-(administrador)	Modifica los datos de la actividad	
	6 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	7-(sistema)	Mensaje datos guardados	El sistema muestra un mensaje por pantalla cuando los datos se han guardado correctamente
	8-(sistema)	Actualiza listado	El sistema actualiza el calendario

Tabla 4-5 CU Introducir Calendario – Modificar actividad

#### a. Diagrama de actividades

Cuando el user de tipo Administrador ha accedido a la aplicación cuenta en su menú con la opción de Introducir calendario, al seleccionarla el sistema muestra en pantalla una tabla con los campos para introducir o modificar los datos de la actividad y un calendario donde estarán resaltados los días en los que hay alguna actividad inscrita. Desde dicha pantalla puede añadir una nueva actividad o modificar una ya existente seleccionando el día al que corresponde.

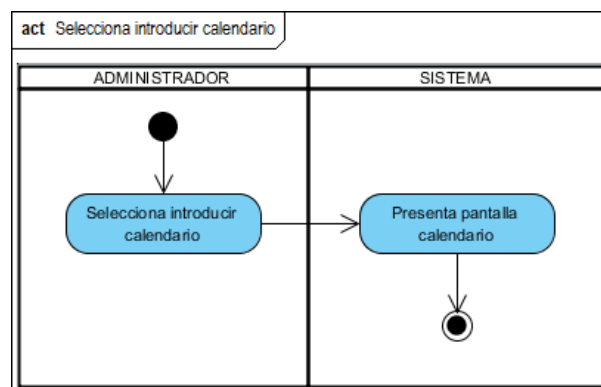


Figura 4-10 DA Introducir Calendario – Selecciona introducir calendario

Si el administrador decide introducir una nueva actividad deberá rellenar la tabla con los datos pertinentes, una vez completado el formulario el sistema guardará los nuevos datos en la base de datos al pulsar el botón 'guardar', mostrando un mensaje para asegurar que se han guardado correctamente tras lo cual se actualizará el listado de actividades.

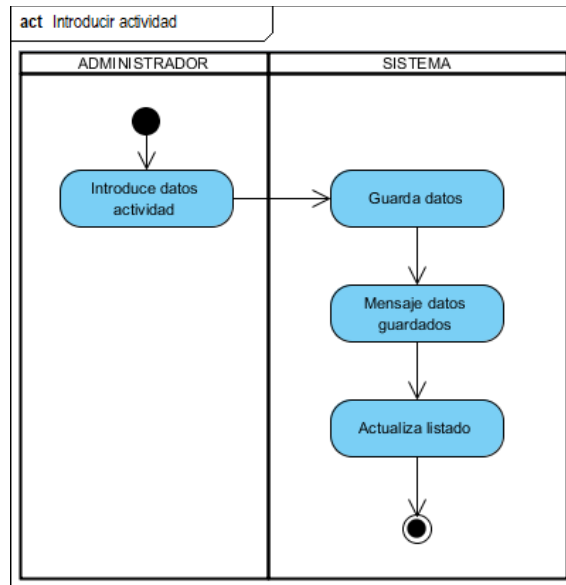


Figura 4-11 DA Introducir Calendario – Introducir actividad

Si, por el contrario, el administrador decide modificar una actividad ya existente, seleccionará un día del calendario de los que están resaltados, ya que esos días son los que tienen las actividades ya registradas. Al seleccionar un día del calendario el sistema abrirá una nueva ventana en la que aparecerá un listado con las actividades programadas para ese día. Para poder modificar los datos de la actividad, el administrador seleccionará la actividad de la lista con lo que el sistema cerrará la ventana y rellenará la tabla con los datos de la actividad seleccionada. Una vez que la tabla esté rellena, el administrador podrá modificar los datos que necesite en el formulario y el sistema los guardará en la base de datos al pulsar el botón 'guardar', mostrando un mensaje para asegurar que se han guardado correctamente tras lo cual se actualizará el listado de actividades.

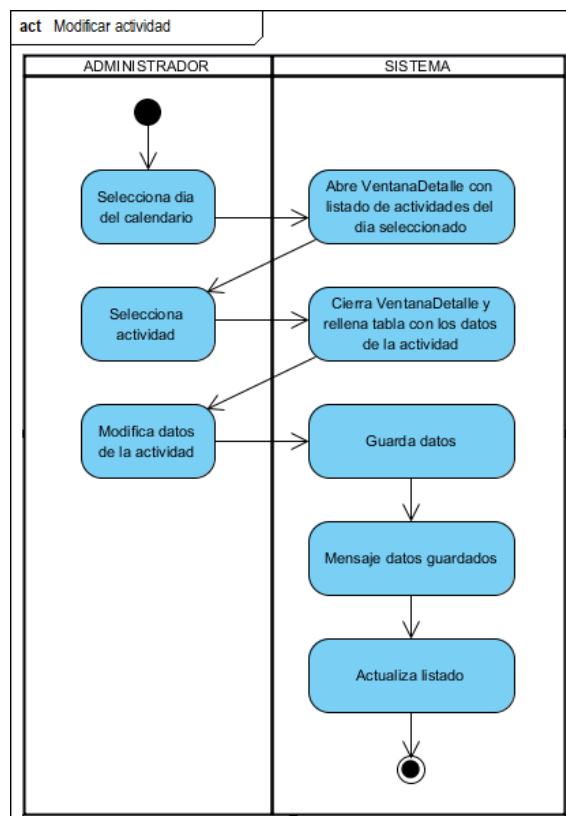


Figura 4-12 DA Introducir Calendario – Modificar actividad

#### 4.1.4.3 «Sub-Sistema» 03. Introducir Voluntario

El subsistema **Introducir Voluntario** se encarga de gestionar los voluntarios en el sistema. Los usuarios de la aplicación o user del tipo Administrador serán los únicos del sistema que cuentan en su menú con la opción de gestionar los voluntarios. La gestión de los voluntarios consiste en introducir nuevos voluntarios en el sistema o modificar los voluntarios ya existentes.

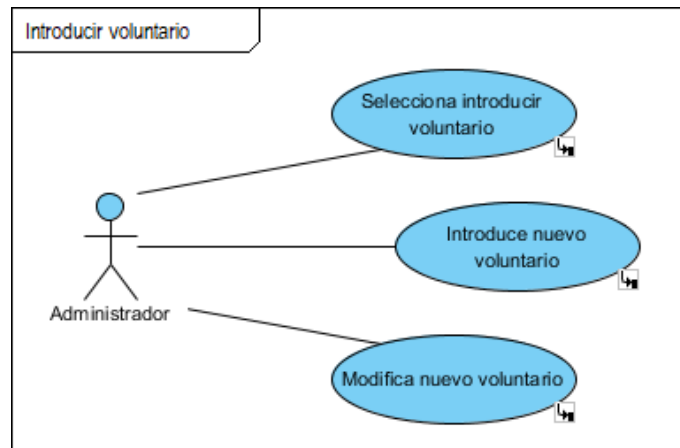


Figura 4-13 Subsistema Introducir Voluntario

<b>Designación</b>	Selecciona introducir voluntario	<b>Identificación</b>	P2.2.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Selecciona introducir voluntario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Cuando el administrador selecciona esta opción, el sistema le permitirá gestionar los voluntarios		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona introducir voluntario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de introducción voluntario, con una celda para buscar y una tabla para introducir datos

Tabla 4-6 CU Introducir Voluntario – Selecciona introducir voluntario

<b>Designación</b>	Introducir nuevo voluntario	<b>Identificación</b>	P2.2.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Introducir nuevo voluntario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite introducir nuevos voluntarios en el sistema		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Introduce datos del voluntario	
	2 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	3-(sistema)	Mensaje datos guardados	El sistema muestra un mensaje por pantalla cuando los datos se han guardado correctamente
	4-(sistema)	Actualiza listado	El sistema actualiza los voluntarios

Tabla 4-7 CU Introducir voluntario – Introducir nuevo voluntario

<b>Designación</b>	Modificar nuevo voluntario	<b>Identificación</b>	P2.2.3
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Modificar nuevo voluntario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite modificar los voluntarios		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Introduce DNI	
	2 (sistema)	Comprueba DNI	El sistema comprueba si existe algún voluntario con el DNI introducido
	3-(sistema)	Rellena tabla	El sistema rellena la tabla con los datos correspondientes al DNI introducido

	4-(administrador)	Modifica datos del voluntario	
	5 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	6-(sistema)	Mensaje datos guardados	El sistema muestra un mensaje por pantalla cuando los datos se han guardado correctamente
	7-(sistema)	Actualiza listado	El sistema actualiza los voluntarios

Tabla 4-8 CU Introducir Voluntario – Modificar nuevo voluntario

### a. Diagrama de actividades

Cuando el user de tipo Administrador ha accedido a la aplicación cuenta en su menú con la opción de Introducir voluntario, al seleccionarla el sistema muestra en pantalla una casilla para introducir el DNI y una tabla donde aparecerán los datos del voluntario. Desde dicha pantalla puede buscar si el voluntario que quiere introducir ya existe o introducirlo directamente.

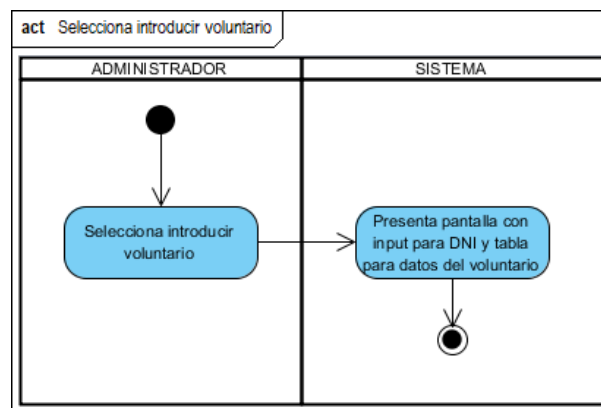


Figura 4-14 DA Introducir voluntario – Selecciona introducir voluntario

Si el administrador decide introducir un nuevo voluntario directamente deberá rellenar la tabla con los datos pertinentes, una vez completado el formulario el sistema guardará los nuevos datos en la base de datos al pulsar el botón 'guardar', mostrando un mensaje para asegurar que se han guardado correctamente tras lo cual se actualizará el listado de voluntarios.

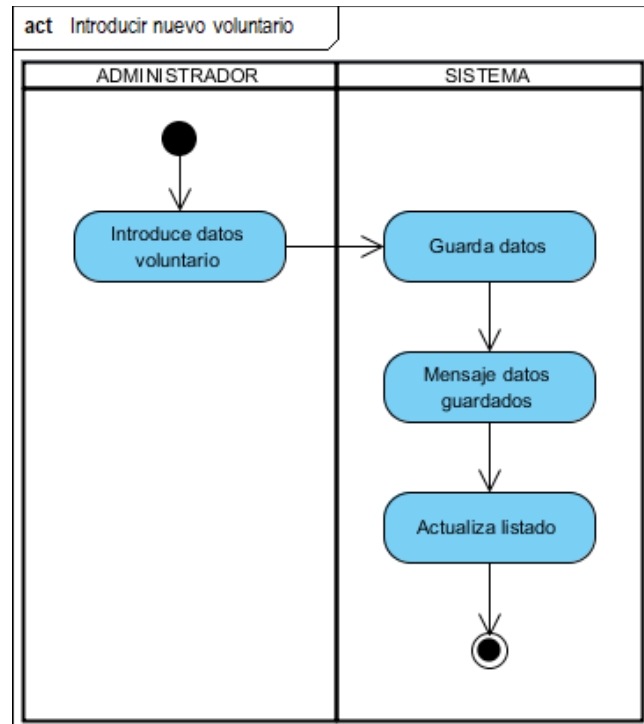


Figura 4-15 DA Introducir Voluntario – Introducir nuevo voluntario

Por otro lado, el administrador puede buscar primero si el voluntario ya existe en la base de datos para que el sistema no de error al duplicar los datos. Para ello, introducirá el DNI del voluntario en la casilla correspondiente, al darle al botón 'buscar' el sistema buscará en la base de datos si existe algún voluntario que corresponda con el DNI introducido y rellenará la tabla con los datos del voluntario. Una vez que la tabla esté rellena, el administrador podrá modificar los datos que necesite en el formulario y el sistema los guardará en la base de datos al pulsar el botón 'guardar', mostrando un mensaje para asegurar que se han guardado correctamente tras lo cual se actualizará el listado de voluntarios.

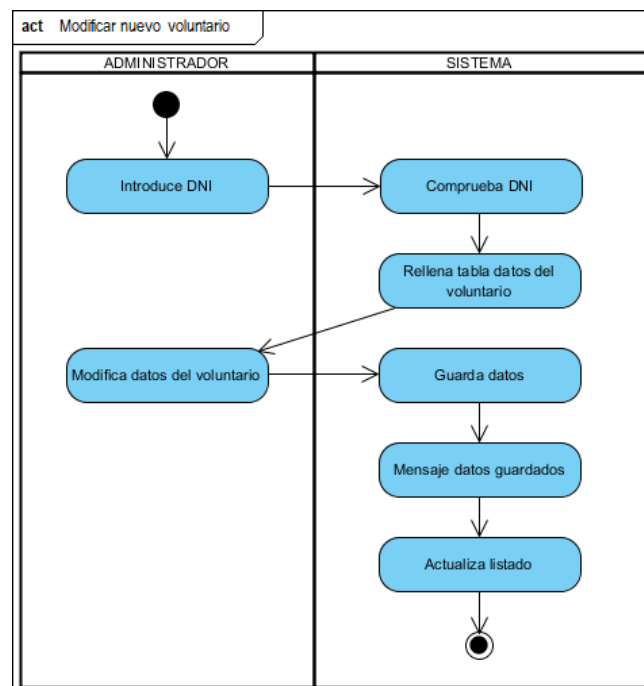


Figura 4-16 DA Introducir Voluntario – Modifica nuevo voluntario

## 4.1.4.4 «Sub-Sistema» 04. Modificar Voluntario

El subsistema **Modificar Voluntario** se encarga de gestionar los voluntarios en el sistema. Los usuarios de la aplicación o user del tipo Administrador serán los únicos del sistema que cuentan en su menú con la opción de gestionar los voluntarios. La gestión de los voluntarios consiste en buscar al voluntario que se desee modificar y modificarlo.

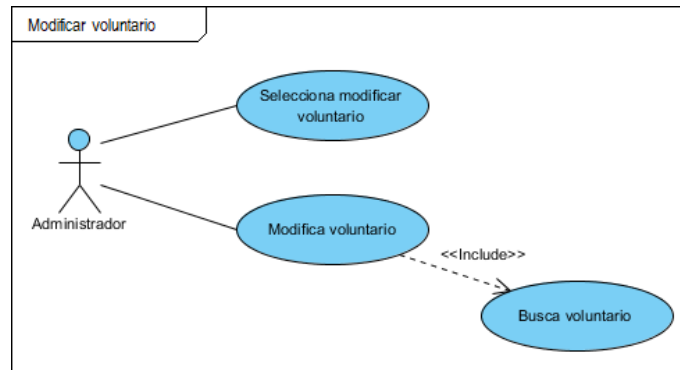


Figura 4-17 Subsistema Modificar Voluntario

<b>Designación</b>	Selecciona modificar voluntario	<b>Identificación</b>	P2.3.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Selecciona modificar voluntario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Cuando el administrador selecciona esta opción, el sistema le permitirá gestionar los voluntarios		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona modificar voluntario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de modificar voluntario, con una tabla para buscar y otra para seleccionar el voluntario

Tabla 4-9 CU Modificar Voluntario – Selecciona modificar voluntario

<b>Designación</b>	Busca voluntario	<b>Identificación</b>	P2.3.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Modifica voluntario	Include	Busca voluntario

<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite buscar los voluntarios		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Introduce datos para buscar	
	2 (sistema)	Busca voluntarios	El sistema busca los voluntarios que coinciden con los datos introducidos
	3-(sistema)	Muestra listado de voluntarios que coinciden	El sistema muestra un listado con los voluntarios que coinciden con la búsqueda

Tabla 4-10 CU Modifica voluntario – Busca voluntario

<b>Designación</b>	Modifica voluntario	<b>Identificación</b>	P2.3.3
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Modifica voluntario
	Modifica voluntario	Include	Busca voluntario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite modificar los voluntarios		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador D. El administrador ha realizado primero una búsqueda de voluntarios		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona voluntario	
	2 (sistema)	Abre VentanaDetalle con tabla de datos	El sistema abre una VentanaDetalle con una tabla con los datos del voluntario seleccionado
	3-(administrador)	Modifica datos del voluntario	
	4 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos



	5-(sistema)	Cierra VentanaDetalle	El sistema cierra la VentanaDetalle
	6-(sistema)	Actualiza listado	El sistema actualiza los voluntarios

Tabla 4-11 CU Modificar Voluntario – Modificar voluntario

### a. Diagrama de actividades

Cuando el user de tipo Administrador ha accedido a la aplicación cuenta en su menú con la opción de Modificar voluntario, al seleccionarla el sistema muestra una pantalla con una tabla para introducir los campos de búsqueda del voluntario y un listado con los voluntarios que corresponden con esos campos.

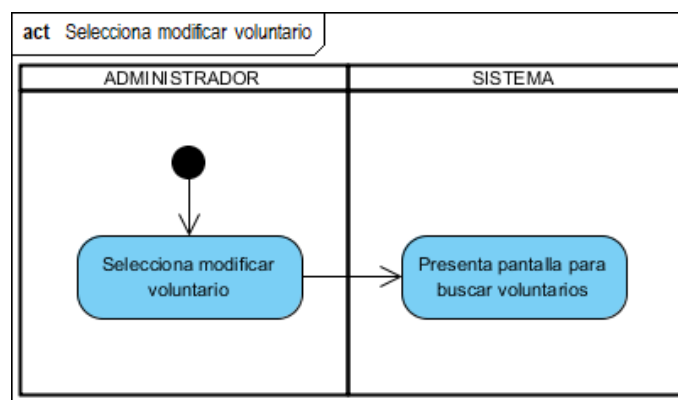


Figura 4-18 DA Modificar voluntario – Selecciona modificar voluntario

Para realizar la búsqueda, el administrador debe rellena uno o varios campos de la tabla de búsqueda y pulsar el botón 'buscar', con ello el sistema mostrará en la pantalla el listado de los voluntarios que coinciden con los campos rellenos.

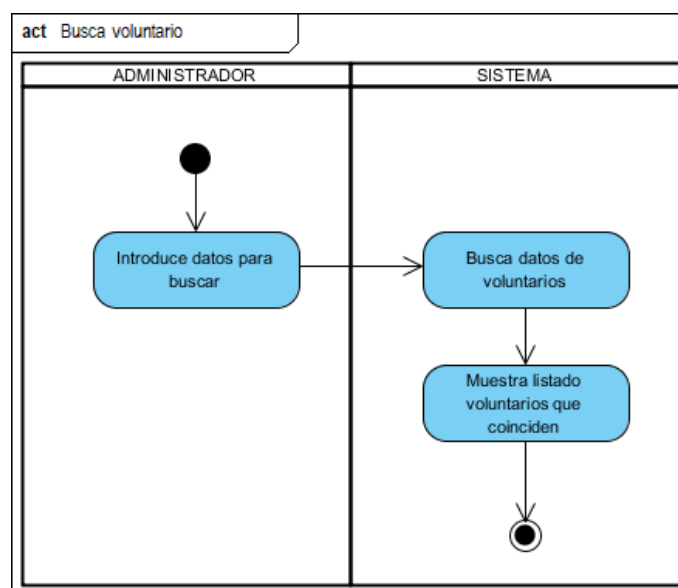


Figura 4-19 DA Modificar Voluntario – Busca voluntario

Una vez que se tiene el listado de voluntarios, el administrador seleccionará el voluntario a modificar haciendo que el sistema abra una nueva pantalla con una tabla en la que están todos los datos del voluntario. En dicha tabla el administrador podrá modificar los datos que desee y el sistema los guardará en la base de datos al pulsar el botón 'guardar', cerrando la ventana cuando se han guardado correctamente tras lo cual se actualizará el listado de voluntarios.

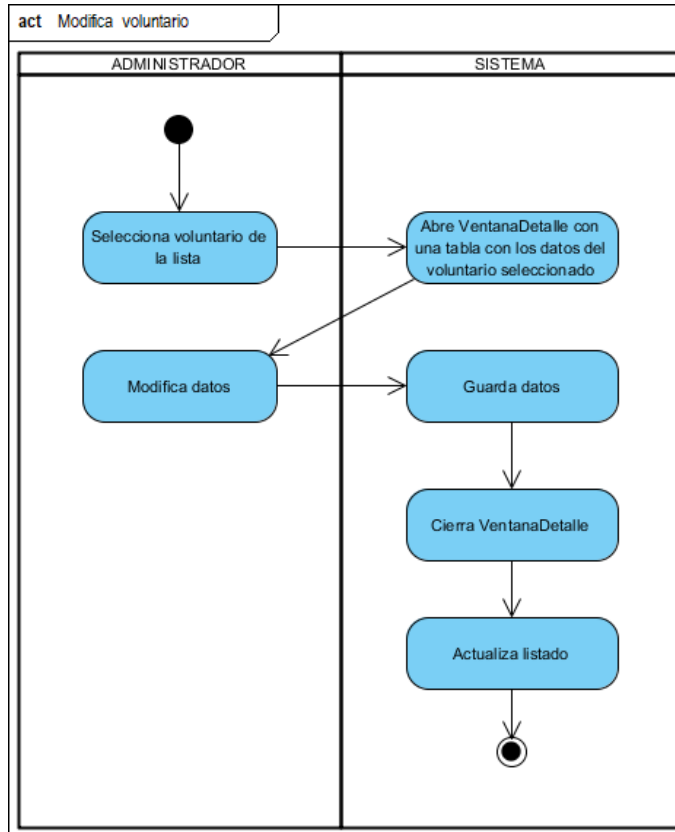


Figura 4-20 DA Modificar Voluntario – Modifica voluntario

#### 4.1.4.5 «Sub-Sistema» 05. Visualizar Usuario

El subsistema **Visualizar Usuario** se encarga de mostrar los usuarios del sistema. Los usuarios de la aplicación o user del tipo Administrador serán los únicos del sistema que cuentan en su menú con esta opción concreta, no podrá modificar los datos ni introducir datos nuevos, sólo ver los ya existentes.

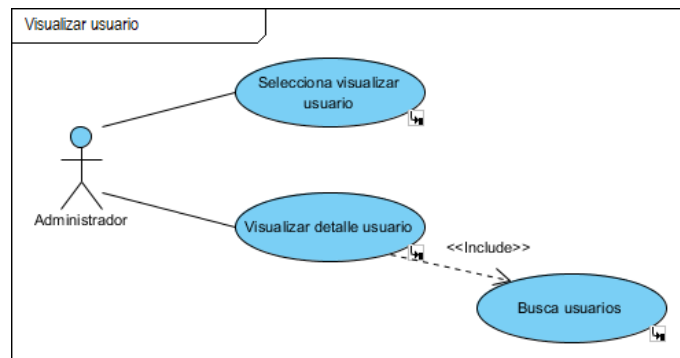


Figura 4-21 Subsistema Visualizar Usuario

<b>Designación</b>	Selecciona visualizar usuario	<b>Identificación</b>	P2.4.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Selecciona visualizar usuario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Cuando el administrador selecciona esta opción, el sistema le permitirá visualizar los usuarios		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona visualizar usuario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de visualizar usuarios, con un desplegable para buscar y una tabla para seleccionar el usuario

Tabla 4-12 CU Visualizar Usuario – Selecciona visualizar usuario

<b>Designación</b>	Busca usuarios	<b>Identificación</b>	P2.4.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Visualiza detalle usuario	Include	Busca usuario
<b>Actores</b>	Administrador		

<b>Descripción</b>	Este proceso permite buscar los voluntarios		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Introduce datos para buscar	
	2 (sistema)	Busca usuarios	El sistema busca los usuarios que coinciden con los datos introducidos
	3-(sistema)	Muestra listado de usuarios que coinciden	El sistema muestra un listado con los usuarios que coinciden con la búsqueda

Tabla 4-13 CU Visualizar Usuario – Busca usuario

<b>Designación</b>	Visualiza detalle usuario	<b>Identificación</b>	P2.4.3
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Administrador	UseCaseLink	Visualiza detalle usuario
	Visualiza detalle usuario	Include	Busca usuario
<b>Actores</b>	Administrador		
<b>Descripción</b>	Este proceso permite visualizar los detalles de los usuarios		
<b>Precondiciones</b>	A. La página del administrador debe estar cargada previamente B. La página del administrador debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Administrador D. El administrador ha realizado primero una búsqueda de usuarios		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(administrador)	Selecciona usuarios	
	2 (sistema)	Abre VentanaDetalle con tabla de datos	El sistema abre una VentanaDetalle con una tabla con los detalles del usuario seleccionado
	3-(administrador)	Selecciona cerrar VentanaDetalle	
	4-(sistema)	Cierra VentanaDetalle	El sistema cierra la VentanaDetalle

Tabla 4-14 CU Visualizar Usuario – Visualiza detalle usuario

### a. Diagrama de actividades

Cuando el user de tipo Administrador ha accedido a la aplicación cuenta en su menú con la opción de Visualizar usuario, al seleccionarla el sistema muestra una pantalla con un desplegable para seleccionar la opción de la búsqueda del usuario y un listado con los usuarios que corresponden con esos campos.

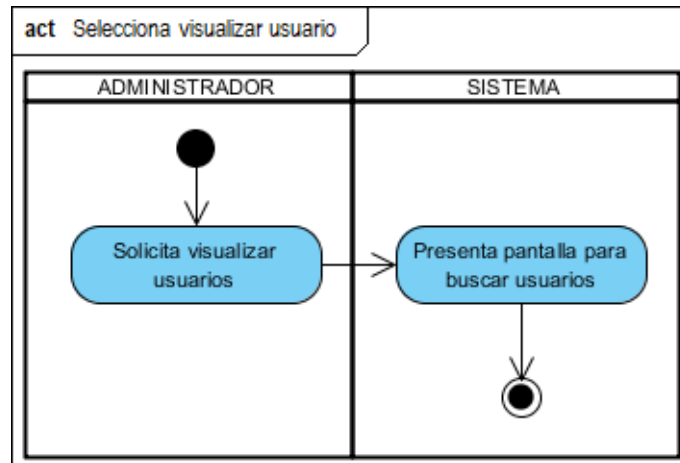


Figura 4-22 DA Visualizar Usuario – Selecciona visualizar usuario

Para realizar la búsqueda, el administrador debe seleccionar la opción en el desplegable de búsqueda y pulsar el botón 'buscar', con ello el sistema mostrará en la pantalla el listado de los usuarios que coinciden con la opción seleccionada.

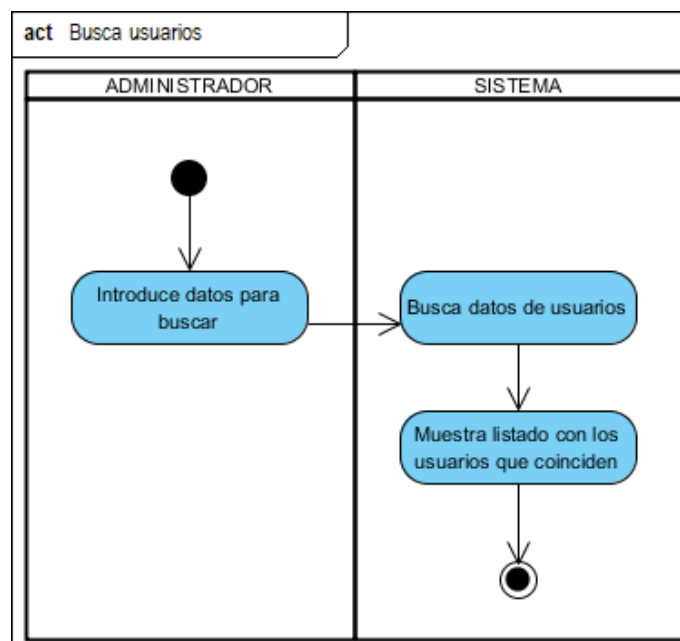


Figura 4-23 DA Visualizar Usuario – Busca usuario

Una vez que se tiene el listado de usuarios, el administrador seleccionará el usuario del que quiera ver los detalles del acompañamiento haciendo que el sistema abra una nueva pantalla con una tabla en la que están todos los detalles de los acompañamientos del usuario seleccionado. Dichos datos no podrán ser modificados por el administrador, así que una vez que ha visto los datos que necesitaba seleccionará el botón 'cancelar' para que el sistema cierre la ventana detalle

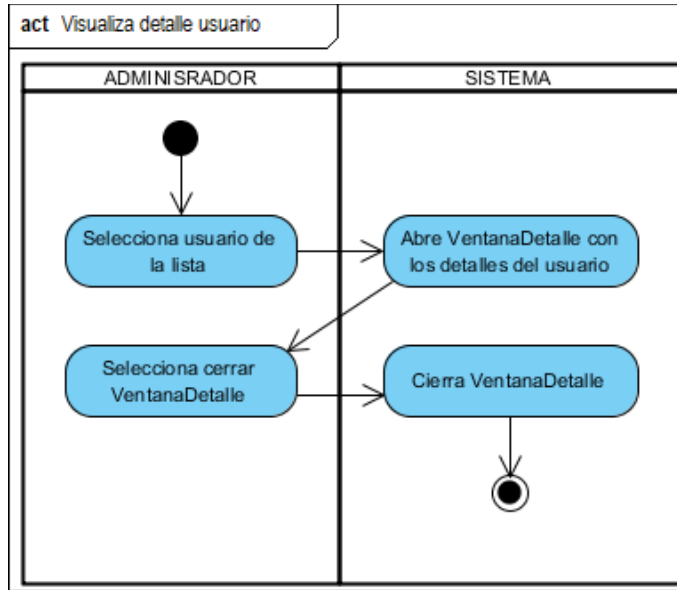


Figura 4-24 DA Visualizar Usuario – Visualiza detalle usuario

#### 4.1.4.6 «Sub-Sistema» 06. Introducir Usuario

El subsistema **Introducir Usuario** se encarga de gestionar los usuarios en el sistema. Los usuarios de la aplicación o user del tipo Personal serán los únicos del sistema que cuentan en su menú con la opción de gestionar los usuarios. La gestión de los usuarios consiste en introducir nuevos usuarios en el sistema o modificar los usuarios ya existentes.

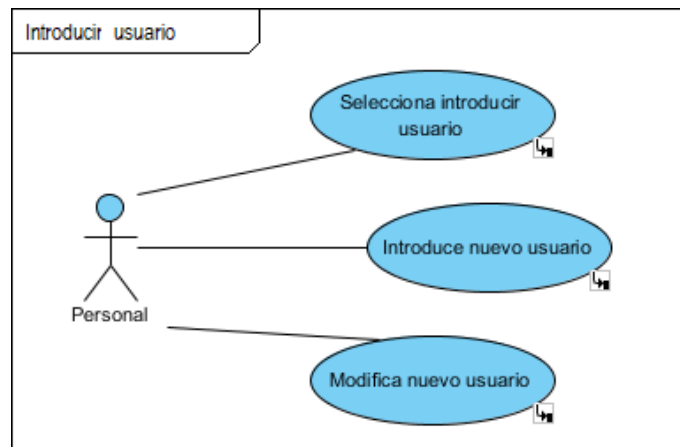


Figura 4-25 Subsistema Introducir Usuario

<b>Designación</b>	Selecciona introducir usuario	<b>Identificación</b>	P3.1.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Personal	UseCaseLink	Selecciona introducir usuario
<b>Actores</b>	Personal		
<b>Descripción</b>	Cuando el personal selecciona esta opción, el sistema le permitirá gestionar los usuarios		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Personal		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(personal)	Selecciona introducir usuario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de introducción usuario, con una celda para buscar y una tabla para introducir datos

Tabla 4-15 CU Introducir Usuario – Selecciona introducir usuario

<b>Designación</b>	Introducir nuevo usuario	<b>Identificación</b>	P3.1.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Personal	UseCaseLink	Introducir nuevo usuario
<b>Actores</b>	Personal		
<b>Descripción</b>	Este proceso permite introducir nuevos usuarios en el sistema		
<b>Precondiciones</b>	A. La página del personal debe estar cargada previamente B. La página del personal debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Personal		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(personal)	Introduce datos del usuario	
	2 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	3-(sistema)	Mensaje datos guardados	El sistema muestra un mensaje por pantalla cuando los datos se han guardado correctamente
	4-(sistema)	Actualiza listado	El sistema actualiza los usuarios

Tabla 4-16 CU Introducir Usuario – Introducir nuevo usuario

<b>Designación</b>	Modifica nuevo usuario	<b>Identificación</b>	P3.1.3
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Personal	UseCaseLink	Modificar nuevo usuario
<b>Actores</b>	Personal		
<b>Descripción</b>	Este proceso permite modificar los usuarios		
<b>Precondiciones</b>	A. La página del personal debe estar cargada previamente B. La página del personal debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Personal		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(personal)	Introduce DNI	
	2 (sistema)	Comprueba DNI	El sistema comprueba si existe algún usuario con el DNI introducido
	3-(sistema)	Rellena tabla	El sistema rellena la tabla con los datos correspondientes al DNI introducido



	4-(personal)	Modifica datos del usuario	
	5 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	6-(sistema)	Mensaje datos guardados	El sistema muestra un mensaje por pantalla cuando los datos se han guardado correctamente
	7-(sistema)	Actualiza listado	El sistema actualiza los usuarios

Tabla 4-17 CU Introducir Usuario – Modificar nuevo usuario

### a. Diagrama de actividades

Cuando el user de tipo Personal ha accedido a la aplicación cuenta en su menú con la opción de Introducir usuario, al seleccionarla el sistema muestra en pantalla una casilla para introducir el DNI y una tabla donde aparecerán los datos del usuario. Desde dicha pantalla puede buscar si el usuario que quiere introducir ya existe o introducirlo directamente.

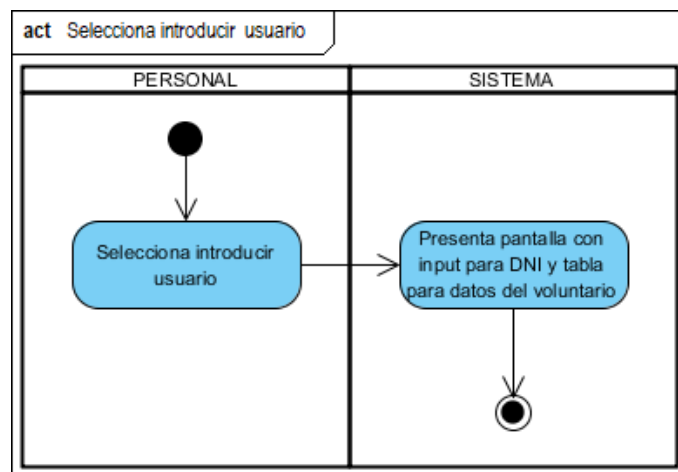


Figura 4-26 DA Introducir Usuario – Selección de introducir usuario

Si el personal decide introducir un nuevo usuario directamente deberá rellenar la tabla con los datos pertinentes, una vez completado el formulario el sistema guardará los nuevos datos en la base de datos al pulsar el botón 'guardar', mostrando un mensaje para asegurar que se han guardado correctamente tras lo cual se actualizará el listado de usuarios.

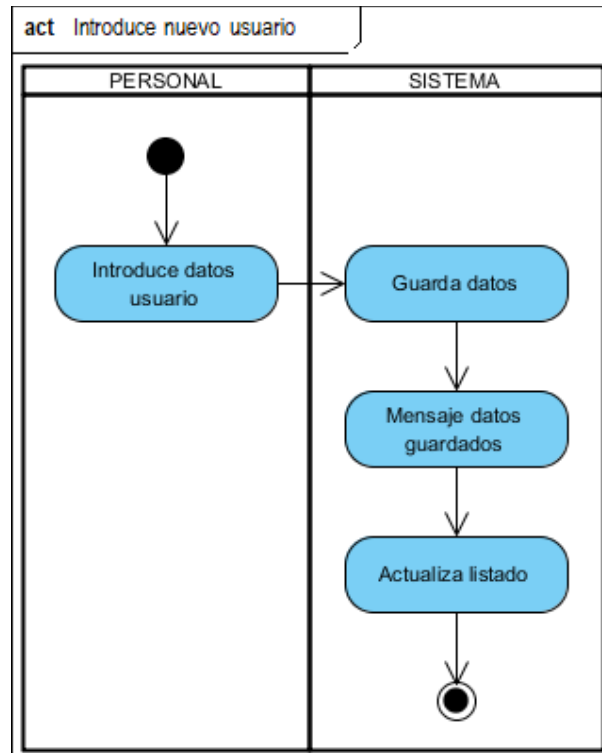


Figura 4-27 DA Introducir Usuario – Introducir nuevo usuario

Por otro lado, el personal puede buscar primero si el usuario ya existe en la base de datos para que el sistema no de error al duplicar los datos. Para ello, introducirá el DNI del usuario en la casilla correspondiente, al darle al botón 'buscar' el sistema buscará en la base de datos si existe algún usuario que corresponda con el DNI introducido y rellenará la tabla con los datos del usuario. Una vez que la tabla esté rellena, el personal podrá modificar los datos que necesite en el formulario y el sistema los guardará en la base de datos al pulsar el botón 'guardar', mostrando un mensaje para asegurar que se han guardado correctamente tras lo cual se actualizará el listado de usuarios.

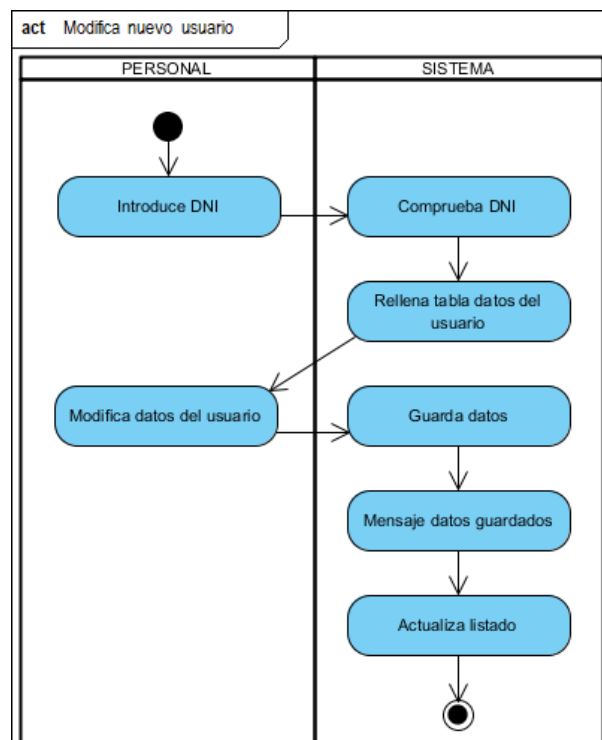


Figura 4-28 DA Introducir Usuario – Modificar nuevo usuario

## 4.1.4.7 «Sub-Sistema» 07. Modificar Usuario

El subsistema **Modificar Usuario** se encarga de gestionar los usuarios en el sistema. Los usuarios de la aplicación o user del tipo Personal serán los únicos del sistema que cuentan en su menú con la opción de gestionar los usuarios. La gestión de los usuarios consiste en buscar al usuario que se desee modificar y modificarlo.

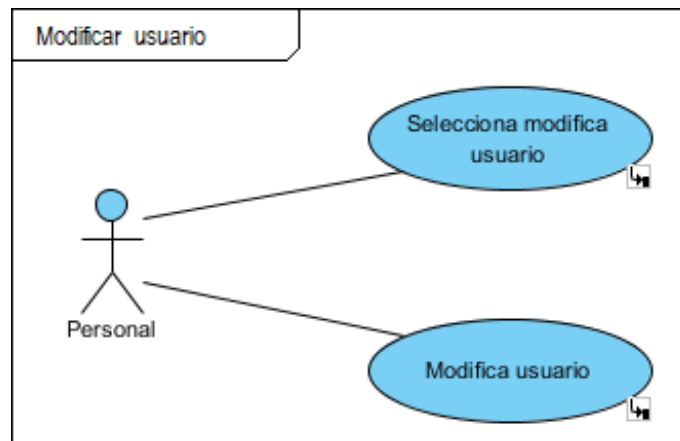


Figura 4-29 Subsistema Modificar Usuario

<b>Designación</b>	Selecciona modificar usuario	<b>Identificación</b>	P3.2.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Personal	UseCaseLink	Selecciona modificar usuario
<b>Actores</b>	Personal		
<b>Descripción</b>	Cuando el personal selecciona esta opción, el sistema le permitirá gestionar los usuarios		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Personal		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(personal)	Selecciona modificar usuario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de modificar voluntario, con una lista de usuarios para seleccionar

Tabla 4-18 CU Modificar Usuario – Selecciona modificar usuario

<b>Designación</b>	Modifica usuario	<b>Identificación</b>	P3.2.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Personal	UseCaseLink	Modifica usuario
<b>Actores</b>	Personal		
<b>Descripción</b>	Este proceso permite modificar los usuarios		
<b>Precondiciones</b>	A. La página del personal debe estar cargada previamente B. La página del personal debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Personal		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(personal)	Selecciona voluntario	
	2 (sistema)	Abre VentanaDetalle con tabla de datos	El sistema abre una VentanaDetalle con una tabla con los datos del usuario seleccionado
	3-(personal)	Modifica datos del usuario	
	4 (sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	5-(sistema)	Cierra VentanaDetalle	El sistema cierra la VentanaDetalle
	6-(sistema)	Actualiza listado	El sistema actualiza los voluntarios

Tabla 4-19 CU Modificar Usuario – Modificar usuario

## a. Diagrama de actividades

Cuando el user de tipo Personal ha accedido a la aplicación cuenta en su menú con la opción de Modificar usuario, al seleccionarla el sistema muestra una pantalla con un listado de usuarios concretos.

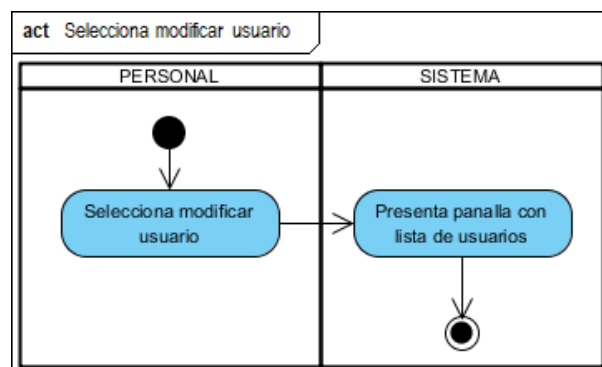


Figura 4-30 DA Modificar Usuario – Selecciona modificar usuario

Una vez que se tiene el listado de usuarios, el personal seleccionará el usuario a modificar haciendo que el sistema abra una nueva pantalla con una tabla en la que están todos los datos del usuario. En dicha tabla el personal podrá modificar los datos que desee y el sistema los guardará en la base de datos al pulsar el botón 'guardar', cerrando la ventana cuando se han guardado correctamente tras lo cual se actualizará el listado de usuarios.

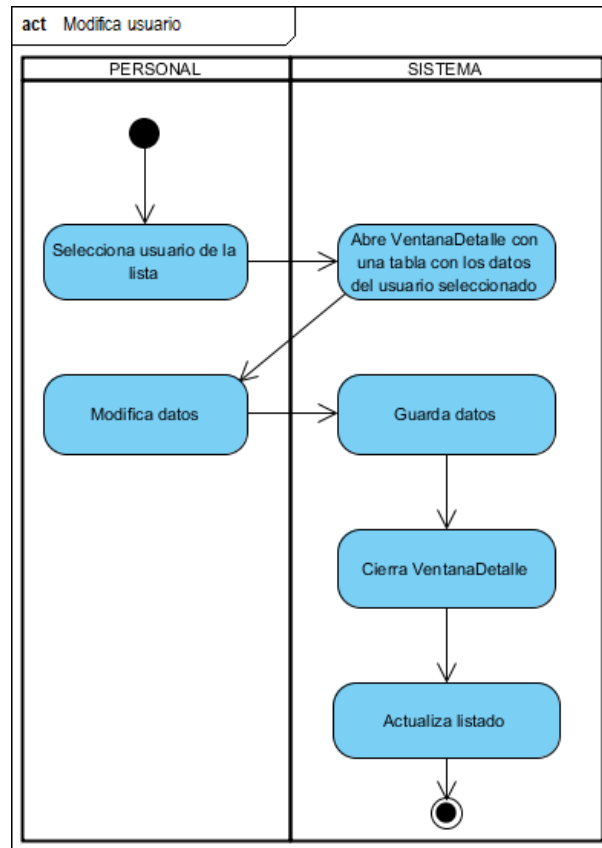


Figura 4-31 DA Modificar Usuario – Modificar usuario

## 4.1.4.8 «Sub-Sistema» 08. Visualizar Calendario

El subsistema **Visualizar Calendario** se encarga de mostrar las actividades en el sistema. Los usuarios de la aplicación o user del tipo Voluntario serán los únicos del sistema que cuentan en su menú con esta opción, no podrán modificar los datos ni introducir datos nuevos, sólo ver los ya existentes.

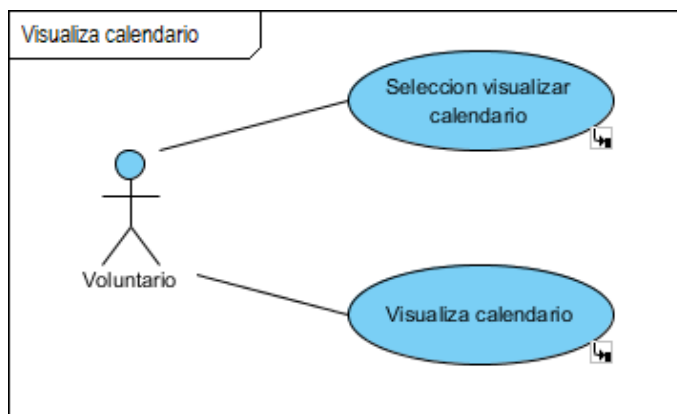


Figura 4-32 Subsistema Visualizar Calendario

<b>Designación</b>	Selecciona calendario	visualizar	<b>Identificación</b>	P4.1.1
<b>Relaciones</b>	<b>Origen</b>		<b>Estereotipo</b>	<b>Destino</b>
	Voluntario		UseCaseLink	Selecciona visualizar calendario
<b>Actores</b>	Voluntario			
<b>Descripción</b>	Cuando el voluntario selecciona esta opción, el sistema le permitirá visualizar las actividades del calendario			
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Administrador			
<b>Postcondiciones</b>	Ninguna			
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>	
	1-(voluntario)	Selecciona visualizar calendario		
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de inicio, con una tabla de datos y un calendario	

Tabla 4-20 CU Visualizar Calendario – Selecciona visualizar calendario

<b>Designación</b>	Visualizar actividad	<b>Identificación</b>	P4.1.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Voluntario	UseCaseLink	Visualizar actividad
<b>Actores</b>	Voluntario		
<b>Descripción</b>	Este proceso permite visualizar las actividades del calendario		
<b>Precondiciones</b>	A. La página del voluntario debe estar cargada previamente B. La página del voluntario debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Voluntario		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(voluntario)	Selecciona un día del calendario	
	2 (sistema)	Abre nueva ventana con las actividades	El sistema abre una nueva ventana con un listado de las actividades registradas en el calendario
	3-(voluntario)	Selecciona actividad del listado	
	4-(sistema)	Cierra ventana y rellena tabla de datos	El sistema cierra la ventana con el listado de actividades y rellena la tabla de datos de la página principal

Tabla 4-21 CU Visualizar Calendario – Visualiza actividad

## b. Diagrama de actividades

Cuando el user de tipo Voluntario ha accedido a la aplicación cuenta en su menú con la opción de Visualizar calendario, al seleccionarla el sistema muestra en pantalla una tabla donde aparecerán los datos de la actividad y un calendario donde estarán resaltados los días en los que hay alguna actividad inscrita. Desde dicha pantalla puede seleccionar el día que quiera ver en detalle.

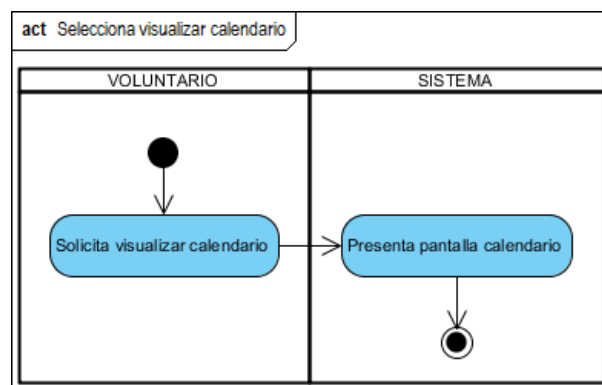


Figura 4-33 DA Visualiza Calendario – Selecciona visualizar calendario

Al seleccionar un día del calendario el sistema abrirá una nueva ventana en la que aparecerá un listado con las actividades programadas para ese día. Para poder ver en detalle los datos de la actividad, el voluntario seleccionará la actividad de la lista con lo que el sistema cerrará la ventana y rellenará la tabla con los datos de la actividad seleccionada. Dichos datos no podrán ser modificados por el voluntario, así que una vez que ha visto los datos que necesitaba podrá realizar otra acción o cerrar la aplicación.

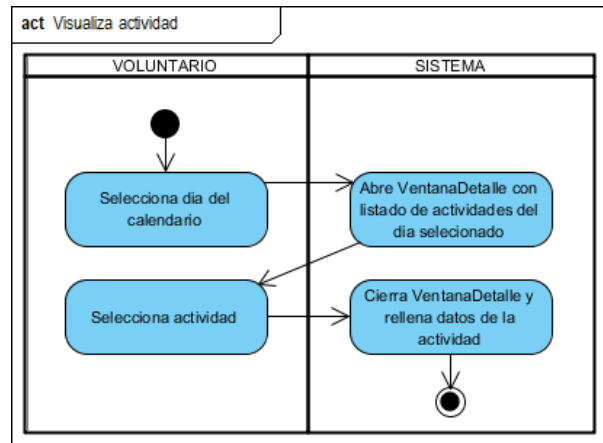


Figura 4-34 DA Visualiza Calendario – Visualiza actividad



#### 4.1.4.9 «Sub-Sistema» 09. Visualizar Voluntarios del día

El subsistema **Visualizar Voluntarios del** se encarga de mostrar los voluntarios del sistema. Los usuarios de la aplicación o user del tipo Voluntario serán los únicos del sistema que cuentan en su menú con esta opción concreta, no podrá modificar los datos ni introducir datos nuevos, sólo ver los ya existentes.

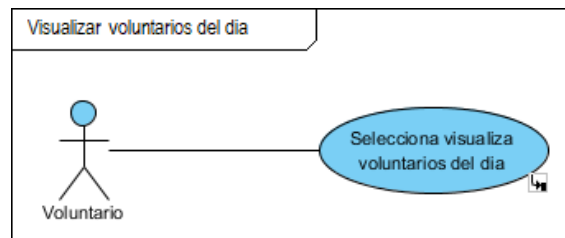


Figura 4-35 Subsistema Visualizar Voluntarios del día

<b>Designación</b>	Selecciona voluntarios del día	visualizar	<b>Identificación</b>	P4.2.1
<b>Relaciones</b>	<b>Origen</b>		<b>Estereotipo</b>	<b>Destino</b>
	Voluntario		UseCaseLink	Selecciona visualizar voluntarios del día
<b>Actores</b>	Voluntario			
<b>Descripción</b>	Cuando el voluntario selecciona esta opción, el sistema le permitirá visualizar los voluntarios			
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Voluntario			
<b>Postcondiciones</b>	Ninguna			
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>	
	1-(voluntario)	Selecciona visualizar voluntarios del día		
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de visualizar voluntarios del día, con un listado de voluntarios	

Tabla 4-22 CU Visualizar Voluntarios del día – Selecciona visualizar voluntarios del día

a. Diagrama de actividades

Cuando el user de tipo Voluntario ha accedido a la aplicación cuenta en su menú con la opción de Visualizar voluntarios del día, al seleccionarla el sistema muestra una pantalla un listado de voluntarios concreto. Dichos datos no podrán ser modificados por el voluntario, así que una vez que ha visto los datos que necesitaba podrá realizar otra acción o cerrar la aplicación.

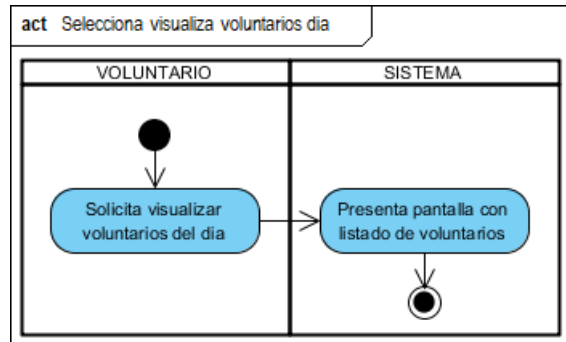


Figura 4-36 DA Visualizar Voluntarios del día – Selecciona visualizar voluntarios del día

## 4.1.4.10 «Sub-Sistema» 10. Visualizar Usuarios del día

El subsistema **Visualizar Usuarios del día** se encarga de mostrar los usuarios del sistema. Los usuarios de la aplicación o user del tipo Voluntario serán los únicos del sistema que cuentan en su menú con esta opción concreta, no podrá modificar los datos ni introducir datos nuevos, sólo ver los ya existentes.

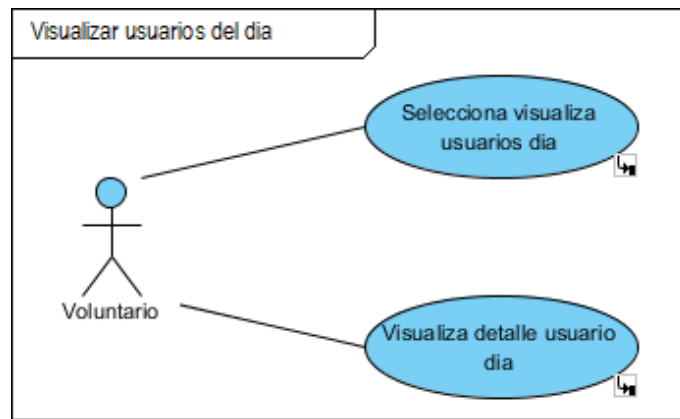


Figura 4-37 Subsistema Visualizar Usuarios del día

<b>Designación</b>	Selecciona visualizar usuario del día	<b>Identificación</b>	P4.3.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Voluntario	UseCaseLink	Selecciona visualizar usuario del día
<b>Actores</b>	Voluntario		
<b>Descripción</b>	Cuando el voluntario selecciona esta opción, el sistema le permitirá visualizar los usuarios		
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Voluntario		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(voluntario)	Selecciona visualizar usuario	
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de visualizar usuarios, con un listado de los usuarios

Tabla 4-23 CU Visualizar Usuarios del día – Selecciona visualiza usuario día

<b>Designación</b>	Visualiza detalle usuario día	<b>Identificación</b>	P4.3.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Voluntario	UseCaseLink	Visualiza detalle usuario día
<b>Actores</b>	Voluntario		
<b>Descripción</b>	Este proceso permite visualizar los detalles de los usuarios		
<b>Precondiciones</b>	A. La página del voluntario debe estar cargada previamente B. La página del voluntario debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Voluntario		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(voluntario)	Selecciona usuarios	
	2 (sistema)	Abre VentanaDetalle con tabla de datos	El sistema abre una VentanaDetalle con una tabla con los detalles del usuario seleccionado
	3-(voluntario)	Selecciona cerrar VentanaDetalle	
	4-(sistema)	Cierra VentanaDetalle	El sistema cierra la VentanaDetalle

Tabla 4-24 CU Visualizar Usuarios del día – Visualiza detalle usuario día

## a. Diagrama de actividades

Cuando el user de tipo Voluntario ha accedido a la aplicación cuenta en su menú con la opción de Visualizar usuario usuarios del día, al seleccionarla el sistema muestra una pantalla un listado de usuarios concretos.

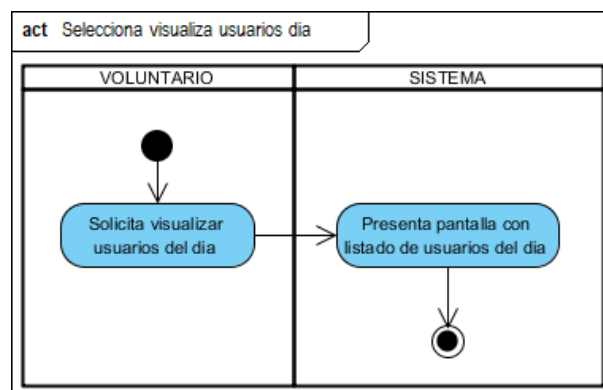


Figura 4-38 DA Visualizar Usuarios del día – Selección visualiza usuario día

Una vez que se tiene el listado de usuarios, el voluntario seleccionará el usuario del que quiera ver los detalles del acompañamiento haciendo que el sistema abra una nueva pantalla con una tabla en la que están todos los detalles de los acompañamientos del usuario seleccionado. Dichos datos no podrán ser modificados por el voluntario, así que una vez que ha visto los datos que necesitaba seleccionará el botón 'cancelar' para que el sistema cierre la ventana detalle

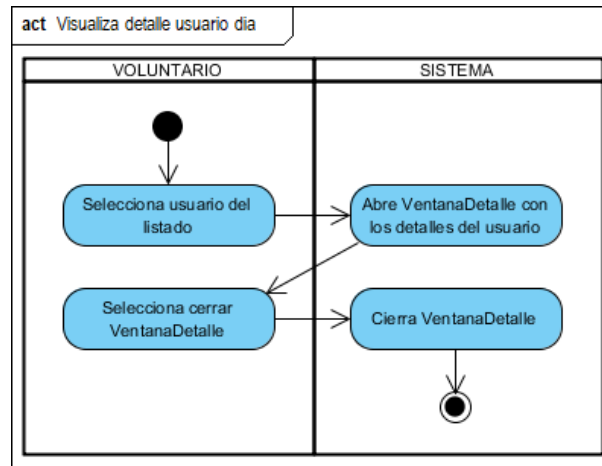


Figura 4-39 DA Visualizar Usuarios del día – Visualiza detalle usuario día

#### 4.1.4.11 «Sub-Sistema» 11. Registro actividad

El subsistema **Registro Actividad** se encarga de gestionar el registro de acompañamientos realizados en el sistema. Los usuarios de la aplicación o user del tipo Voluntario serán los únicos del sistema que cuentan en su menú con la opción de gestionar el registro. La gestión del registro consiste en introducir los nuevos acompañamientos que han realizado los voluntarios.

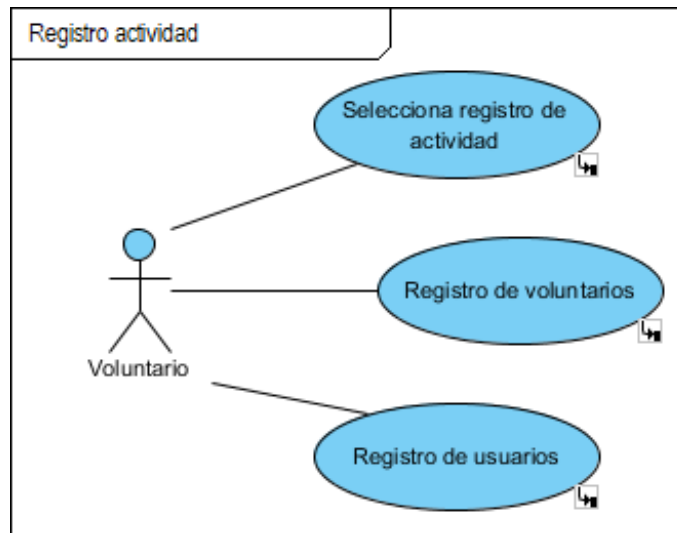


Figura 4-40 Subsistema Registro Actividad

<b>Designación</b>	Selecciona registro actividad		<b>Identificación</b>	P4.4.1
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>	
	Voluntario	UseCaseLink	Selecciona registro actividad	
<b>Actores</b>	Voluntario			
<b>Descripción</b>	Cuando el voluntario selecciona esta opción, el sistema le permitirá gestionar los acompañamientos			
<b>Precondiciones</b>	A. El user que realiza esta gestión debe ser del tipo Voluntario			
<b>Postcondiciones</b>	Ninguna			
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>	
	1-(voluntario)	Selecciona registro actividad		
	2 (sistema)	Presenta la pantalla de inicio	El sistema presenta la pantalla de registro de actividad, con un listado de voluntarios y otro de usuarios	

Tabla 4-25 CU Registro Actividad – Selecciona registro actividad

<b>Designación</b>	Registro voluntario	<b>Identificación</b>	P4.4.2
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Voluntario	UseCaseLink	Registro voluntario
<b>Actores</b>	Voluntario		
<b>Descripción</b>	Este proceso permite introducir el nuevo acompañamiento del voluntario		
<b>Precondiciones</b>	A. La página del voluntario debe estar cargada previamente B. La página del voluntario debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Voluntario		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(voluntario)	Selecciona voluntario	
	2 (sistema)	Abre VentanaDetalle	El sistema abre una Ventanaetalle con una tabla de los datos de asistencia del voluntario
	3-(voluntario)	Introduce datos	
	4-(sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	5-(sistema)	CierraVentanaDetalle	El sistema cierra la VentranaDetalle
	6-(sistema)	Actualiza listado	El sistema actualiza el registro

Tabla 4-26 CU Registro Actividad – Registro voluntario

<b>Designación</b>	Registro usuario	<b>Identificación</b>	P4.4.3
<b>Relaciones</b>	<b>Origen</b>	<b>Estereotipo</b>	<b>Destino</b>
	Voluntario	UseCaseLink	Registro usuario
<b>Actores</b>	Voluntario		
<b>Descripción</b>	Este proceso permite introducir el nuevo acompañamiento del usuario		
<b>Precondiciones</b>	A. La página del voluntario debe estar cargada previamente B. La página del voluntario debe tener acceso al servidor web C. El user que realiza esta gestión debe ser del tipo Voluntario		
<b>Postcondiciones</b>	Ninguna		
<b>Escenario</b>	<b>Pasos</b>	<b>Acción</b>	<b>Resultado</b>
	1-(voluntario)	Selecciona usuario	

	2 (sistema)	Abre VentanaDetalle	El sistema abre una VentanaDetalle con una tabla de los datos de asistencia del usuario
	3-(voluntario)	Introduce datos	
	4-(sistema)	Guarda datos	El sistema guarda los datos de la tabla en la base de datos
	5-(sistema)	CierraVentanaDetalle	El sistema cierra la VentranaDetalle
	6-(sistema)	Actualiza listado	El sistema actualiza el registro

Tabla 4-27 CU Registro Actividad – Registro usuario

#### a. Diagrama de actividades

Cuando el user de tipo Voluntario ha accedido a la aplicación cuenta en su menú con la opción de Registro de Actividad, al seleccionarla el sistema muestra en pantalla con un listado de voluntario y otro de usuarios.

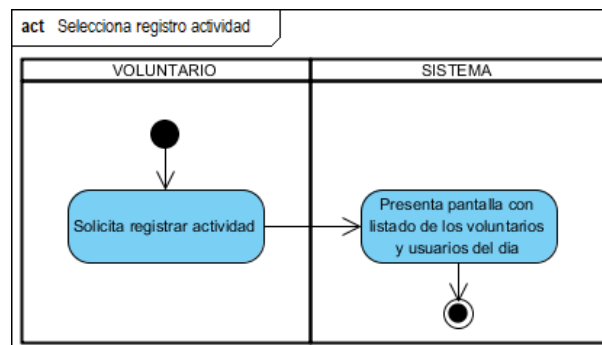


Figura 4-41 DA Registro Actividad – Selecciona registro actividad

Una vez que se tiene el listado de voluntarios, el voluntario seleccionará el voluntario al que quiera registrarle la asistencia haciendo que el sistema abra una nueva pantalla con una tabla en la que estarán algunos datos del voluntario. En dicha tabla el voluntario podrá añadir la nueva asistencia y el sistema la guardará en la base de daos al pulsar el botón 'guardar', cerrando la ventana cuando se hayan guardado los datos correctamente tras lo cual se actualizará el registro.



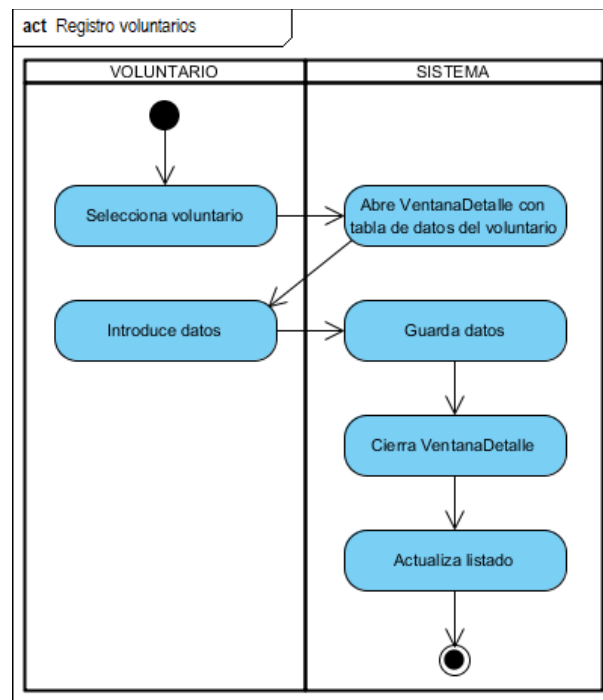


Figura 4-42 DA Registro Actividad – Registro voluntario

Paralelamente, una vez que se tiene el listado de usuarios, el voluntario seleccionará el usuario al que quiera registrarle la asistencia haciendo que el sistema abra una nueva pantalla con una tabla en la que estarán algunos datos del usuario. En dicha tabla el voluntario podrá añadir la nueva asistencia y comentario, y el sistema la guardará en la base de datos al pulsar el botón 'guardar', cerrando la ventana cuando se hayan guardado los datos correctamente tras lo cual se actualizará el registro

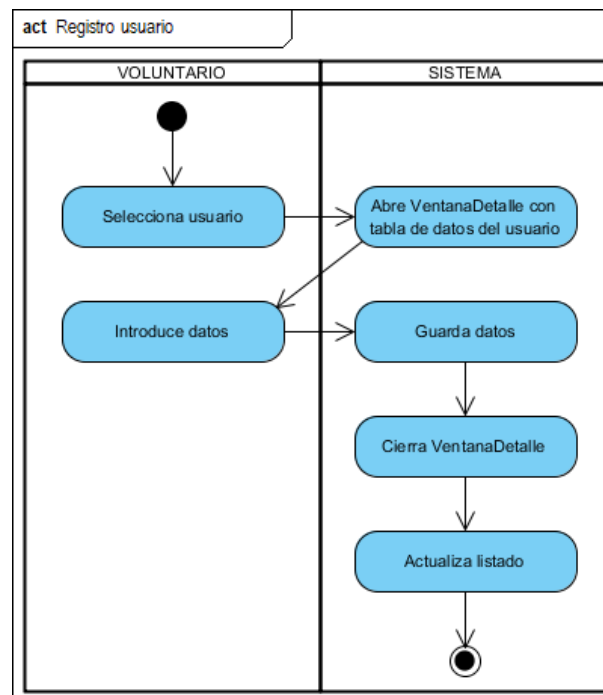


Figura 4-43 DA Registro Actividad – Registro usuario

#### 4.1.5 Prototipos de interfaces de la aplicación

En este sub-apartado se detallará el diseño que presentará la aplicación web mediante un diagrama de navegación y modelos de representación para ayudar a construir la interfaz gráfica y comprender los diagramas de casos de usos y sus requisitos.

Se usarán plantillas que representarán los diferentes prototipos, o esquemas de interfaces de usuarios de la aplicación, para completar el diseño. Esto permitirá comprender y especificar las interacciones entre los actores y la aplicación.

La forma en la que se presentará la información al usuario de la aplicación es un aspecto importante en una aplicación web ya que los esquemas de las interfaces del usuario permiten comprender y especificar las interacciones entre los actores y la aplicación durante la fase de análisis correlacionándolos con los requisitos capturados con anterioridad. Los esquemas de las interfaces del usuario de la aplicación no solo ayudan posteriormente a construir una adecuada interfaz gráfica en la aplicación, sino que ayudan a comprender mejor los casos de usos y los requisitos [8].

A continuación, se muestra el diagrama de navegación de la interfaz gráfica que proporciona una representación con los posibles recorridos de navegación para orientar al usuario de la aplicación:

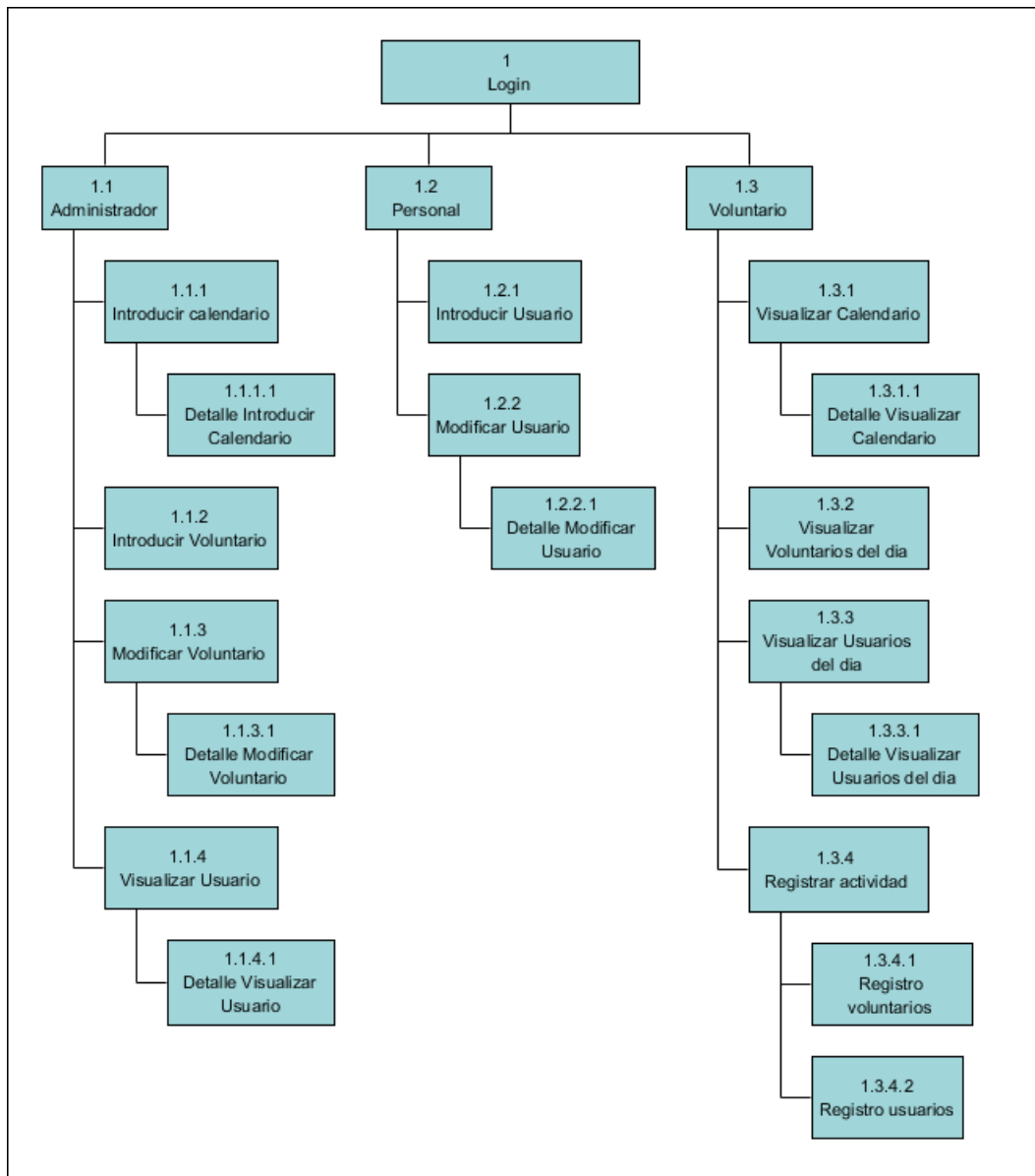


Figura 4-44 Diagrama de navegación – Aplicación web

Los esquemas de las interfaces del usuario son representaciones con características conceptuales que por medio de algunos trazos permiten definir plantillas que obliguen a mantener a las interfaces similares a lo largo de la aplicación

Los prototipos o esquemas interfaces de los usuarios que se han diseñado son:

- Prototipo Login
- Prototipo Login – Error pantalla inicial
- Prototipo Administrador
- Prototipo Administrador – Introducir Calendario
- Prototipo Administrador – Introducir Calendario – Detalle Introducir Calendario
- Prototipo Administrador – Introducir Voluntario
- Prototipo Administrador – Modificar Voluntario
- Prototipo Administrador – Modificar Voluntario – Detalle Modificar Voluntario
- Prototipo Administrador – Visualizar Usuario
- Prototipo Administrador – Visualizar Usuario – Detalle Visualizar Usuario
- Prototipo Personal
- Prototipo Personal – Introducir Usuario
- Prototipo Personal – Modificar Usuario
- Prototipo Personal – Modificar Usuario – Detalle Modificar Usuario
- Prototipo Voluntario
- Prototipo Voluntario – Visualizar Calendario
- Prototipo Voluntario – Visualizar Calendario – Detalle Visualizar Calendario
- Prototipo Voluntario – Visualizar Voluntario del día
- Prototipo Voluntario – Visualizar Usuario del día
- Prototipo Voluntario – Visualizar Usuario del día – Detalle Visualizar Usuario del día
- Prototipo Voluntario – Registrar actividad
- Prototipo Voluntario – Registrar actividad – Registro voluntario
- Prototipo Voluntario – Registrar actividad – Registro usuario

La interfaz gráfica resultante será ligeramente diferente a los prototipos debido a que durante el diseño de la aplicación web se pueden descubrir errores o modificaciones que mejoren el resultado final del proyecto.

Todos los prototipos serán hechos según el esquema de cabecera y cuerpo. La cabecera estará en la parte superior y en ella ira el título de la página en la que estamos y un menú correspondiente, mientras que el cuerpo si variará de un prototipo a otro dependiendo de las necesidades de la pantalla pudiendo contener desde formularios hasta listados.

#### 4.1.5.1 Prototipo Login

La pantalla inicial será la primera página que el usuario vea al cargar la aplicación web. En esta página login, la cabecera contendrá el menú principal y el cuerpo que estará en la parte central y contendrá el formulario de validación de password con los campos user y password a rellenar por el usuario y el botón 'Iniciar sesión, para acceder a la aplicación.

Este prototipo muestra una interfaz de usuario para el inicio de sesión. La ventana tiene un título 'PANTALLA INICIAL' y una barra de navegación con botones de retroceso, avance, cerrar y inicio. Debajo de la barra, el texto 'INICIO' indica la sección actual. El cuerpo principal contiene un formulario con dos campos de entrada: 'USER' y 'PASSWORD', cada uno con un botón de flecha para seleccionar texto. Debajo de estos campos hay un botón rectangular con el texto 'INICIAR SESION'.

Figura 4-45 Prototipo Login

Si los datos introducidos para iniciar sesión no se corresponden con los datos de la base de datos, el sistema mostrará una pantalla semejante con un mensaje de error.

Este prototipo muestra la misma interfaz de usuario que la Figura 4-45, pero con un mensaje de error. El título de la ventana es 'ERROR PANTALLA INICIAL'. El texto 'INICIO' sigue presente. En el centro del formulario, un mensaje de error indica: 'USER O PASSWORD INCORRECTO' y 'VUELVA A INTRODUCIRLO(color)'. Debajo del mensaje, los campos 'USER' y 'PASSWORD' y el botón 'INICIAR SESION' permanecen visibles.

Figura 4-46 Prototipo Login – Error pantalla inicial

#### 4.1.5.2 Prototipo Administrador

La página del administrador es la presentación de las opciones de las que dispone este tipo de user. El cuerpo de esta página estará vacío salvo un mensaje que nos informa de que la validación del user de tipo Administrador se ha hecho correctamente.

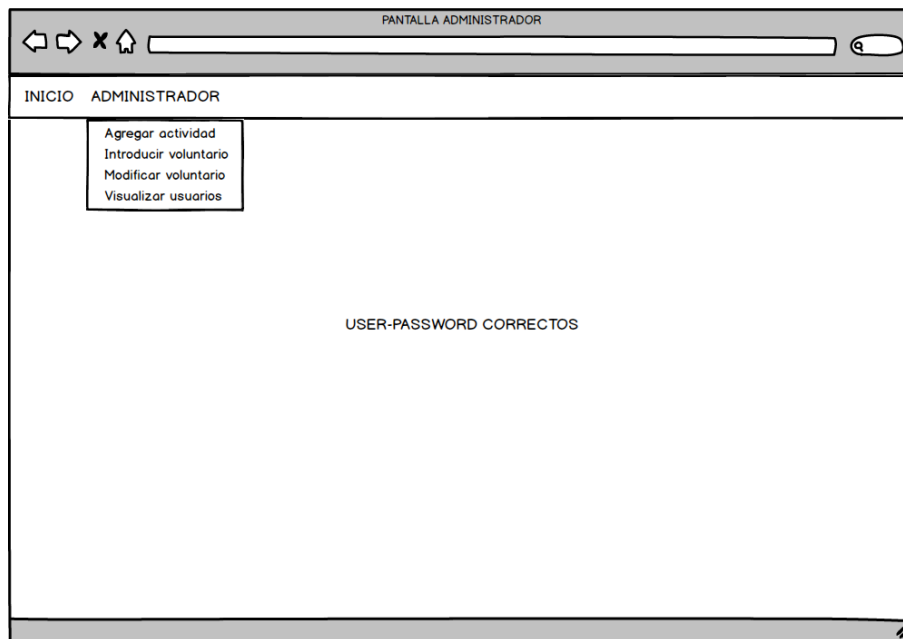


Figura 4-47 Prototipo Administrador

El menú desplegable que aparece en la cabecera de la página nos muestra las diferentes acciones que el administrador puede realizar:

- Agregar actividad.
- Introducir voluntario.
- Modificar voluntario.
- Visualizar usuarios.

Este menú es el mismo para las cuatro páginas del administrador, que se desarrollarán más extensamente en los subsiguientes apartados.

#### 4.1.5.3 Prototipo Administrador – Introducir Calendario

A la página de introducción del calendario solo puede acceder el user de tipo Administrador y en ella podrá gestionar las actividades del voluntariado. Esta página presentará el mismo menú que las otras tres páginas del administrador.

El cuerpo principal consta de dos partes, en la parte superior del cuerpo aparecerá una tabla con los campos referentes a los datos de la actividad que puede ser introducida por el administrador y el botón de 'guardar' para cuando termine de introducir los datos.

En la parte inferior del cuerpo aparecerá un mensaje del día actual en el que estamos y un calendario en el que estarán resaltados los días en los que haya alguna actividad inscrita. Dichos días se podrán seleccionar haciendo que el sistema abra una nueva ventana para todas las actividades planificadas para ese día.

Si el administrador no desea seleccionar ningún día ni introducir ninguna actividad, puede cambiar de opción en el menú, seleccionando la acción que desee, o cerrar la sesión con el botón destinado para ello.

Figura 4-48 Prototipo Administrador – Introducir Calendario

Al hacer clic en el calendario se abrirá una nueva pantalla donde se podrá ver con detalle las actividades programadas para ese día en un listado ordenado con cinco columnas: identificador, día, programa, nombre, descripción.

Al seleccionar una actividad de la lista el sistema cerrará esta ventana y rellenará la tabla de la página anterior con los datos de la actividad seleccionada para que el administrador pueda modificarla. También dispone de un botón de ‘cancelar’ por si el administrador no desea modificar ninguna de las actividades de ese día.

Dia	Nombre	Programa	Descripción

Figura 4-49 Prototipo Administrador – Introducir Calendario – Detalle Introducir Calendario

#### 4.1.5.4 Prototipo Administrador – Introducir Voluntario

A la página de introducción del voluntario solo puede acceder el user de tipo Administrador y en ella podrá gestionar los voluntarios de la base de datos. Esta página presentará el mismo menú que las otras tres páginas del administrador.

El cuerpo principal de la página consta de dos partes, la parte superior cuenta con una celda para introducir el DNI del voluntario que queremos introducir para comprobar si ya está registrado en la base de datos mediante el botón ‘comprobar’ y la parte inferior con todos los campos a introducir para registrar al nuevo voluntario.

La parte inferior además de contar con las celdas para introducir los datos del voluntario cuenta con los botones de ‘guardar’ y ‘cerrar sesión’. El administrador sólo tiene que rellenar los campos de los datos del voluntario y darle al botón de ‘guardar’ para que el sistema guarde los datos en la base de datos, una vez que los datos hayan sido guardados con éxito el sistema mostrará un mensaje por pantalla con dicha confirmación, con lo cual el administrador ya puede elegir la siguiente acción a realizar, introducir a otro voluntario, cambiar de opción en el menú o cerrar la sesión.

Aunque es posible introducir al nuevo voluntario sin comprobar antes si este está ya registrado, es conveniente hacerlo para evitar que el sistema de fallos al duplicar el elemento.

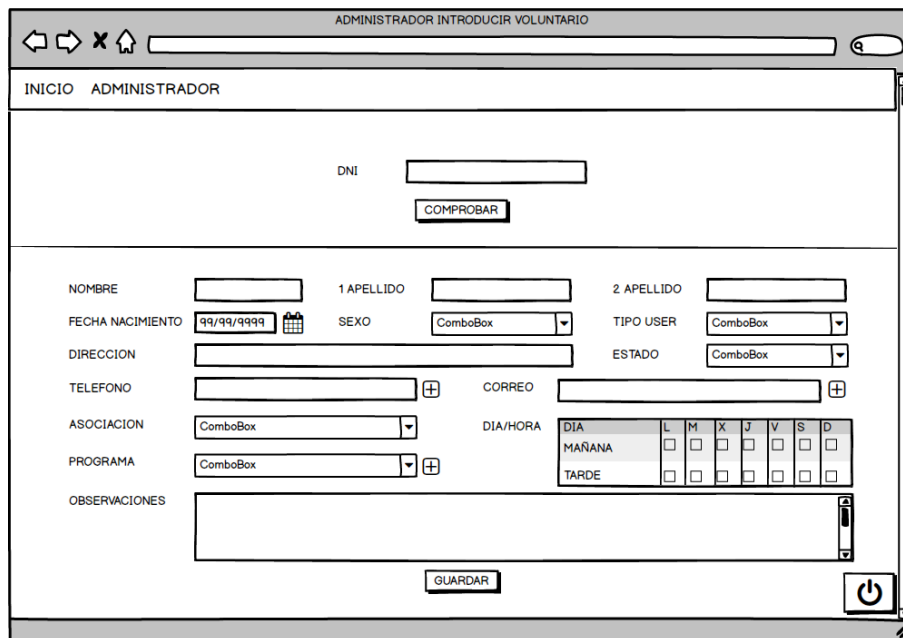


Figura 4-50 Prototipo Administrador – Introducir Voluntario

#### 4.1.5.5 Prototipo Administrador – Modificar Voluntario

A la página de modificación del voluntario solo puede acceder el user de tipo Administrador y en ella podrá gestionar los voluntarios que ya están registrados en la base de datos. Esta página presentará el mismo menú que las otras tres páginas del administrador.

El cuerpo principal de la página consta de dos partes, la parte superior cuenta una tabla con tres campos (Nombre, Programa, Dia/Hora) para introducir los datos de búsqueda con un botón de ‘buscar’. El administrador podrá introducir uno o varios campos para realizar la búsqueda.

La parte inferior cuenta con un listado ordenado, con siete columnas: DNI, Nombre, 1º Apellido, 2º Apellido, Programa, Día y Estado, que se rellenará con los voluntarios de la base de datos que coincidan con los datos introducidos en los campos de búsqueda. La parte inferior también dispone de un botón de ‘cerrar sesión’ para cuando el administrador quiera salir de la aplicación.

Este prototipo muestra una interfaz web con un título 'ADMINISTRADOR MODIFICAR VOLUNTARIO'. En la parte superior izquierda hay un menú con 'INICIO' y 'ADMINISTRADOR'. El formulario principal contiene los siguientes campos:

- NOMBRE: campo de texto.
- PROGRAMA: ComboBox.
- DÍA: ComboBox.
- Botón 'COMPROBAR'.

Debajo del formulario hay una tabla con las siguientes columnas: NOMBRE, APELLIDOS, PROGRAMA, DÍA, ESTADO y ASISTENCIA. La tabla está vacía. En la parte inferior derecha hay un botón de 'cerrar sesión' (ícono de apagado).

Figura 4-51 Prototipo Administrador – Modificar Voluntario

Una vez relleno el listado, el administrador seleccionará un voluntario y el sistema abrirá una nueva ventana con todos los campos a introducir para modificar los datos del voluntario seleccionado.

Además de contar con las celdas para introducir los datos del voluntario cuenta con los botones de ‘guardar’ y ‘cancelar’. El administrador sólo tiene que rellenar los campos de los datos del voluntario y darle al botón de ‘guardar’ para que el sistema guarde los datos en la base de datos, una vez que los datos hayan sido guardados con éxito el sistema cerrará esta ventana volviendo a la pantalla de modificación del voluntario, con lo cual el administrador ya puede elegir la siguiente acción a realizar, realizar otra búsqueda, modificar otro voluntario, cambiar de opción en el menú o cerrar la sesión.

Este prototipo muestra una interfaz web con un título 'DETALLE MODIFICAR VOLUNTARIO'. El formulario principal contiene los siguientes campos:

- NOMBRE: campo de texto.
- 1 APELLIDO: campo de texto.
- 2 APELLIDO: campo de texto.
- FECHA NACIMIENTO: campo de texto con ícono de calendario.
- SEXO: ComboBox.
- TIPO USER: ComboBox.
- DIRECCION: campo de texto.
- ESTADO: ComboBox.
- TELEFONO: campo de texto con ícono de teléfono.
- CORREO: campo de texto con ícono de correo.
- ASOCIACION: ComboBox.
- DÍA/HORA: tabla de selección de días y horas.
- PROGRAMA: ComboBox.
- OBSERVACIONES: campo de texto con scroll.
- Botón 'GUARDAR'.
- Botón de 'cerrar sesión' (ícono de apagado).

La tabla de selección de días y horas es la siguiente:

DÍA	L	M	X	J	V	S	D
MAÑANA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TARDE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 4-52 Prototipo Administrador – Modificar Voluntario – Detalle Modificar Voluntario



#### 4.1.5.6 Prototipo Administrador – Visualizar Usuario

A la página de visualización de usuarios solo puede acceder el user de tipo Administrador y en ella podrá visualizar los usuarios que ya están registrados en la base de datos. Esta página presentará el mismo menú que las otras tres páginas del administrador.

El cuerpo principal de la página consta de dos partes, la parte superior cuenta una tabla con un desplegable para seleccionar el programa que desee buscar y un botón de ‘comprobar’.

La parte inferior cuenta con un listado ordenado, con siete columnas: Nombre, 1º Apellido, 2º Apellido, Nacimiento, Programa, Estado, Cama e Ingreso, que se rellenará con los usuarios de la base de datos que coincidan con el programa seleccionado en la búsqueda. La parte inferior también dispone de un botón de ‘cerrar sesión’ para cuando el administrador quiera salir de la aplicación.

El prototipo muestra una interfaz de usuario con un navegador web. En la parte superior, hay un menú con 'INICIO' y 'ADMINISTRADOR'. Debajo, se encuentran los campos de búsqueda: 'NOMBRE' con un campo de texto, 'PROGRAMA' con un menú desplegable etiquetado como 'ComboBox', y un botón 'COMPROBAR'. Abajo de esto hay una tabla con las siguientes columnas: NOMBRE, APELLIDOS, F. NACIM, SEXO, PROGRAMA, ESTADO, CAMA, ESTANCIAS ANTERIORES, VISITAS, COMENTARIOS y F. INGRESO. La tabla está vacía. En la parte inferior derecha de la interfaz hay un botón de 'cerrar sesión' (power icon).

Figura 4-53 Prototipo Administrador – Visualizar Usuario

Una vez relleno el listado, el administrador seleccionará un usuario y el sistema abrirá una nueva ventana. Dicha ventana cuenta con dos partes, la superior aparece el nombre y apellidos del usuario seleccionado y en la parte inferior cuenta con un listado con los detalles de los comentarios de ese usuario con la fecha en la que fueron introducidos. Estos datos no podrán ser modificados por el administrador, sólo puede visualizarlos.

Además de esto, en la parte inferior de la ventana hay un botón de ‘cancelar’ para salir de la ventana lo que hará que vuelva a la pantalla de visualización de usuario donde podrá seleccionar otro usuario para ver sus detalles, realizar otra búsqueda, cambiar de opción en el menú o cerrar la sesión.

El prototipo muestra una interfaz de usuario para el detalle de un usuario. En la parte superior, hay tres campos de texto etiquetados como 'NOMBRE', '1 APELLIDO' y '2 APELLIDO'. Debajo, hay una tabla con dos columnas: 'FECHA' y 'COMENTARIOS'. La tabla está vacía. En la parte inferior derecha de la interfaz hay un botón de 'cancelar' (power icon).

Figura 4-54 Prototipo Administrador – Visualizar Usuario – Detalle Visualizar Usuario

#### 4.1.5.7 Prototipo Personal

La página del personal es la presentación de las opciones de las que dispone este tipo de user. El cuerpo de esta página estará vacío salvo un mensaje que nos informa de que la validación del user de tipo Personal se ha hecho correctamente.

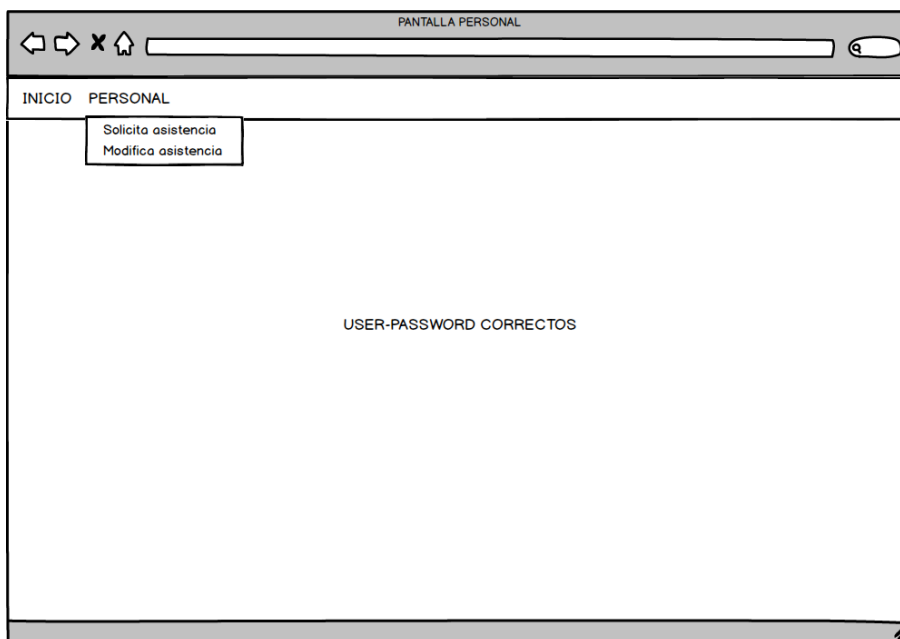


Figura 4-55 Prototipo Personal

El menú desplegable que aparece en la cabecera de la página nos muestra las diferentes acciones que el personal puede realizar:

- Solicitar asistencia.
- Modificar asistencia.

Este menú es el mismo para las dos páginas del personal, que se desarrollarán más extensamente en los subsiguientes apartados.

#### 4.1.5.8 Prototipo Personal – Introducir Usuario

A la página de solicitar asistencia solo puede acceder el user de tipo Personal y en ella podrá gestionar los usuarios de la base de datos. Esta página presentará el mismo menú que la otra página del personal de modificación de usuarios.

El cuerpo principal de la página consta de dos partes, la parte superior cuenta con una celda para introducir el DNI del usuario que queremos introducir para comprobar si ya está registrado en la base de datos mediante el botón ‘comprobar’ y la parte inferior con todos los campos a introducir para registrar al nuevo usuario.

La parte inferior además de contar con las celdas para introducir los datos del usuario cuenta con los botones de ‘guardar’ y ‘cerrar sesión’. El personal sólo tiene que rellenar los campos de los datos del usuario y darle al botón de ‘guardar’ para que el sistema guarde los datos en la base de datos, una vez que los datos hayan sido guardados con éxito el sistema mostrará un mensaje por pantalla con dicha confirmación, con lo cual el personal ya puede elegir la siguiente acción a realizar, introducir a otro usuario, cambiar de opción en el menú o cerrar la sesión.

Aunque es posible introducir al nuevo usuario sin comprobar antes si este está ya registrado es conveniente hacerlo para evitar que el sistema de fallos al duplicar el elemento.

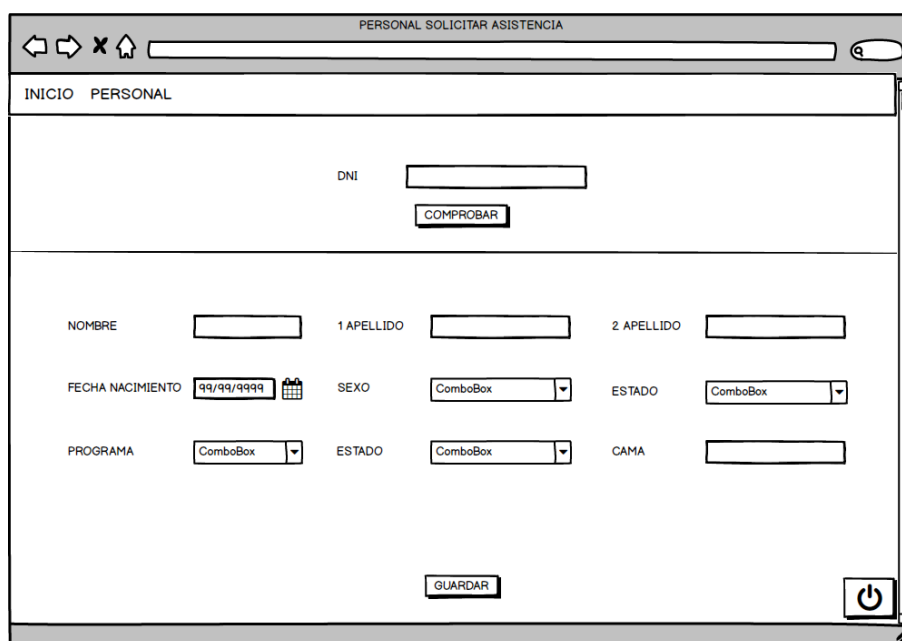


Figura 4-56 Prototipo Personal – Introducir Usuario

#### 4.1.5.9 Prototipo Personal – Modificar Usuario

A la página de modificación del usuario solo puede acceder el user de tipo Personal y en ella podrá gestionar los usuarios que ya están registrados en la base de datos. Esta página presentará el mismo menú que la otra página del personal de introducción de usuarios.

El cuerpo principal de la página consta de una sola parte con un listado ordenado, con siete columnas: DNI, Nombre, 1º Apellido, 2º Apellido, Programa, Estado, Cama y Comentarios, que se rellenará con los usuarios de la base de datos. También dispone de un botón de ‘cerrar sesión’ para cuando el personal quiera salir de la aplicación.

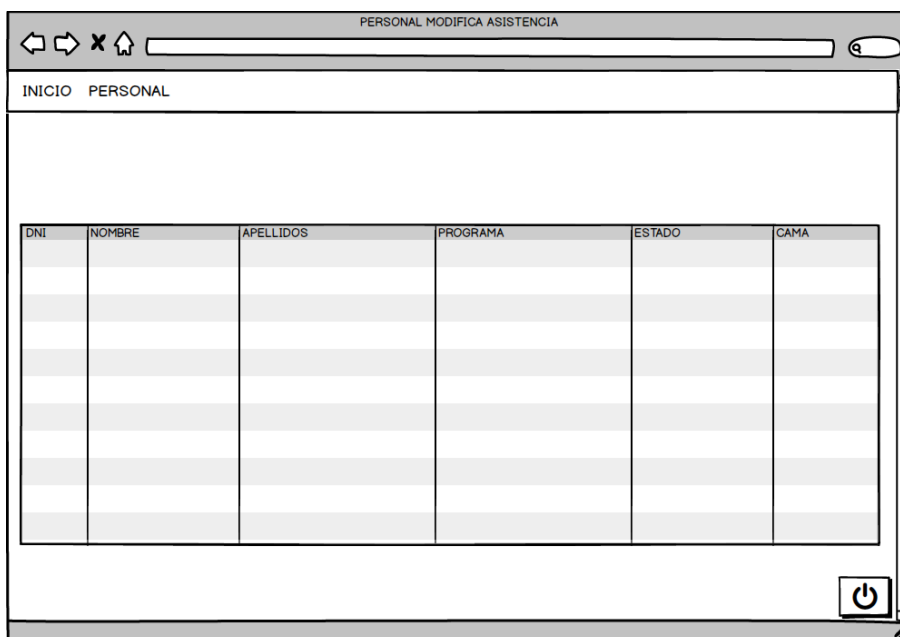


Figura 4-57 Prototipo Personal – Modificar Usuario

El personal seleccionará un usuario y el sistema abrirá una nueva ventana con todos los campos a introducir para modificar los datos del usuario seleccionado.

Además de contar con las celdas para introducir los datos del usuario cuenta con los botones de ‘guardar’ y ‘cancelar’. El personal sólo tiene que rellenar los campos de los datos del usuario y darle al botón de ‘guardar’ para que el sistema guarde los datos en la base de datos, una vez que los datos hayan sido guardados con éxito el sistema cerrará esta ventana volviendo a la pantalla de modificación del usuario, con lo cual el personal ya puede elegir la siguiente acción a realizar, modificar otro usuario, cambiar de opción en el menú o cerrar la sesión.

DETALLE MODIFICAR ASISTENCIA

NOMBRE  1 APELLIDO  2 APELLIDO

FECHA NACIMIENTO  SEXO  ESTADO

PROGRAMA  ESTADO  CAMA

Figura 4-58 Prototipo Personal – Modificar Usuario – Detalle Modificar Usuario

#### 4.1.5.10 Prototipo Voluntario

La página del voluntario es la presentación de las opciones de las que dispone este tipo de user. El cuerpo de esta página estará vacío salvo un mensaje que nos informa de que la validación del user de tipo Voluntario se ha hecho correctamente.

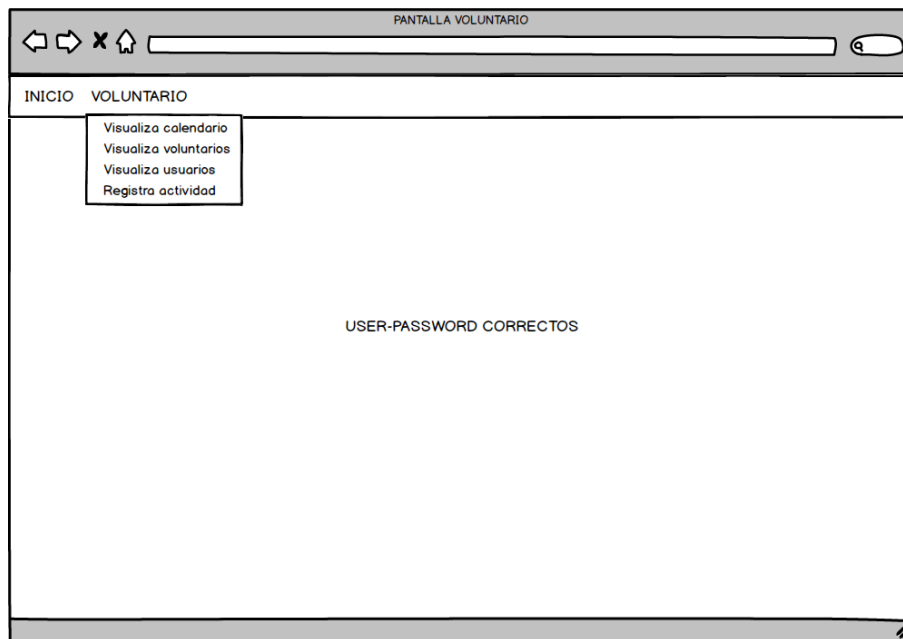


Figura 4-59 Prototipo Voluntario

El menú desplegable que aparece en la cabecera de la página nos muestra las diferentes acciones que el voluntario puede realizar:

- Visualizar calendario.
- Visualizar voluntarios del día.
- Visualizar usuarios del día.
- Registrar actividad.

Este menú es el mismo para las cuatro páginas del voluntario, que se desarrollarán más extensamente en los subsiguientes apartados.

#### 4.1.5.11 Prototipo Voluntario – Visualizar Calendario

A la página de visualización del calendario solo puede acceder el user de tipo Voluntario y en ella podrá visualizar las actividades del calendario. Esta página presentará el mismo menú que las otras tres páginas del voluntario.

El cuerpo principal consta de dos partes, en la parte superior del cuerpo aparecerá una tabla con los campos referentes a los datos de la actividad. En la parte inferior del cuerpo aparecerá un mensaje del día actual en el que estamos y un calendario en el que estarán resaltados los días en los que haya alguna actividad inscrita. Dichos días se podrán seleccionar haciendo que el sistema abra una nueva ventana para todas las actividades planificadas para ese día.

Si el voluntario no desea seleccionar ningún, puede cambiar de opción en el menú, seleccionando la acción que desee, o cerrar la sesión con el botón destinado para ello.

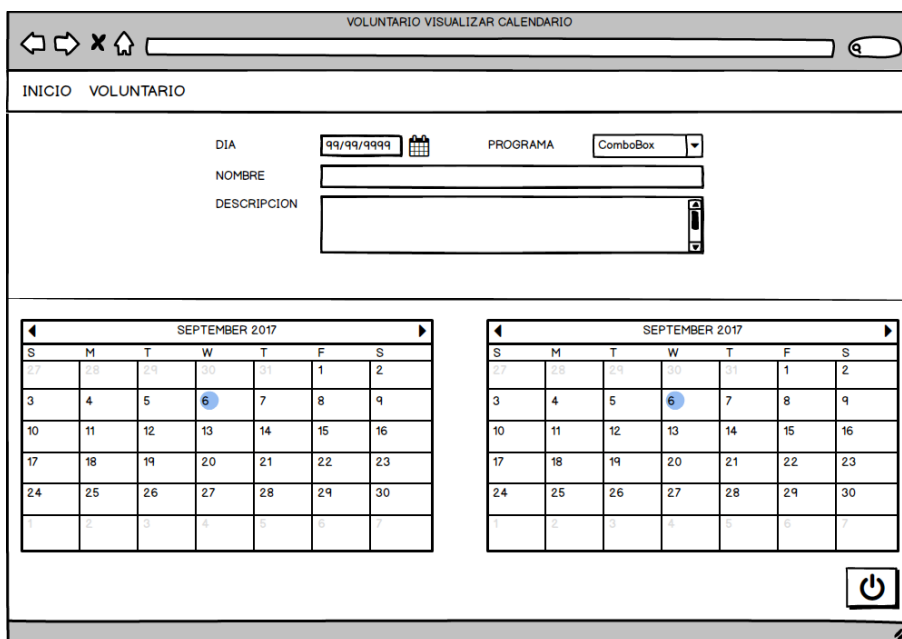


Figura 4-60 Prototipo Voluntario – Visualizar Calendario

Al hacer clic en el calendario se abrirá una nueva pantalla donde se podrá ver con detalle las actividades programadas para ese día en un listado ordenado con cinco columnas: identificador, día, programa, nombre, descripción.

Al seleccionar una actividad de la lista el sistema cerrará esta ventana y rellenará la tabla de la página anterior con los datos de la actividad seleccionada. Estos datos no podrán ser modificados por el voluntario, sólo puede visualizarlos.

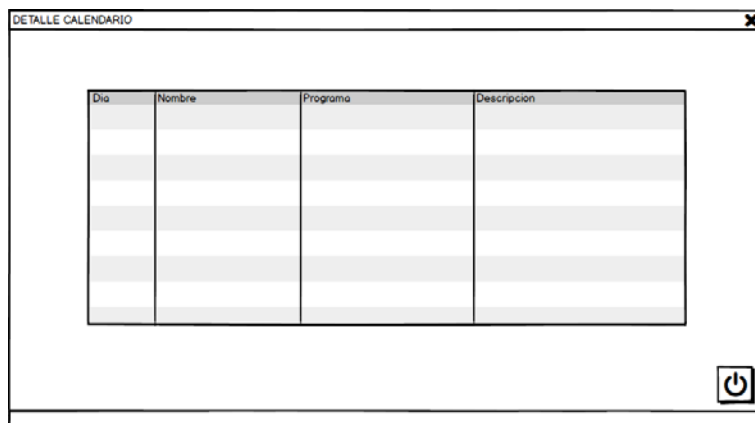


Figura 4-61 Prototipo Voluntario – Visualizar Calendario – Detalle Visualizar Calendario

#### 4.1.5.12 Prototipo Visualizar Voluntario del día

A la página de visualización de voluntarios del día solo puede acceder el user de tipo Voluntario y en ella podrá visualizar los voluntarios que ya están registrados en la base de datos. Esta página presentará el mismo menú que las otras tres páginas del voluntario.

El cuerpo principal de la página consta de una parte con un listado ordenado, con cinco columnas: Nombre, 1º Apellido, 2º Apellido, Teléfono y Asociación, que se rellenará con los voluntarios de la base de datos. Estos datos no podrán ser modificados por el voluntario, sólo puede visualizarlos. Cuando haya visto lo que necesitaba podrá realizar otra acción seleccionando la que desee del menú o cerrar la sesión con el botón destinado para ello.

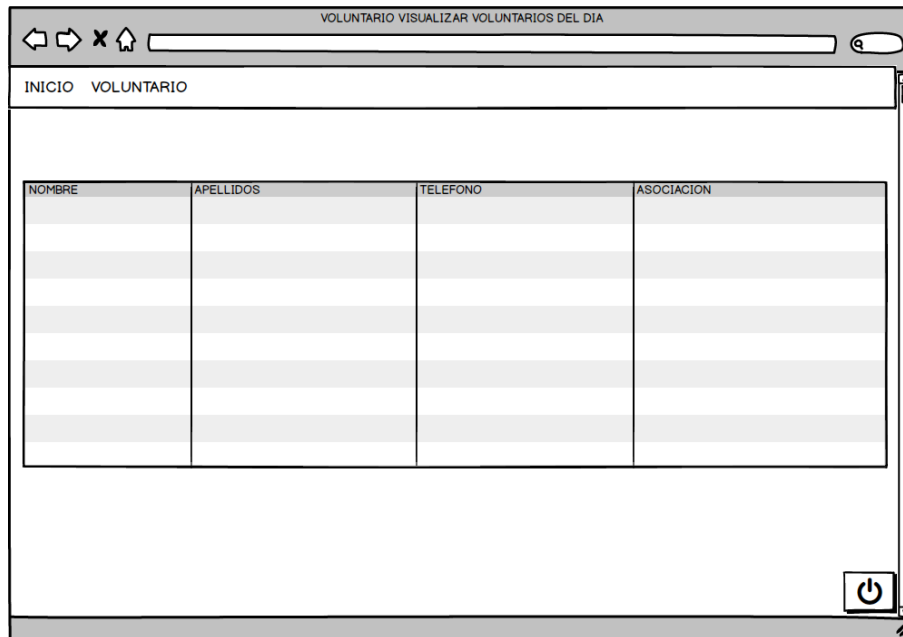


Figura 4-62 Prototipo Voluntario – Visualizar Voluntario del día

#### 4.1.5.13 Prototipo Visualizar Usuario del día

A la página de visualización de usuarios del día solo puede acceder el user de tipo Voluntario y en ella podrá visualizar los usuarios que ya están registrados en la base de datos. Esta página presentará el mismo menú que las otras tres páginas del voluntario.

El cuerpo principal de la página consta de una parte con un listado ordenado, con siete columnas: Nombre, 1º Apellido, 2º Apellido, Nacimiento, Programa, Estado, Cama e Ingreso, que se rellenará con los usuarios de la base de datos. Estos datos no podrán ser modificados por el voluntario, sólo puede visualizarlos. Cuando haya visto lo que necesitaba podrá realizar otra acción seleccionando la que desee del menú o cerrar la sesión con el botón destinado para ello.

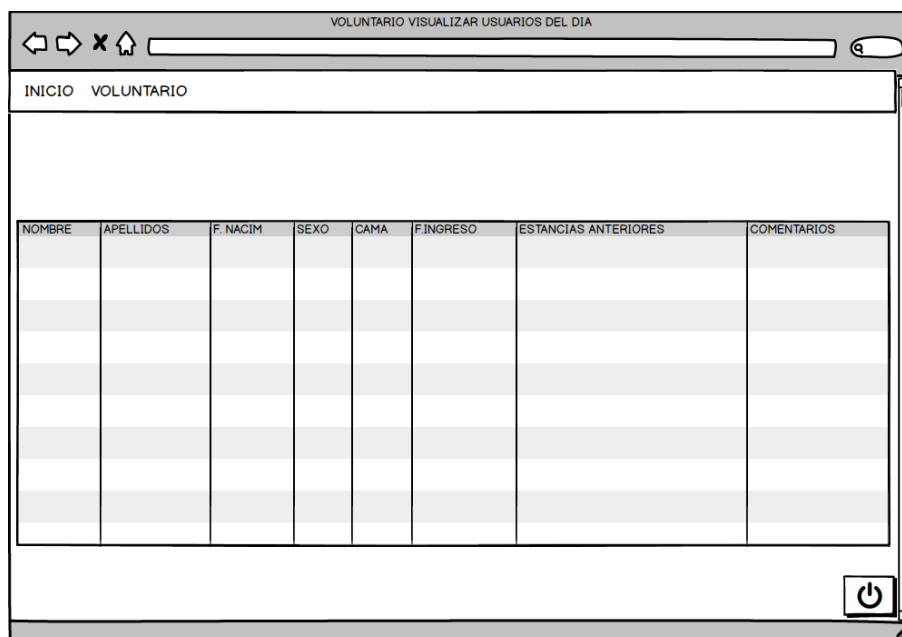


Figura 4-63 Prototipo Voluntario – Visualizar Usuario del día

Una vez relleno el listado, el voluntario seleccionará un usuario y el sistema abrirá una nueva ventana. Dicha ventana cuenta con dos partes, la superior aparece el nombre y apellidos del usuario seleccionado y en la parte inferior cuenta con un listado con los detalles de los comentarios de ese usuario con la fecha en la que fueron introducidos. Estos datos no podrán ser modificados por el voluntario, sólo puede visualizarlos.

Además de esto, en la parte inferior de la ventana hay un botón de ‘cancelar’ para salir de la ventana lo que hará que vuelva a la pantalla de visualización de usuario del día donde podrá seleccionar otro usuario para ver sus detalles, cambiar de opción en el menú o cerrar la sesión.

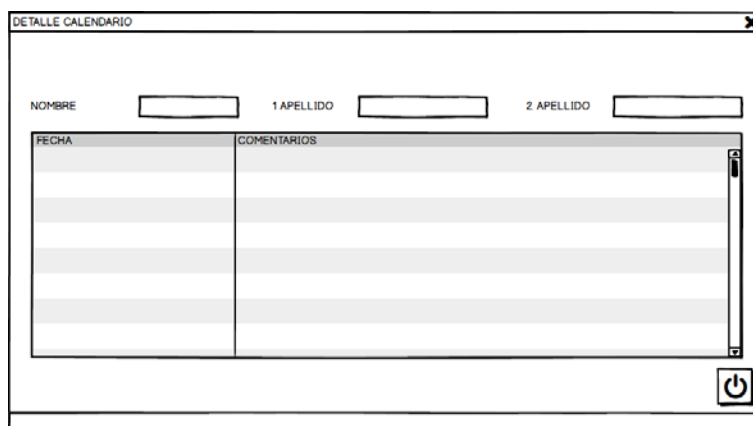


Figura 4-64 Prototipo Voluntario – Visualizar Usuario del día – Detalle Visualizar Usuario del día



#### 4.1.5.14 Prototipo Voluntario – Registrar actividad

A la página de registro de actividad solo puede acceder el user de tipo Voluntario y en ella podrá registrar las actividades realizadas por el voluntario y los acompañamientos a los usuarios que ya están registrados en la base de datos. Esta página presentará el mismo menú que las otras tres páginas del voluntario.

El cuerpo principal de la página consta de dos partes, la parte superior cuenta con un listado ordenado, con cinco columnas: Nombre, 1º Apellido, 2º Apellido, Teléfono y Asociación, que se rellenará con los voluntarios de la base de datos. Estos datos no podrán ser modificados por el voluntario, sólo puede visualizarlos. Para registrar la actividad realizada debe seleccionar el voluntario que ha realizado la actividad, con ello el sistema abrirá una nueva ventana para introducir los datos de la actividad.

La parte inferior cuenta con un listado ordenado, con siete columnas: Nombre, 1º Apellido, 2º Apellido, Nacimiento, Programa, Estado, Cama e Ingreso, que se rellenará con los usuarios de la base de datos. Estos datos no podrán ser modificados por el voluntario, sólo puede visualizarlos. Para registrar el acompañamiento realizado debe seleccionar el usuario que ha recibido el acompañamiento, con ello el sistema abrirá una nueva ventana para introducir los datos del acompañamiento.

Cuando haya registrado todos los datos podrá realizar otra acción seleccionando la que desee del menú o cerrar la sesión con el botón destinado para ello.

VOLUNTARIO REGISTRAR ACTIVIDAD			
INICIO VOLUNTARIO			
NOMBRE	APELLIDOS	TELEFONO	ASOCIACION

NOMBRE	APELLIDOS	F. NACIM	SEXO	CAMA	F. INGRESO	ESTANCIAS ANTERIORES	COMENTARIOS

Figura 4-65 Prototipo Voluntario – Registrar actividad

Una vez relleno el listado de voluntarios, el voluntario seleccionará un voluntario y el sistema abrirá una nueva ventana con todos los campos a introducir para registrar la actividad.

Además de contar con las celdas para introducir los datos del voluntario cuenta con los botones de ‘guardar’ y ‘cancelar’. El voluntario sólo tiene que rellenar el campos de la fecha puesto que los otros campos están ya rellenos con los datos de la selección y darle al botón de ‘guardar’ para que el sistema guarde los datos en la base de datos, una vez que los datos hayan sido guardados con éxito el sistema cerrará esta ventana volviendo a la pantalla de registro de actividad, con lo cual el voluntario ya puede elegir la siguiente acción a realizar, registrar otra actividad, registrar un acompañamiento, cambiar de opción en el menú o cerrar la sesión.

DETALLE REGISTRAR VOLUNTARIO

DNI  FECHA

NOMBRE  1 APELLIDO  2 APELLIDO

Figura 4-66 Prototipo Voluntario – Registrar actividad – Registro voluntario

Una vez relleno el listado de usuarios, el voluntario seleccionará un usuario y el sistema abrirá una nueva ventana con todos los campos a introducir para registrar el acompañamiento.

Además de contar con las celdas para introducir los datos del voluntario cuenta con los botones de ‘guardar’ y ‘cancelar’. El voluntario sólo tiene que rellenar el campos de la fecha y el del comentario puesto que los otros campos están ya rellenos con los datos de la selección y darle al botón de ‘guardar’ para que el sistema guarde los datos en la base de datos, una vez que los datos hayan sido guardados con éxito el sistema cerrará esta ventana volviendo a la pantalla de registro de actividad, con lo cual el voluntario ya puede elegir la siguiente acción a realizar, registrar otro acompañamiento, registrar una actividad, cambiar de opción en el menú o cerrar la sesión.

DETALLE REGISTRAR USUARIO

DNI  FECHA

NOMBRE  1 APELLIDO  2 APELLIDO

DESCRIPCION

Figura 4-67 Prototipo Voluntario – Registrar actividad – Registro usuario

## 4.2 Diseño de la Base de datos

### 4.2.1 Modelo relacional

La base de datos contiene toda la información necesaria para poder gestionar el voluntariado de forma eficiente, será organizada y representada por un modelo relacional. Actualmente el modelo relacional es el más utilizado ya que permite modelar y administrar los datos dinámicamente. En este tipo de modelo, la información es representada mediante tablas o relaciones, donde cada fila de la tabla se interpreta como una relación ordenada de valores.

A continuación, es mostrado el modelo relacional de la base de datos creada para este proyecto:

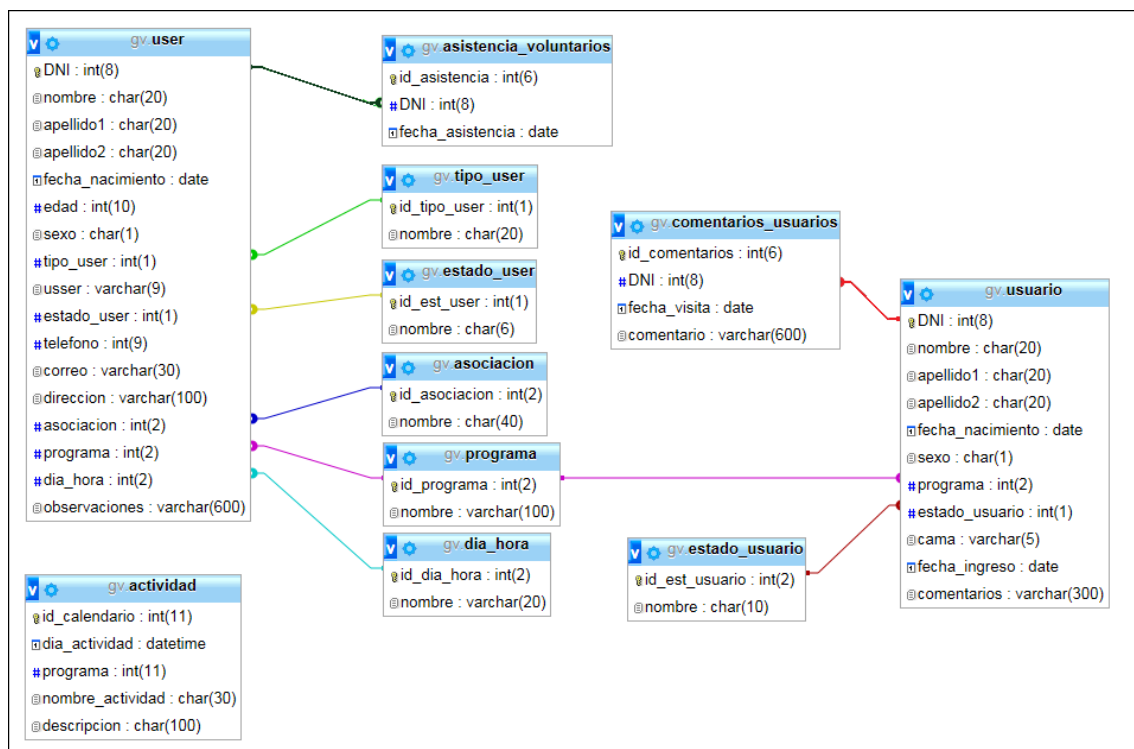


Figura 4-68 Modelo relacional

### 4.2.2 Diagrama clases

Las relaciones de las diferentes carpetas que forman la aplicación prototipo y las distintas funciones que componen la capa de la base de datos están detalladas a continuación.

En la siguiente figura se muestra un ejemplo de cómo estructurar la carpeta de contiene los archivos de la aplicación prototipo, para ello se ha utilizado como ejemplo las carpetas y archivos que componen las páginas relacionadas con el calendario, Agregar Calendario y Visualizar Calendario, el resto de módulos estarán basados en el mismo patrón.

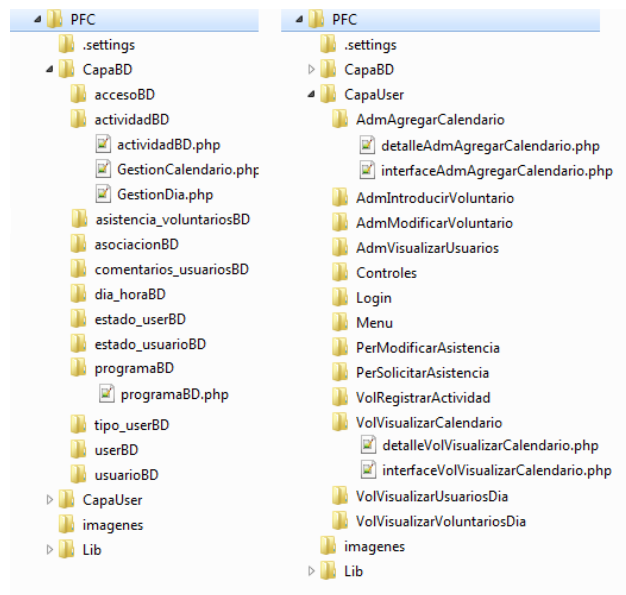


Figura 4-69 Carpetas de la aplicación web

En la siguiente figura se representa gráficamente la relación de cada uno de los archivos creados y las funciones que contienen alguno de ellos. El esquema es genérico y el nombre de la carpeta donde está el archivo se corresponde con las 'xxxx'. También se muestra la relación con las diferentes opciones que muestra la aplicación, por ejemplo, los archivos php que generan la interfaz de visualización de calendario sería: interfaceVolVosualizarCalendario.php, detalleVolVisualizarCalendario.php, GestionCalendario.php, GestionDia.php, actividad.php, programa.php.

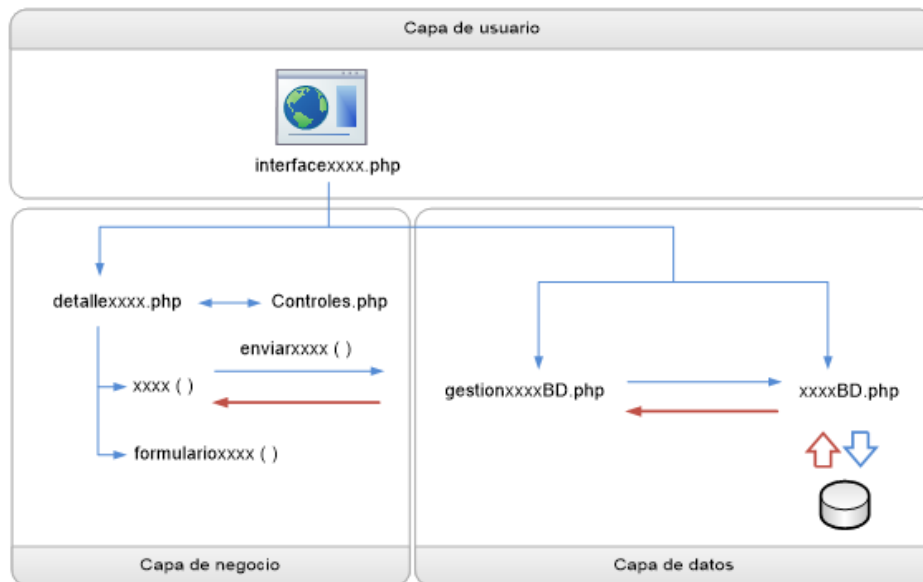


Figura 4-70 Relación entre archivos

Los diagramas de clases son un tipo de diagrama de estructura estática que describen las estructuras de un sistema mostrando sus clases, atributos y operaciones (o métodos), y las relaciones entre los objetos.

Las clases representan los bloques de construcción más importantes de cualquier sistema orientado a objetos. Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Su representación gráfica se muestra en la siguiente figura [2].

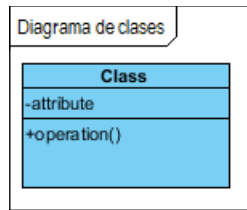


Figura 4-71 Diagrama de clase

Dicha notación es independiente al lenguaje de programación. Primero se representa el nombre de la clase, después aparecen los atributos y por último se muestran las operaciones.

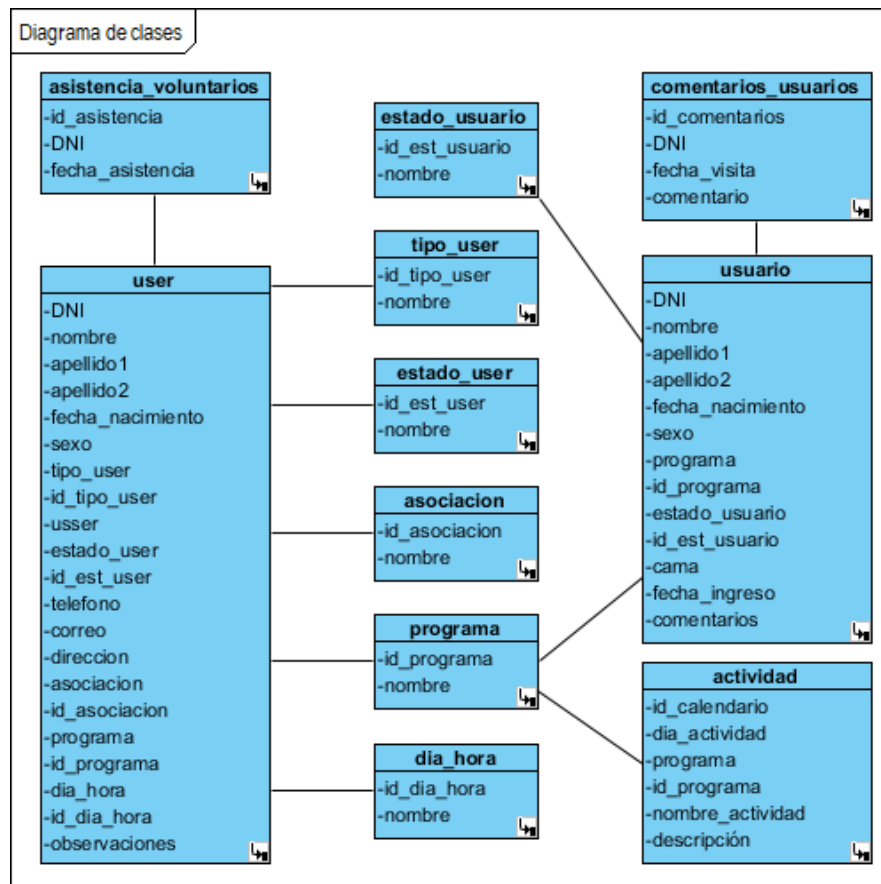


Figura 4-72 Diagrama de clases –Base Datos

La conexión entre las distintas clases se puede ver en el diagrama de clases. En él se puede apreciar que la clase user se conecta con la clase tipo\_user, esta relación es de la forma N:1, cada user sólo puede tener un tipo\_user para distinguir las opciones que se le presentan en la aplicación según si es administrador, personal o voluntario. La clase user tiene la misma relación con las clases estado\_user, asociación, programa y dia\_hora. Además de estas conexiones, la clase user se relaciona con la clase asistencia\_voluntarios de la forma 1:N puesto que cada user puede tener N asistencias registradas, cada una con un identificador distinto ya que dos entradas pueden tener varios atributos iguales pero nunca pueden tenerlos todos iguales.

Las conexiones de la clase usuario son similares a las de la clase user, puesto que se relaciona de la forma N:1 con las clases estado\_usuario y programa ya que sólo pueden pertenecer a un programa concreto y tener un estado diferenciado, y de la forma 1:N con la clase comentarios\_usuarios puesto que cada user puede tener N comentarios registrados, cada uno con un identificador distinto.

A su vez, la clase actividad se conecta con la clase programa de la forma N:1, una actividad sólo puede pertenecer a un programa.

### 4.2.3 Descripción de las tablas

En este apartado se realizará una descripción de las diferentes tablas que forman la base de datos y de los distintos campos que contienen cada una de ellas:

- **USER:** Almacena los datos de cada uno de los usuarios de la aplicación o user que están registrados en la base de datos, tiene asociado el tipo de usuario de la aplicación y el estado en el que se encuentra, la asociación a la que pertenece, el programa en el que realiza la actividad y el día en que la realiza. La tabla está compuesta por los siguientes campos:
  - **DNI:** identificador de cada uno de los user, se corresponde con el número del DNI del user en cuestión. Este campo es la clave principal de la tabla.
  - **NOMBRE:** nombre del user registrado en la base de datos.
  - **APELLIDO 1:** primer apellido del user registrado en la base de datos.
  - **APELLIDO 2:** segundo apellido del user registrado en la base de datos.
  - **FECHA\_NACIMIENTO:** fecha de nacimiento del user registrado en la base de datos.
  - **EDAD:** edad del user registrado en la base de datos
  - **SEXO:** sexo del user registrado en la base de datos, se representa con un número siendo 0-Masculino y 1-Femenino.
  - **TIPO\_USER:** asocia los tipos de user creados con la tabla de user.
  - **USSER:** nombre del user registrado en la base de datos con el que accederá a la aplicación
  - **ESTADO\_USER:** asocia los estados de user creados con la tabla de user.
  - **TELEFONO:** teléfono del user registrado en la base de datos.
  - **CORREO:** correo del user registrado en la base de datos.
  - **DIRECCION:** dirección del user registrado en la base de datos.
  - **ASOCIACION:** asocia las asociaciones creadas con la tabla de user.
  - **PROGRAMA:** asocia los programas creados con la tabla de user.
  - **DIA\_HORA:** asocia los turnos creados con la tabla de user.
  - **OBSERVACIONES:** almacena datos reseñables del user registrado en la base de datos que no corresponden con ninguno de los otros campos de la tabla.
  
- **TIPO\_USER:** Almacena el tipo de user de cada uno de los user de la base de datos, pueden presentar cualquiera de los tipos: administrador/personal/voluntario. La tabla está compuesta por los siguientes campos:
  - **ID\_TIPO\_USER:** identificador de cada uno de los tipos de user, se corresponde con un número entero, siendo 1-administrador, 2-personal y 3-voluntario. Este campo es la clave principal de la tabla.
  - **NOMBRE:** designación del tipo de user que tiene asociado su relativo identificador.

- **ESTADO\_USER:** Almacena el estado en el que se encuentra cada uno de los user en la base de datos, alta/baja. La tabla está compuesta por los siguientes campos:
  - **ID\_EST\_USER:** identificador de cada uno de los estados de los user, se corresponde con un número entero, siendo 1-activo y 2-baja. Este campo es la clave principal de la tabla.
  - **NOMBRE:** designación del estado del user que tiene asociado su relativo identificador.
  
- **ASOCIACION:** Almacena la asociación a la que está asociado cada uno de los user en la base de datos. La tabla está compuesta por los siguientes campos:
  - **ID\_ASOCIACION:** identificador de cada una de las asociaciones, se corresponde con un número entero. Este campo es la clave principal de la tabla.
  - **NOMBRE:** designación de la asociación que tiene asociado su relativo identificador.
  
- **PROGRAMA:** Almacena el programa al que está asociado cada uno de los user en la base de datos. La tabla está compuesta por los siguientes campos:
  - **ID\_PROGRAMA:** identificador de cada uno de los programas, se corresponde con un número entero. Este campo es la clave principal de la tabla.
  - **NOMBRE:** designación de la asociación que tiene asociado su relativo identificador.
  
- **DIA\_HORA:** Almacena el día y la hora a la que está asociado cada uno de los user en la base de datos. La tabla está compuesta por los siguientes campos:
  - **ID\_DIA\_HORA:** identificador de cada uno de los turnos, se corresponde con un número entero. Este campo es la clave principal de la tabla.
  - **NOMBRE:** designación del turno que tiene asociado su relativo identificador.
  
- **USUARIO:** Almacena los datos de cada uno de los usuarios que están registrados en la base de datos, tiene asociado el estado en el que se encuentra y el programa del que se beneficia. La tabla está compuesta por los siguientes campos:
  - **DNI:** identificador de cada uno de los usuarios, se corresponde con el número del DNI del usuario en cuestión. Este campo es la clave principal de la tabla.
  - **NOMBRE:** nombre del usuario registrado en la base de datos.
  - **APELLIDO 1:** primer apellido del usuario registrado en la base de datos.

- **APELLIDO 2:** segundo apellido del usuario registrado en la base de datos.
  - **FECHA\_NACIMIENTO:** fecha de nacimiento del usuario registrado en la base de datos.
  - **SEXO:** sexo del usuario registrado en la base de datos, se representa con un número siendo 0-Masculino y 1-Femenino.
  - **PROGRAMA:** asocia los programas creados con la tabla de usuarios.
  - **ESTADO\_USUARIO:** asocia los estados de usuarios creados con la tabla de usuarios.
  - **CAMA:** número de habitación y cama del usuario registrado en la base de datos para que el voluntario sepa a donde tiene que dirigirse.
  - **FECHA\_INGRESO:** fecha del último ingreso del usuario registrado en la base de datos.
  - **COMENTARIOS:** almacena datos reseñables del usuario registrado en la base de datos que no corresponden con ninguno de los otros campos de la tabla. Este comentario sólo puede modificarlo el personal.
- **ESTADO\_USUARIO:** Almacena el estado en el que se encuentra cada uno de los usuarios en la base de datos, ingreso/alta. La tabla está compuesta por los siguientes campos:
    - **ID\_EST\_USUARIO:** identificador de cada uno de los estados de los usuarios, se corresponde con un número entero, siendo 1-ingreso y 2-alta. Este campo es la clave principal de la tabla.
    - **NOMBRE:** designación del estado del usuario que tiene asociado su relativo identificador.
- **ASISTENCIA\_VOLUNTARIOS:** Almacena los datos de cada una de las asistencias de los voluntarios que están registrados en la base de datos. La tabla está compuesta por los siguientes campos:
    - **ID\_ASISTENCIA:** identificador de cada una de las asistencias registradas de los voluntarios, va aumentando en una unidad de manera ordenada por cada asistencia nueva que se registra en la base de datos. Este campo es la clave principal de la tabla.
    - **DNI:** asocia los DNI de los voluntarios registrados con la tabla de asistencias de los voluntarios.
    - **FECHA\_ASISTENCIA:** fecha de la asistencia de los voluntarios registrada en la base de datos.
- **COMENTARIOS\_USUARIOS:** Almacena los datos de cada uno de los acompañamientos que han recibido los usuarios y están registrados en la base de datos. La tabla está compuesta por los siguientes campos:
    - **ID\_COMENTARIOS:** identificador de cada uno de los acompañamientos a los usuarios registrados, va aumentando en una unidad de manera ordenada por cada asistencia nueva que se registra en la base de datos. Este campo es la clave principal de la tabla.



- **DNI:** asocia los DNI de los usuarios registrados con la tabla de comentarios de los usuarios.
  - **FECHA\_VISITA:** fecha del acompañamiento a los usuarios registrada en la base de datos
  - **COMENTARIO:** almacena los comentarios sobre los usuarios que los voluntarios que los han acompañado consideran reseñables. Este comentario sólo puede modificarlo el voluntario.
- 
- **ACTIVIDAD:** Almacena los datos de cada una de las actividades que están registrados en la base de datos, tiene asociado el programa al que corresponde. La tabla está compuesta por los siguientes campos:
    - **ID\_CALEDARIO:** identificador de cada una de las actividades registradas, va aumentando en una unidad de manera ordenada por cada actividad nueva que se registra en la base de datos. Este campo es la clave principal de la tabla.
    - **DIA\_ACTIVIDAD:** fecha de la actividad registrada en la base de datos.
    - **PROGRAMA:** asocia los programas creados con la tabla de actividades.
    - **NOMBRE\_ACTIVIDAD:** nombre que se le da a la actividad registrada en la base de datos para localizarla.
    - **DESCRIPCION:** breve descripción de la actividad registrada en la base de datos para explicar en que consiste.

#### 4.2.4 Desarrollo de la Base de datos

Para poder comenzar con el diseño y desarrollo de la base de datos, es necesario tener instalado previamente un servidor web. Apache distribuye el paquete de instalación XAMPP de manera gratuita. Dicho paquete es fácil de instalar configurado por defecto con todas las opciones activadas, y contiene MySQL, PHP y Perl. XAMPP tiene versiones para instalar en Windows, Linux y OS X [9].

Una vez instalado el servidor web, se crea la base de datos mediante la herramienta phpMyAdmin donde se realizará la administración de MySQL a través de la web, ubicado en la dirección *http://localhost/phpmyadmin*. Con esta herramienta se creará y editará las diferentes tablas necesarias para la aplicación que se desarrollará en este trabajo.

Cuando se accede a la dirección, se abrirá una ventana principal del administrador phpMyAdmin donde tendremos que registrarnos con un Usuario y una Contraseña [10] [11].



Figura 4-73 phpMyAdmin – Ventana principal

Una vez se ha accedido, se pulsará sobre la pestaña ‘Base de Datos’; donde aparecerá una nueva ventana para crear nuevas bases de datos. Para crear la nueva base de datos tan bastará con escribir su nombre, en este caso ‘gv’ y colocar un cotejamiento ‘utf8\_spanish\_c’ para evitar posibles errores futuros con algún carácter del alfabeto en español como puede ser las tildes y la letra ‘ñ’.

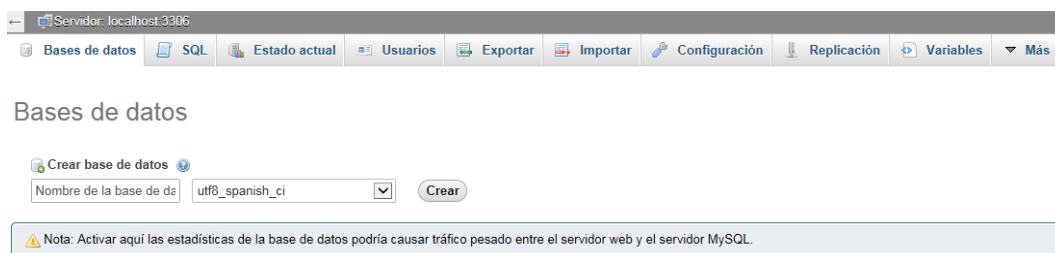


Figura 4-74 phpMyAdmin – Crear Base de Datos

Tras crear la base de datos, aparecerá directamente la opción de crear una tabla, puesto que la base de datos creada está vacía. Por defecto el número de columnas que creará en la tabla será de cuatro, aunque podemos cambiar dicho número según el tamaño que vaya a tener nuestra tabla.

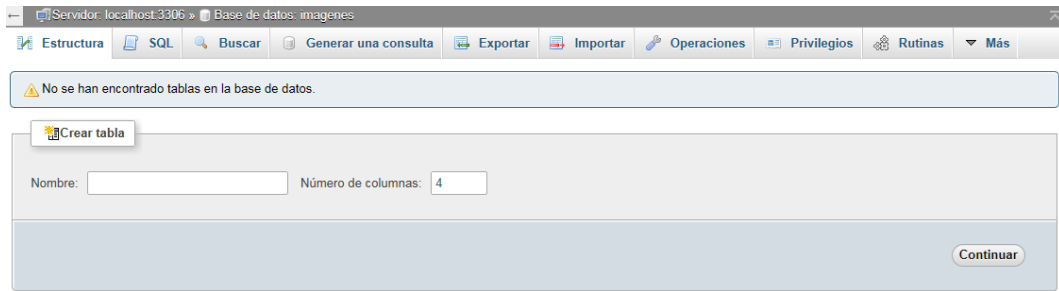


Figura 4-75 phpMyAdmin – Crear tabla

Una vez definido el nombre y el número de columnas de nuestra tabla se pulsará el botón ‘continuar’ lo que nos llevará a una ventana donde se crearán las distintas columnas de esa nueva tabla. Desde aquí podremos previsualizar como quedaría la tabla en SQL o simplemente guardarla una vez que se han definido todos los campos necesarios.

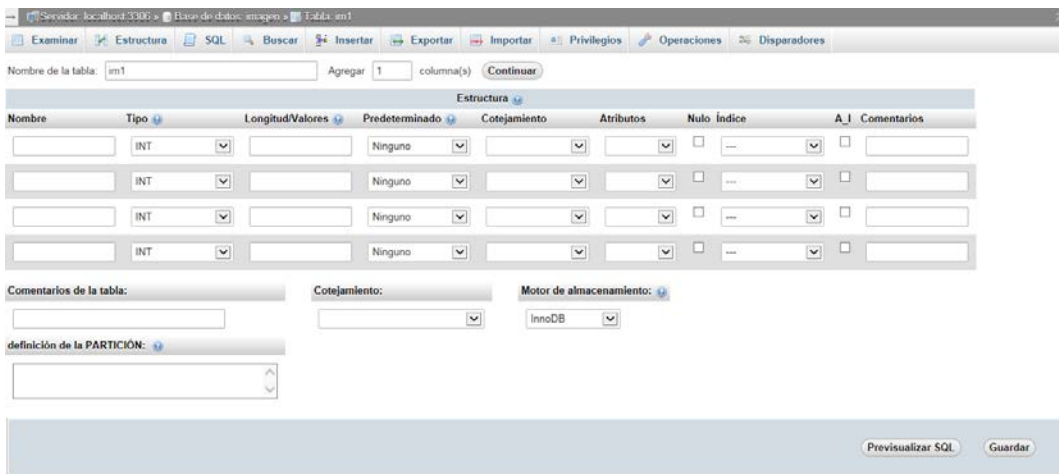


Figura 4-76 phpMyAdmin – Estructura tabla

Se podrán crear todas las tablas necesarias desde la página principal de la base de datos en cuestión donde se muestran las tablas ya creadas o desde el desplegable de la izquierda en la opción de ‘nueva’. Estas dos opciones nos llevarán al paso anterior donde definiremos el contenido de las distintas columnas de la tabla.

Esto puede ser hecho tantas veces como necesitemos dependiendo del número de tablas que requiera nuestra base de datos.

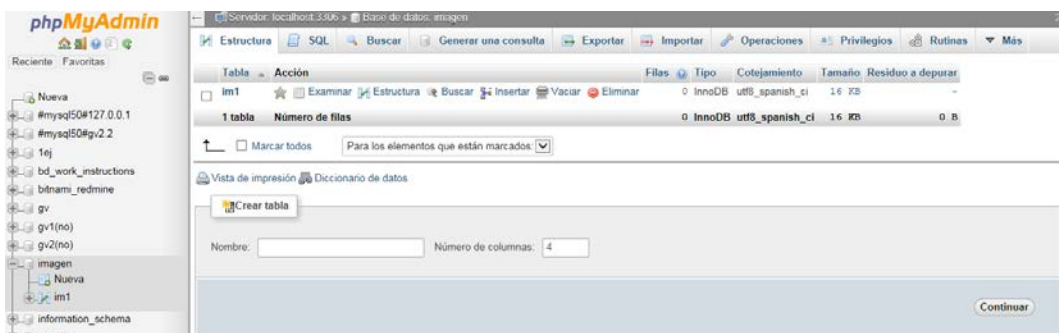


Figura 4-77 phpMyAdmin – Nueva tabla

Si se necesita acceder de nuevo a alguna de las tablas creadas después de guardar su configuración para editar alguna columna o añadir columnas nuevas, lo primero que se debe hacer es seleccionar la tabla que se encuentra a la izquierda de la herramienta phpMyAdmin. Una vez seleccionada la tabla, se selecciona la pestaña 'Estructura' situada en la parte superior de la herramienta, se marca el campo que se quiere configurar y se selecciona la opción de 'cambiar' o 'eliminar'. Si lo que se quiere es añadir una nueva columna solo es necesario añadir el número de columnas que se quiere agregar a la tabla, posición donde se va a crear ('al final de la tabla', 'al comienzo de la tabla' o 'después de campo') y se pulsa 'continuar'.

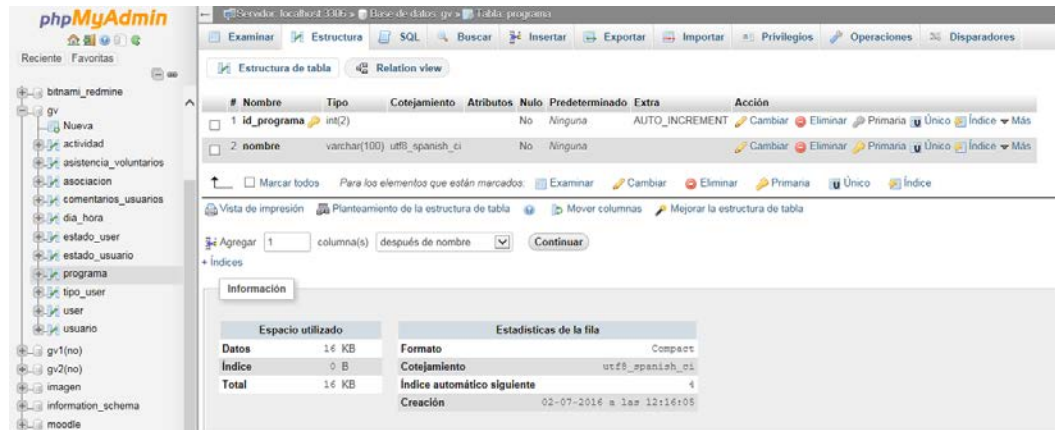


Figura 4-78 phpMyAdmin – Modificar tabla

La definición de los diferentes campos que aparecen al crear la tabla es:

- Nombre: nombre del campo.
- Tipo: tipo del valor del campo. Los tipos de valores que se pueden elegir:
  - Alfanumérico: "VARCHAR" o "CHAR".
  - Número entero: "INT"
  - Número decimal: "DECIMAL", "FLOAT", "DOUBLE" o "REAL".
  - Fecha: "DATE", "DATETIME", "TIMESTAMP", "TIME" o "YEAR".
  - Memo: "TEXT", "TINYTEXT", "MEDIUMTEXT" o "LONGTEXT". Dependiendo del tamaño que se quiere almacenar se elige un tipo u otro.
- Longitud/Valores: Tamaño máximo del campo. Por ejemplo, un texto de tamaño 10 sólo podrá tener 10 caracteres.
- Predeterminado: Asigna un valor predeterminado, en el caso de que no se rellene un campo.
- Cotejamiento: Conjunto de caracteres que tendrá el campo, siendo en este caso 'utf8\_spanish\_ci' por el motivo comentado anteriormente para la base de datos en general.
- Atributos: Dependientes del tipo de dato. Por ejemplo, que un numérico solo sea positivo.
- Nulo: El campo permite valores nulos si es marcado si es marcado como 'NULL' o si debe rellenarse obligatoriamente.
- Índice: Indica si el campo está indexado, donde se crea una clave primaria que identifica de manera única a cada registro.
- A\_I: Si el campo es un autoincremento, al crear nuevos registros, se aumenta el número del anterior en una unidad, así los registros quedan numerados.
- Comentarios: Permite agregar comentarios descriptivos del campo o su contenido

A continuación, se diseñarán cada una de las tablas creando los diferentes atributos que formarán la base de datos de esta aplicación:

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>DNI</i>	int(8)				No		
nombre	char(20)		Utf8_spanish_ci		No		
apellido1	char(20)		Utf8_spanish_ci		No		
apellido2	char(20)		Utf8_spanish_ci		No		
fecha_nacimiento	date	NULL			Sí		
edad	int(10)	NULL			Sí		
sexo	char(1)	NULL	Utf8_spanish_ci		Sí		
tipo_user	int(1)	3			No		
Usser	varchar(9)	NULL	Utf8_spanish_ci		Sí		
estado_user	int(1)	1	Utf8_spanish_ci		No		
telefono	int(9)	NULL			Sí		
correo	varchar(30)	NULL	Utf8_spanish_ci		Sí		
direccion	varchar(100)	NULL	Utf8_spanish_ci		Sí		
asociacion	int(2)	NULL			Sí		
programa	int(2)	NULL			Sí		
dia_hora	int(2)	NULL			Sí		
observaciones	varchar(600)	NULL	Utf8_spanish_ci		Sí		

Tabla 4-28 Estructura de USER

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_tipo_user</i>	int(1)	Ninguna			No	Si	
nombre	char(20)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-29 Estructura de TIPO\_USER

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_est_user</i>	int(1)	Ninguna			No	Si	
nombre	char(6)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-30 Estructura de ESTADO\_USER

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_asociacion</i>	int(2)	Ninguna			No	Si	
nombre	char(40)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-31 Estructura de ASOCIACION

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_programa</i>	int(2)	Ninguna			No	Si	
nombre	varchar(100)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-32 Estructura de PROGRAMA

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_dia_hora</i>	int(2)	Ninguna			No	Si	
nombre	varchar(20)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-33 Estructura de DIA\_HORA

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>DNI</i>	int(8)	Ninguna			No		
nombre	char(20)	Ninguna	Utf8_spanish_ci		No		
apellido1	char(20)	Ninguna	Utf8_spanish_ci		No		
apellido2	char(20)	Ninguna	Utf8_spanish_ci		No		
fecha_nacimiento	date	Ninguna			No		
sexo	char(1)	Ninguna	Utf8_spanish_ci		No		
programa	int(2)	Ninguna			No		
estado_usuario	int(1)	Ninguna			No		
cama	varchar(5)	Ninguna	Utf8_spanish_ci		No		
fecha_ingreso	date	Ninguna			No		
comentarios	varchar(300)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-34 Estructura de USUARIO

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_est_usuario</i>	int(2)	Ninguna			No	Si	
nombre	char(10)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-35 Estructura de ESTADO\_USUARIO

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_asistencia</i>	int(6)	Ninguna			No	Si	
DNI	int(8)	NULL			Sí		
fecha_asistencia	date	Ninguna	Utf8_spanish_ci		No		

Tabla 4-36 Estructura de ASISTENCIA\_VOLUNTARIOS

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_comentarios</i>	int(6)	Ninguna			No	Si	
DNI	int(8)	Ninguna			No		
fecha_visita	date	Ninguna			No		
comentario	varchar(600)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-37 Estructura de COMENTARIOS\_USUARIOS

Nombre	Tipo(Long)	Predeterm.	Cotejamiento	Atributos	Nulo	A_I	Coment.
<i>id_calendario</i>	int(11)	Ninguna			No	Si	
día_actividad	datetime	Ninguna			No		
programa	int(11)	Ninguna			No		
nombre_actividad	char(30)	Ninguna	Utf8_spanish_ci		No		
descripcion	char(100)	Ninguna	Utf8_spanish_ci		No		

Tabla 4-38 Estructura de ACTIVIDAD

Una vez configurada cada una de las tablas, se procederá a rellenar con datos los campos creados para posteriormente utilizarlos en la aplicación. Para ello se debe seleccionar la tabla que se encuentra a la izquierda de la herramienta phpMyAdmin. Una vez seleccionada la tabla, se selecciona la pestaña ‘insertar’ situada en la parte superior de la herramienta, esta pestaña contiene todas las columnas de la tabla en fila mostrando su tipo, una columna para seleccionar si es una función y una columna para introducir el valor. Podemos ir introduciendo las filas de una en una o elegir el número de filas que queremos insertar de una vez en la parte inferior de la pantalla. Se grabará la información al darle al botón de continuar. Cuando se rellene los diferentes campos habrá que tener en cuenta que los campos con autoincremento no se tendrán que rellenar

Figura 4-79 phpMyAdmin – Introducir fila

Una vez rellenas las tablas podremos modificar las distintas filas si lo necesitamos. Para ello se debe seleccionar la tabla que se encuentra a la izquierda de la herramienta phpMyAdmin. Una vez seleccionada la tabla, se selecciona la pestaña ‘examinar’ situada en la parte superior de la herramienta, esta pestaña muestra todas las filas de la tabla seleccionada y nos dará la opción de editar o borrar las filas, de una en una o por grupo.

Figura 4-80 phpMyAdmin – Modificar fila



A continuación, se muestra cada una de las tablas total o parcialmente:

DNI	nombre	apellido1	apellido2	fecha_nacimiento	edad	sexo	tipo_user	usser	estado_user	telefono	correo	direccion	asociacion	programa	dia_hora	observaciones
12345678	Madministrador	iglesias	caso	1980-01-01	NULL	1	1	mariglcas	1	123456789	5548	55698	2	1	1	44587
87654321	Mvoluntario	iglesias	caso	1981-02-02	NULL	0	3	mic	1	123456790	5549	55699	3	1	7	44588
98765432	Mpersonal	iglesias	caso	1982-03-03	NULL	1	2	person	1	123456791	5550	55700	4	3	3	44589
11112229	114	225	336	1980-04-01	NULL	1	3	696	1	123456792	5551	55701	5	1	4	44590
11112230	115	226	337	1980-05-01	NULL	1	3	697	1	123456793	5552	55702	2	2	5	44591
11112231	116	227	338	1980-06-01	NULL	0	3	698	1	123456794	5553	55703	3	3	6	44592
11112232	117	228	339	1980-07-01	NULL	1	3	699	1	123456795	5554	55704	4	1	7	44593
11112233	118	229	340	1980-08-01	NULL	1	3	700	1	123456796	5555	55705	5	2	8	44594
11112234	119	230	341	1980-09-01	NULL	0	3	701	1	123456797	5556	55706	2	3	9	44595
11112235	120	231	342	1980-10-01	NULL	1	3	702	1	123456798	5557	55707	3	1	1	44596
11112236	121	232	343	1980-11-01	NULL	1	3	703	1	123456799	5558	55708	4	2	2	44597
11112237	122	233	344	1980-12-01	NULL	0	3	704	1	123456800	5559	55709	5	3	3	44598
11112238	123	234	345	1980-01-01	NULL	1	3	705	1	123456801	5560	55710	2	1	4	44599
11112239	124	235	346	1980-02-01	NULL	1	3	706	1	123456802	5561	55711	3	2	5	44600
11112240	125	236	347	1980-03-01	NULL	0	3	707	1	123456803	5562	55712	4	3	6	44601
11112241	126	237	348	1980-04-01	NULL	1	3	708	1	123456804	5563	55713	5	1	7	44602
11112242	127	238	349	1980-05-01	NULL	1	3	709	1	123456805	5564	55714	2	2	8	44603
11112243	128	239	350	1980-06-01	NULL	0	3	710	1	123456806	5565	55715	3	3	9	44604
11112244	129	240	351	1980-07-01	NULL	1	3	711	1	123456807	5566	55716	4	1	1	44605
11112245	130	241	352	1980-08-01	NULL	1	3	712	1	123456808	5567	55717	5	2	2	44606
11112246	131	242	353	1980-09-01	NULL	0	3	713	1	123456809	5568	55718	2	3	3	44607
11112247	132	243	354	1980-10-01	NULL	1	3	714	1	123456810	5569	55719	3	1	4	44608
11112248	133	244	355	1980-11-01	NULL	1	3	715	1	123456811	5570	55720	4	2	5	44609
11112249	134	245	356	1980-12-01	NULL	0	3	716	1	123456812	5571	55721	5	3	6	44610
11112250	135	246	357	1980-01-01	NULL	1	3	717	1	123456813	5572	55722	2	1	7	44611
11112251	136	247	358	1980-02-01	NULL	1	3	718	1	123456814	5573	55723	3	2	8	44612
11112252	137	248	359	1980-03-01	NULL	0	3	719	1	123456815	5574	55724	4	3	9	44613
11112253	138	249	360	1980-04-01	NULL	1	3	720	1	123456816	5575	55725	5	1	1	44614
11112254	139	250	361	1980-05-01	NULL	1	3	721	1	123456817	5576	55726	2	2	2	44615
11112254	139	250	361	1980-05-01	NULL	1	3	721	1	123456817	5576	55726	2	2	2	44615
11112255	140	251	362	1980-06-01	NULL	0	3	722	1	123456818	5577	55727	3	3	3	44616
11112256	141	252	363	1980-07-01	NULL	1	3	723	1	123456819	5578	55728	4	1	4	44617
11112257	142	253	364	1980-08-01	NULL	1	3	724	1	123456820	5579	55729	5	2	5	44618
11112258	143	254	365	1980-09-01	NULL	0	3	725	1	123456821	5580	55730	2	3	6	44619
11112259	144	255	366	1980-10-01	NULL	1	3	726	1	123456822	5581	55731	3	1	7	44620
11112260	145	256	367	1980-11-01	NULL	1	3	727	1	123456823	5582	55732	4	2	8	44621
11112261	146	257	368	1980-12-01	NULL	0	3	728	1	123456824	5583	55733	5	3	9	44622
11112262	147	258	369	1980-01-01	NULL	1	3	729	1	123456825	5584	55734	2	1	1	44623

Tabla 4-39 Volcado parcial de USER

DNI	nombre	apellido1	apellido2	fecha_nacimiento	sexo	programa	estado_usuario	cama	fecha_ingreso	comentarios
11111111	350011	3500	3500	1984-04-04	1	2	2	337	2017-05-13	djdydikj
19874560	334	334	334	2017-04-20	1	3	2	112	2017-06-20	654654
19874561	3344	3345s	334d	0000-00-00		2	1	102	0000-00-00	řlkřlk
19874562	334444	334l	334k	0000-00-00		2	1	102	0000-00-00	llkkllř
19874563	335h	3355	335	1998-01-04	0	1	1	101	2017-02-14	lkji
19874564	336	3365	336	1979-06-07	1	1	1	314	2017-06-04	řlkpokl
19874565	337	337	337	1982-03-03	0	1	1	555	2017-02-07	lpřl
19874566	338	3388	338	0000-00-00		1	1	104	0000-00-00	sksju
19874567	339	339	339	1984-05-05	1	1	2	424	2017-02-14	jdudid
19874568	340	340	340	1985-06-06	0	2	1	502	2017-01-31	judkd
19874569	341	341	341	1986-07-07	0	2	2	332	2017-02-19	kdjuis
19874570	3422	3422	342	1987-08-08	0	3	1	113	2017-02-13	djudik
19874571	343	343	343	1988-09-09	0	2	1	208	2017-02-06	jdudld
19874572	344	344	344	1989-10-10	1	3	2	455	2017-02-14	djududik
19874573	3455	345	345	1990-11-11	1	3	1	114	2017-02-14	djhdydud
19874574	346	346	346	1991-12-12	1	2	1	209	2017-02-06	djdudi
19874575	347	347	347	1980-01-01	1	2	1	333	2017-02-13	hdydudj
19874576	348	348	348	1990-01-04	1	1	1	115	2017-02-14	ddydu
19874577	3499	349	349	2017-06-15	1	1	1	210	2017-06-14	dhdydu
19874578	3500	3500	3500	2017-06-15	0	0	0	337	2017-06-15	llřkl
19874579	351	351	351	1984-05-05	0	1	2	425	2017-02-05	dhdhydu
19874580	352	352	352	1985-06-06	0	2	1	211	2017-02-05	ddudid
19874581	353	353	353	1986-07-07	0	3	1	335	2016-12-05	dhdyu
19874582	354	354	354	1987-08-08	0	1	1	116	2017-01-17	dhdhydu
19874583	355	355	355	1988-09-09	0	2	1	212	2017-01-16	kdiddo
19874584	356	356	356	1989-10-10	1	3	2	456	2017-01-17	djudud
19874585	357	357	357	1990-11-11	1	1	1	117	2017-06-08	řlkřlk
19874586	358	358	358	1991-12-12	0	2	1	213	2017-06-04	poipok
19874587	359	359	359	1980-01-01	0	3	1	336	2017-06-22	řlkpo
19874588	360	360	360	1981-02-02	0	1	1	118	2017-06-21	lkjo
19874589	361	361	361	1982-03-03	0	2	1	214	2017-06-20	řpk
19874590	362	362	362	1983-04-04	1	3	1	337	2017-06-19	řpl
19874591	363	363	363	1984-05-05	0	1	2	426	2017-06-24	lpo
19874592	364	364	364	1985-06-06	0	2	1	215	2017-06-25	lpřo
19874593	365	365	365	1986-07-07	1	3	1	338	2017-06-11	přlo
19874594	366	366	366	1987-08-08	1	1	1	119	2017-06-13	lpoř
19874595	367	367	367	1988-09-09	0	2	1	216	2017-06-19	řplř

Tabla 4-40 Volcado parcial de USUARIO

id_tipo_user	nombre
0	TIPO USER
1	ADMINISTRADOR
2	PERSONAL
3	VOLUNTARIO

Tabla 4-41 Volcado completo de TIPO\_USER

id_est_user	nombre
0	ESTADO
1	ALTA
2	BAJA

Tabla 4-42 Volcado completo de ESTADO\_USER

id_asociacion	nombre
0	ASOCIACION
1	hospital
2	ADIMA
3	Cruz Roja
4	CRJ
5	AECC
6	Universidad de Sevilla
7	ASAENES

Tabla 4-43 Volcado completo de ASOCIACION

id_programa	nombre
0	PROGRAMA
1	Diver
2	Unidad de Donacion de Tiempo Libre
3	Ocupación de Tiempo Libre para Ninos Hospitalizados

Tabla 4-44 Volcado completo de PROGRAMA

id_dia_hora	nombre
0	DIA
1	Lunes Mañana
2	Lunes Tarde
3	Martes Mañana
4	Martes Tarde
5	Miércoles Mañana
6	Miércoles Tarde
7	Jueves Mañana
8	Jueves Tarde
9	Viernes Mañana
10	Viernes Tarde
11	Sábado Mañana
12	Sábado Tarde
13	Domingo Mañana
14	Domingo Tarde

Tabla 4-45 Volcado completo de DIA\_HORA

id_est_usuario	nombre
0	ESTADO
1	INGRESO

Tabla 4-46 Volcado completo de ESTADO\_USUARIO

id_asistencia	DNI	fecha_asistencia
1	10001111	2017-04-05
2	12345678	2017-04-05
3	10001111	2017-04-12
4	10001111	2017-04-04
5	12345678	2017-04-04
7	11112226	2017-06-20
8	11112226	2017-06-13
9	11112226	2017-06-11

10	11112226	2017-06-18
11	11112341	2017-06-27
12	11112226	2017-06-27
13	11112398	2017-08-26
14	11112398	2017-09-06

Tabla 4-47 Volcado completo de ASISTENCIA\_VOLUNTARIOS

id_comentarios	DNI	fecha_visita	comentario
1	19874560	2017-05-10	ñññlp
2	19874560	2017-05-11	pol
3	19874560	2017-05-12	kjhguy
4	19874561	2017-05-08	lkjg
5	19874561	2017-05-10	lkjgouy
6	19874561	2017-05-11	lhguyf
7	19874561	2017-05-14	lhgutfgh
8	19874561	2017-05-16	kjhfiyftg
9	19874563	2017-05-01	kjhfiyt
10	19874563	2017-05-03	jfdutrck
11	19874598	2017-05-02	ñljh
12	19874598	2017-05-04	ljhf
13	19874598	2017-05-11	lkhf
14	19874598	2017-05-12	lkjg
15	19874599	2017-05-15	jhfyt
16	19874599	2017-05-16	kjgdiyr
17	19874600	2017-05-16	lkhfiyt
18	19874600	2017-05-17	lhkft
19	19874600	2017-05-18	ljdyi
20	19874598	2017-05-17	ljhfiyt
21	19874563	2017-06-19	kaaa
22	19874563	2017-06-19	kklokj
23	19874563	2017-06-12	adadad

Tabla 4-48 Volcado completo de COMENTARIOS\_USUARIOS

id_calendario	dia_actividad	programa	nombre_actividad	descripcion
1	2017-03-03 00:00:00	1	ll	pkn
2	2016-09-08 00:00:00	2	ll	pkn
3	2016-10-06 00:00:00	1	pn	pknh
4	2016-10-12 00:00:00	2	pkn	ojnl
5	2017-08-13 00:00:00	3	mmimmm	mmiim
6	2016-12-10 00:00:00	1	REAL BETIS BALOMPIE	17h0 HUVR, Asociación Amigos del Centenario
7	2016-12-12 00:00:00	1	Buscando la Navidad	11:00 HUVM, Felicitacion del equipo de Todo por una Sonrisa
8	2016-12-12 00:00:00	1	Buscando la Navidad	17h00 HUVM, Felicitacion del equipo de Todo por una Sonrisa
9	2016-12-10 00:00:00	1	Festival ANDEX	17h00 HUVR, Salón de actos Hospital de la Mujer
10	2016-12-14 00:00:00	1	CD La Hispalense	17h00 HUVR, Mago, Regalos, etc
23	2017-03-14 00:00:00	1	Todo por una Sonrisa	18h00 HUVM
24	2017-03-07 00:00:00	1	Coro Campanilleros	17h00 HUVR
25	2017-03-16 00:00:00	1	Caravana de la Salud	10h00 HUVR
26	2016-12-14 00:00:00	1	Sana, sana, cuentito con ganas	17h00 HUVM, Cuentos con Carloco
27	2016-12-15 00:00:00	1	DIA DE MEXICO	11h30 HUVR, Mariachis Mexicanos
28	2016-12-15 00:00:00	1	Felicitacion de Navidad	17h30 HUVM, Plan integral para el fomento de la lectoescritura de la Universidad de Sevilla
29	2016-12-15 00:00:00	1	Grupo Joven Ayuntamiento El Ar	18h00 HUVR, Títeres, manualidades, cuentacuentos, etc
30	2016-12-15 00:00:00	1	Fundacion Theodora Hospital In	18h30 HUVR

Tabla 4-49 Volcado completo de ACTIVIDAD

Cuando se ha terminado de cargar todos los datos en las distintas tablas, solo faltaría conectar la base de datos a la aplicación web, y para ello será necesario cargar las carpetas de la aplicación que contiene los archivos php y html en el directorio donde esté instalado la herramienta phpMyAdmin. En este caso el directorio donde está localizado todos los archivos es: C:\Bitnami\redmine-3.1.0-0\mysql\data\gv.

Más adelante se explicará como a través del lenguaje php se llama a la base de datos para leer y editar en los distintos campos de las tablas de la base de datos.

### 4.3 Diseño de la interfaz de usuario

En este apartado se definirá la interfaz de usuario de la aplicación, medio con el que se comunicará con la máquina, dispositivo móvil o servidor web dependiendo del subsistema. La interfaz de una aplicación es uno de los puntos más importantes del sistema para los usuarios del sistema por lo que se debe ser sencilla y de fácil comprensión.

#### 4.3.1 Arquitectura del sistema

La arquitectura elegida para la aplicación es cliente-servidor de tres capas, que está compuesto por tres componentes:

- Capa de presentación - Navegador web
- Capa de negocio - Servidor web **Apache PHP**
- Capa de datos - Servidor base de datos **MySQL**

Esto significa que la arquitectura generalmente está compartida por:

- Cliente es el equipo que solicita los recursos, equipado con una interfaz de usuario (navegador web) para la presentación. En el cliente se ejecutará el código **JavaScript**, mientras el código **PHP** se ejecutará en el servidor devolviendo éste las páginas formateadas en **HTML**, que es el lenguaje que comprende la entidad cliente.
- Servidor de aplicaciones es el software del servidor web cuya tarea es proporcionar los recursos solicitados, requiere de otro servidor para hacerlo, en este caso del servidor de datos.
- Servidor de datos es el servidor que proporciona al servidor de aplicaciones los datos que requiere. El servidor de base de datos **MySQL** es el que proporcionará los datos solicitados por el servidor de aplicación. El servidor de aplicaciones sólo se puede comunicar con la base de datos a través de un controlador que actúa de intermediario con la base de datos [12].

A continuación, se detallarán los diferentes lenguajes de programación y las herramientas utilizadas para el diseño y desarrollo de la aplicación.

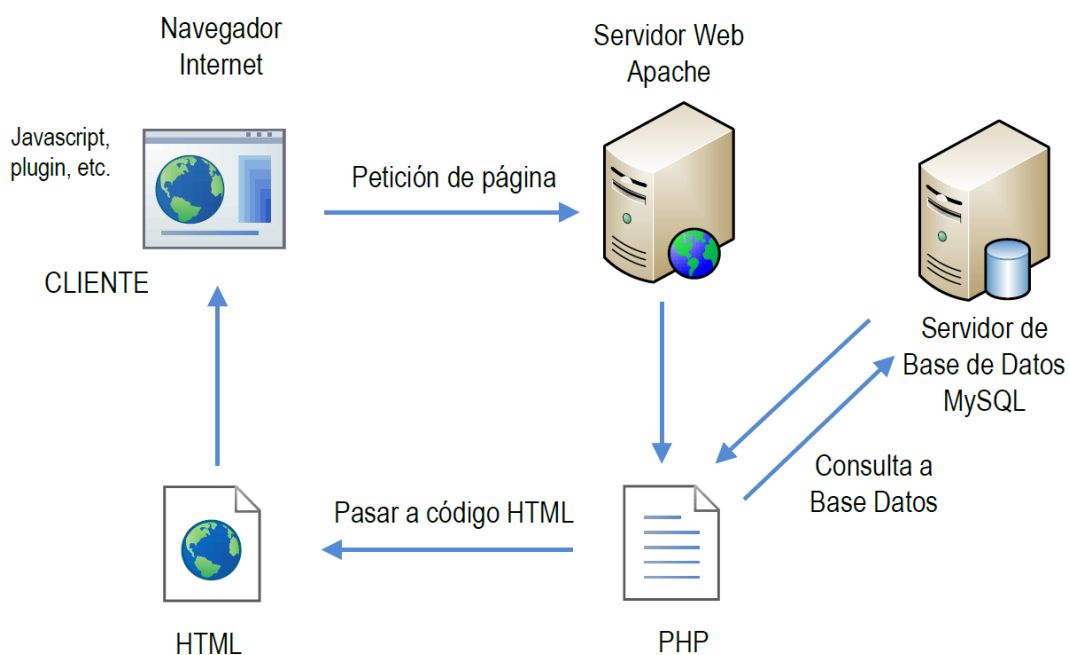


Figura 4-81 Arquitectura Cliente-Servidor

El funcionamiento que presenta la arquitectura de tres capas del sistema es:

1. El navegador web solicita la página dinámica.
2. El servidor web localiza la página y la envía al servidor de aplicaciones.
3. El servidor de aplicaciones busca instrucciones en la página.
4. El servidor web envía la consulta al controlador de la base de datos.
5. El controlador ejecuta la consulta en la base de datos.
6. El juego de registros se devuelve al controlador.
7. El controlador pasa el juego de registros al servidor de aplicaciones.
8. El servidor de aplicaciones inserta los datos en una página y luego pasa la página al servidor web.
9. El servidor web envía la página finalizada al navegador solicitante [13].

### 4.3.2 HTML

HTML, HyperText Markup Language, es uno de los lenguajes más usado para la elaboración de páginas web. Se utiliza para describir la estructura de la página y el contenido en forma de texto, aplicando etiquetas de formato que indican cómo se presenta el documento y cómo se vincula a otros documentos. Es un estándar, compuesto por recomendaciones publicadas por un consorcio internacional: el World Wide Web Consortium (W3C).

Está desarrollado mediante referencias, para la inserción de un elemento en la página, como por ejemplo una imagen, video, script, etc.; se hace referencia a la ubicación de dicho elemento mediante texto, en vez de introducirlo directamente en el código de la página. Esto hace que al final se obtenga una página web solo texto donde el navegador web es el encargado de interpretar el código del lenguaje HTML y unir todos los elementos y visualizar la página final.

HTML pretende ser un lenguaje que pueda ser interpretado de la misma manera por cualquier navegador web actualizado. Sin embargo, las diferentes versiones han incorporado y suprimido diversas características para hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (ordenadores, portátiles, smartphones, tabletas, etc.). Por ello, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Si el navegador no está actualizado, no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior [14].

### 4.3.3 PHP

PHP, Hypertext Preprocessor, es un lenguaje de programación de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incluido en HTML, es decir, las páginas de PHP pueden contener lenguaje HTML. El código PHP es ejecutado en el servidor y el resultado enviado al navegador, normalmente una página HTML.

Debido a la constante evolución, actualmente PHP incluye una interfaz de línea de comandos que puede ser usada en aplicaciones de sesiones gráficas independientes, en la mayoría de los servidores web, en casi todos los sistemas operativos y en diversas plataformas, sin coste alguno.

Actualmente, es considerado uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico a hacer uso de él, como tecnología de servidor. Sigue siendo desarrollado con nuevas funciones por el grupo PHP desde su creación [15] [16].



#### 4.3.4 CSS

CSS, Cascading Style Sheets, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. Permite controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML [17].

#### 4.3.5 Javascript

Javascript es un lenguaje de programación del lado del cliente (client-side) más utilizado, implementado como parte de un navegador web que permite mejorar la interfaz de usuario y páginas web dinámicas. Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript, orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo.

Permite crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, con que cuenta este lenguaje es el propio navegador y todos los elementos que hay dentro de una página.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. JavaScript de Netscape también presenta ciertas diferencias con Jscript, siendo este último una implementación de ECMAScript de Microsoft creado para evitar problemas relacionados con la marca. Debido a las incompatibilidades que pueden presentar ambos dialectos, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM) que añaden Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera la versión 7, Mozilla Application Suite y Mozilla Firefox desde su primera versión.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

En las páginas web HTML se utilizaba Javascript para realizar operaciones en el lado de las aplicaciones cliente, sin acceso a funciones del servidor. JavaScript se interpreta en la aplicación cliente al mismo tiempo que las sentencias van descargándose junto con el código HTML [18].

#### 4.3.6 MySQL

MySQL, es el sistema de gestión de bases de datos SQL, Structured Query Language, Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB, una compañía comercial fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio.

MySQL proporciona un servidor de base de datos muy rápido, fiable, fácil de usar, multi-threaded, multitud de usuarios y robusto, que permite trabajar en entornos cliente/servidor. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo, así como para integrarse en software para ser distribuidos [19].

### 4.3.7 Librerías Javascript

En las aplicaciones web suelen usarse librerías puesto que proporcionan algunas ventajas a la hora de programar:

- Cuando la aplicación es grande, ayudan a estructurar el código de forma ordenada.
- Si existe mucha interacción del usuario en el navegador, algunas herramientas nos ayudan a lidiar con ello de forma sencilla.

Sin embargo, su uso también plantea algunas desventajas:

- Aumenta la cantidad de bytes que debe descargar la página, aumentado su tiempo de carga.
- Para las cosas sencillas, es posible que más que ayudar complique la codificación.
- Javascript es un entorno muy cambiante, una librería equivocada puede quedarse sin soporte en el futuro.

Puesto que Javascript sigue desarrollándose, el número de librerías que puede usar sigue aumentando, haciendo prácticamente imposible conocerlas todas. Algunas de las más populares son [20]:

- **Angular.js:** es la librería MVW (Model View WhatEver) de Google que se ha encargado de traer orden a las aplicaciones JavaScript y potenciar las Arquitecturas SPA.
- **React.js:** es la librería de Facebook orientada a la gestión de del UI. Usa los conceptos de DataFlows y programación Reactiva para simplificar el proceso de actualización de la vista.
- **Meteor.js:** es una librería que apoya el concepto de aplicaciones JavaScript Isomórficas, aquellas aplicaciones que pueden ejecutar su código JavaScript en cliente y servidor.
- **Moment.js:** es una librería orientada a la gestión de fechas, consigue que el manejo de variables temporales sea algo mucho más sencillo.
- **jQuery.js:** es una de las librerías clásicas, standard a la hora de manipular el árbol DOM. Hoy por hoy su uso sigue estando muy presente, aunque alternativas como Angular enfoquen ya de otra manera.
- **Foundation.js:** es una de las librerías más avanzados a la hora de diseñar soluciones responsive, tiene un fuerte uso de Media Queries y un enfoque Mobile First.
- **BackBone.js:** es una de las librerías JavaScript MVC más utilizados que se apoya en Underscore y jQuery a la hora de abordar soluciones JavaScript y no tiene una curva de aprendizaje tan elevada como Angular.
- **Modernizr:** es la librería que se encarga de validar si nuestro navegador soporta una u otra característica de HTML 5, permitiendo a los desarrolladores actuar en consecuencia. Recordemos que el soporte de HTML5 aunque va en aumento siempre depara sorpresas.
- **Video.js:** es la librería de referencia para la gestión de Videos a través de JavaScript. El API que soporta es muy completa.

Para nuestro proyecto se usarán las librerías jQuery y jQWidgets. jQWidgets representa un marco basado en jQuery para crear aplicaciones basadas en web que se ejecutan en PC, Touch y dispositivos móviles. jQWidgets incluye más de 60 widgets de interfaz de usuario. jQWidgets no es una versión modificada de las herramientas de interfaz de usuario jQuery. Todos los widgets están diseñados desde cero y se basan en un potente núcleo común. El núcleo de referencias proporciona capacidades fundamentales como soporte para extensiones de widgets y herencia, configuraciones de widgets, manejo y enrutamiento de eventos internos, notificaciones de cambio de propiedades,

## 4.4 Herramientas

Las páginas HTML se pueden editar con el propio Bloc de notas u otro editor de texto que presenta Microsoft Windows. Sin embargo, existen editores de HTML que permiten su desarrollo de manera más estructurada y más fácil de manejar para los autores, clasificados en dos tipos:

- Editores que usan etiquetas que permiten editar las páginas utilizando el propio lenguaje HTML.
- Editores que diseñan una página web escribiendo con un editor de texto avanzado, sin tener que escribir el código con las etiquetas, donde es el propio software el encargado de programar internamente la página con el código HTML.

Algunos de los editores más destacados para desarrollar HTML son [21]:

- **Notepad ++**, es un editor de texto y de código fuente gratuito, capaz de soportar varios de lenguaje de programación como puede ser C++, SQL, XML, ASP, etc. que solo puede ejecutarse en Microsoft Windows. La manera de editarse es parecida al *Bloc de notas* ya que puede editarse sin formato y de forma simple, pero con herramientas y opciones más avanzadas [22].

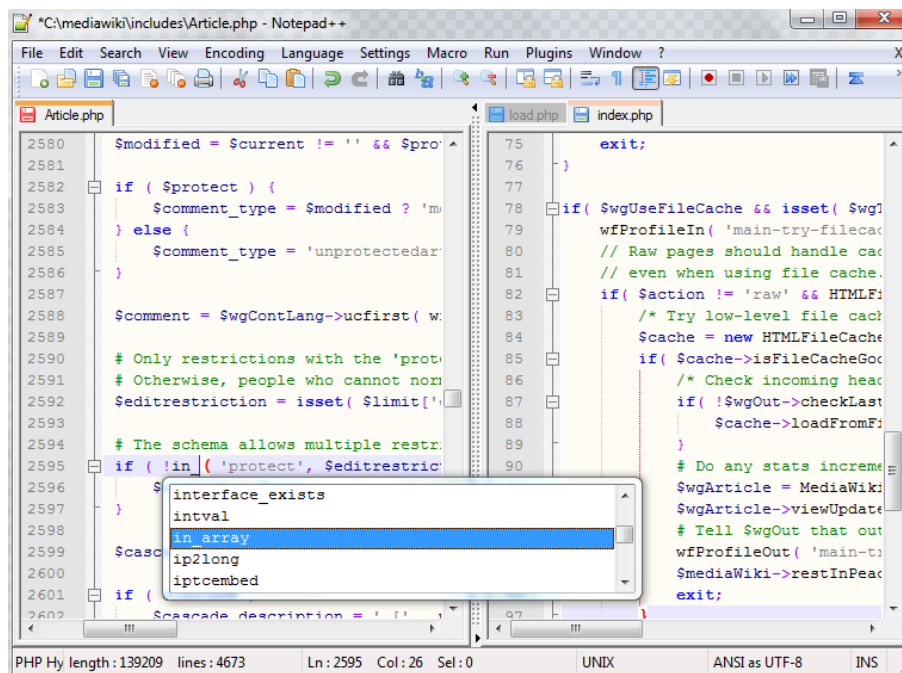


Figura 4-82 Ejemplo de Notepad ++

- **KompoZer**, es un sistema de creación de páginas web gratuita, que combina la gestión de archivos y la edición de páginas WYSIWYG de una manera fácil de usar. Las capacidades WYSIWYG, *What You See Is What You Get* (“lo que ver es lo que obtienes”), son una de las principales atracciones del software, que además permite la edición directa del código, así como una opción de vista dividida de código gráfico. Está diseñado para ser extremadamente fácil de usar, por lo que es ideal para los usuarios no técnicos informáticos que quieren crear una, que presenta un sitio web con aspecto profesional atractivo, sin necesidad de saber HTML o codificación web [23] [24].

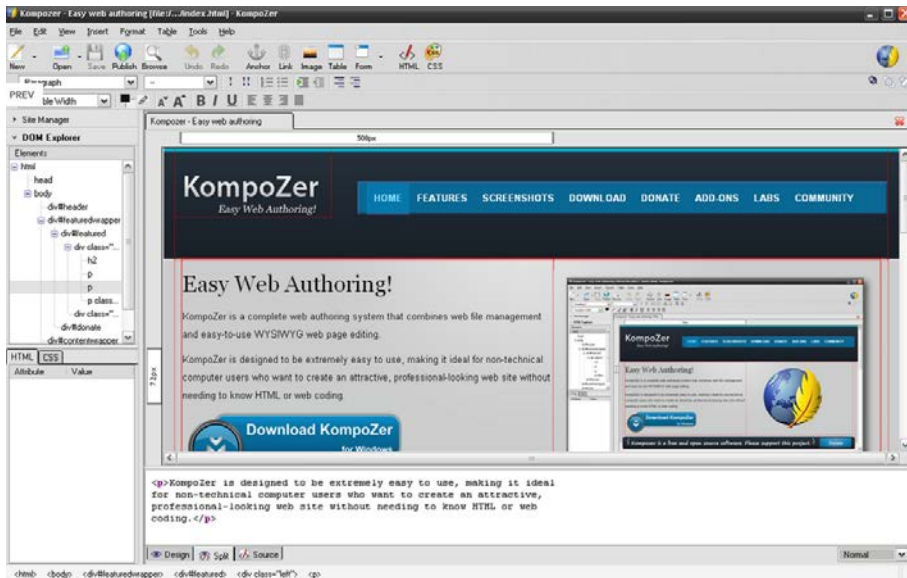


Figura 4-83 Ejemplo de KompoZer

- **Adobe Dreamweaver CS6**, es una herramienta de diseño web de la compañía de software Adobe Systems Incorporated, destinada a la construcción, diseño y edición de sitios, vídeos y aplicaciones Web basados en estándares. Es capaz de crear páginas dinámicas sin escribir una sola línea de programación de una forma muy sencilla y visual, por lo que es uno de los programas más utilizados en el sector del diseño y la programación web por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium [25].

Algunas de las características que presenta este software son:

- Soportan otras tecnologías web como CSS, JavaScript y frameworks del lado servidor.
- Permite ocultar el código HTML de cara al usuario, haciendo que se pueda crear páginas y sitios web sin necesidad de escribir código.
- Permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para pre-visualizar las páginas web.
- Presentan una gran capacidad de extensión y personalización, ya que la mayoría de las acciones que se realizan en editores de HTML, como puede ser insertar imágenes, hipervínculos, etc., se realizan en Javascript-C, haciendo que los archivos del programa no sean instrucciones de C++. Las extensiones, son pequeños programas, que cualquier desarrollador web puede escribir y que cualquiera puede descargar e instalar, ofreciendo así funcionalidades añadidas a la aplicación
- Permite la conexión a Bases de Datos como MySQL y Microsoft Access, para filtrar y mostrar el contenido utilizando tecnología de script como, por ejemplo, ASP, ASP.NET, ColdFusion, JSP y PHP sin necesidad de tener experiencia previa en programación.
- Permite diseñar un menú u otras creaciones de imágenes en un sitio web y después exportar la imagen creada y así utilizarla como una sola en donde ya llevará los vínculos para dicho sitio.

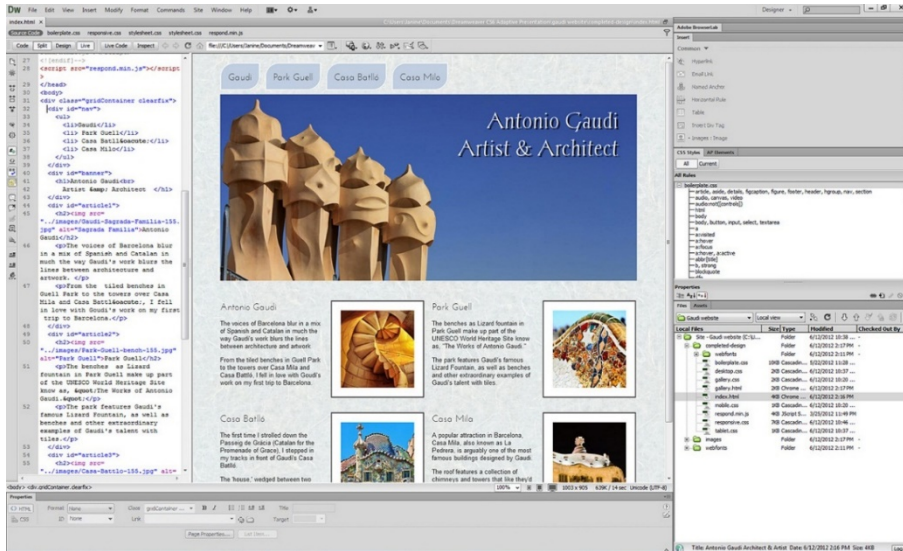


Figura 4-84 Ejemplo de Dreamweaver CS6

- **phpMyAdmin**, es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.

Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas [10] [11].

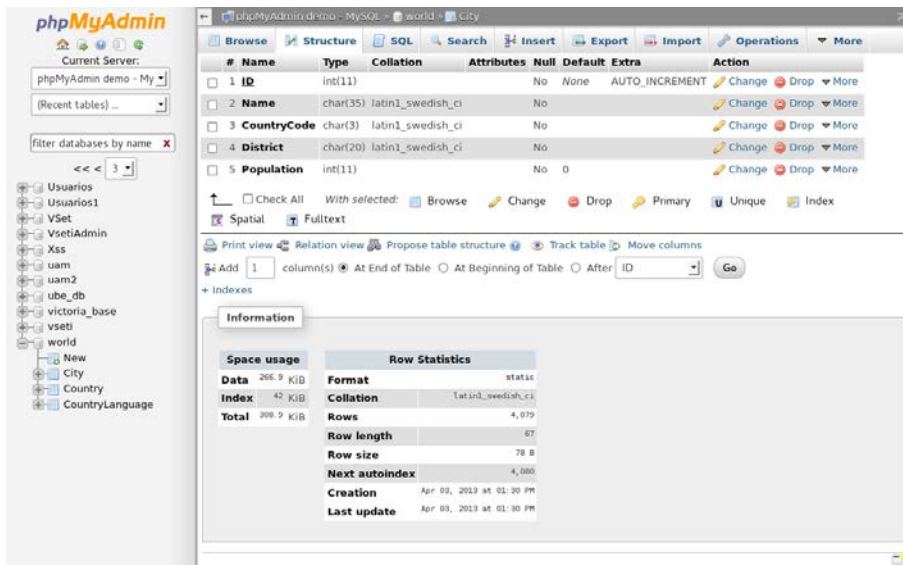


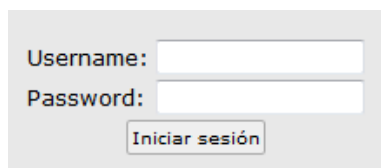
Figura 4-85 Ejemplo de phpMyAdmin

## 4.5 Sistema de Gestión colaborativo

A continuación, se mostrará el código utilizado en todas las páginas creadas para la interfaz de la aplicación web, escribiendo una parte del código utilizado con una figura para mostrar la interfaz obtenida.

### 4.5.1 Acceder al sistema [acceso.php]

Al introducir la dirección `http://localhost:8080/GV/PFC/CapaUser/Login/acceso.php` en el navegador web, se cargará la primera página de la aplicación que corresponde con la página de acceso al sistema. El diseño y los elementos descritos en el apartado Prototipo Login para la página de acceso al sistema, se ha conseguido mediante el pseudocódigo que está en la dirección `\GV\PFC\CapaUser\Login` y que se detalla a continuación:



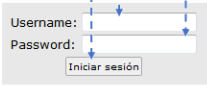
The image shows a simple login form on a light gray background. It consists of two text input fields: the first is labeled 'Username:' and the second is labeled 'Password:'. Below these fields is a button with the text 'Iniciar sesión'.

Figura 4-86 Interfaz Acceso

```

<body id="principal">
  <div id="DatosUser" style="position: absolute; top:50%; left: 50%; margin-left: -100px;">
    <table>
      <tr>
        <td>Username:</td>
        <td><input style="width: 150px;" type="text" id="user" name="user" /></td>
      </tr>
      <tr>
        <td>Password:</td>
        <td><input style="width: 150px;" type="password" id="password" name="password" /></td>
      </tr>
      <tr>
        <td colspan="2" align="center" valign="bottom">
          <input type="button" id="IniciarSesion" value="Iniciar sesión" />
        </td>
      </tr>
    </table>
  </div>
</body>

```



The diagram shows the HTML code on the left and the visual login interface on the right. Blue dashed lines connect the code elements to their corresponding visual elements: the text labels 'Username:' and 'Password:' are connected to the text input fields; the button code is connected to the 'Iniciar sesión' button.

Figura 4-87 Acceso.php

#### 4.5.2 User correcto [userCorrecto.php]

Una vez iniciada la sesión, y según el tipo de user registrado, aparecerá la pantalla de inicio del user. En el caso del Administrador el diseño y los elementos descritos en el apartado Prototipo Administrador para la página de acceso al sistema, se ha conseguido mediante el pseudocódigo que se detalla a continuación:

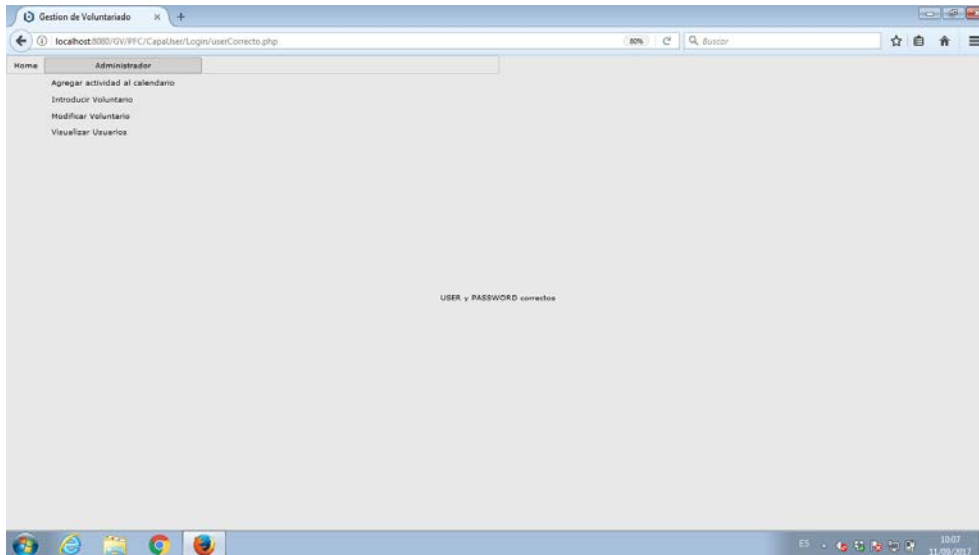


Figura 4-88 Interfaz Administrador



Figura 4-89 userCorrecto.php - Administrador

Análogamente, si el tipo user es Personal lo encontraremos en Prototipo Personal y si el tipo de user es Voluntario lo encontraremos en Prototipo Voluntario.

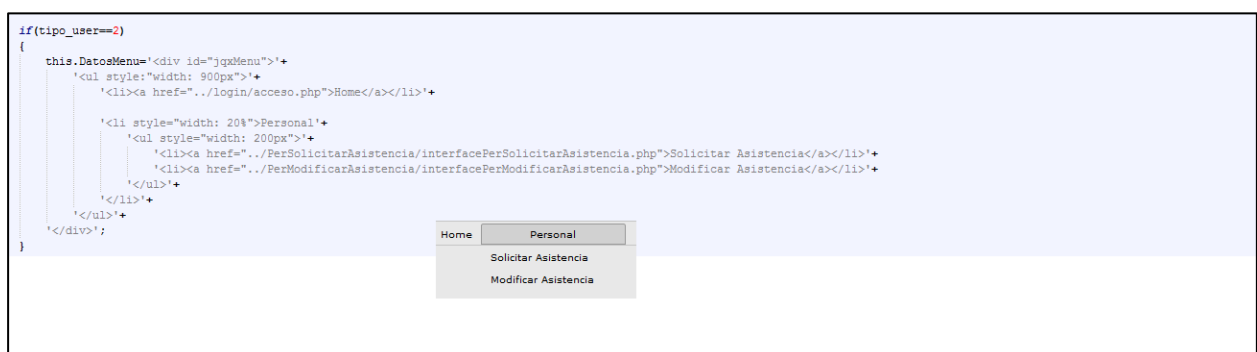


Figura 4-90 userCorrecto.php - Personal

```

if(tipo_user==3)
{
  this.DatosMenu='<div id="jqxMenu">'+
  '<ul style="width: 900px">'+
  '<li><a href=" ../Login/acceso.php">Home</a></li>'+
  '<li style="width: 30%">Voluntario'+
  '<ul style="width: 300px">'+
  '<li><a href=" ../VolVisualizarCalendario/interfaceVolVisualizarCalendario.php">Visualizar calendario</a></li>'+
  '<li><a href=" ../VolVisualizarVoluntariosDia/interfaceVolVisualizarVoluntariosDia.php">Visualizar Voluntarios del Dia</a></li>'+
  '<li><a href=" ../VolVisualizarUsuariosDia/interfaceVolVisualizarUsuariosDia.php">Visualizar Usuarios del Dia</a></li>'+
  '<li><a href=" ../VolRegistrarActividad/interfaceVolRegistrarActividad.php">Registrar Actividad</a></li>'+
  '</ul>'+
  '</li>'+
  '</ul>'+
  '</div>';
}

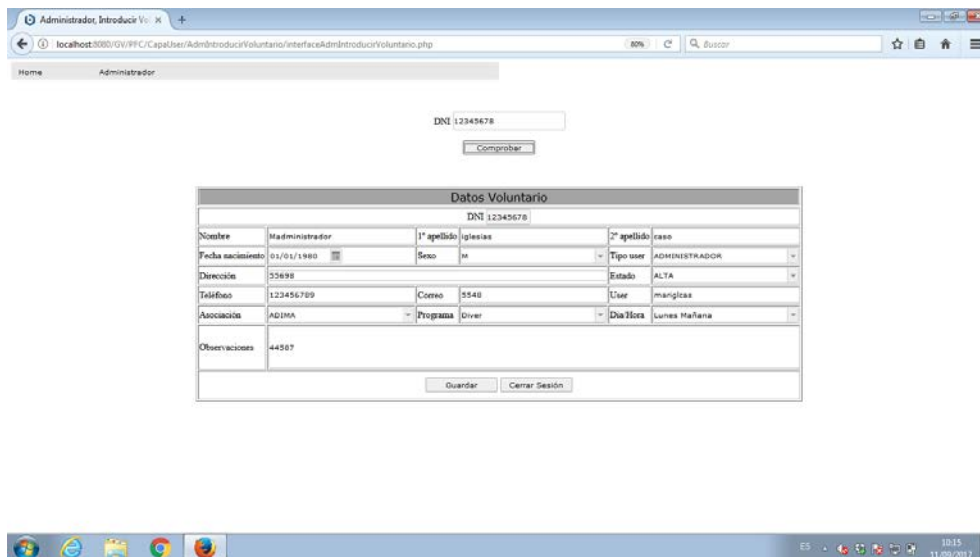
```



Figura 4-91 userCorrecto.php - Voluntario

### 4.5.3 Introducir calendario [interfaceAdmAgregarCalendario.php]

El diseño y los elementos descritos en el apartado Prototipo Administrador – Introducir Calendario para la página de introducir actividades, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\AdmAgregarCalendario y que se detalla a continuación:



Datos Voluntario					
DNI 12345678					
Nombre	Madministrador	1º apellido	iglesias	2º apellido	caso
Fecha nacimiento	01/01/1980	Sexo	m	Tipo user	ADMINISTRADOR
Dirección	55698	Estado		Estado	ALTA
Teléfono	123456789	Correo	5548	User	mangicaa
Asociación	ADIMA	Programa	Diver	Día Hora	Lunes Mañana
Observaciones	44587				

Figura 4-92 Interfaz Agregar Calendario



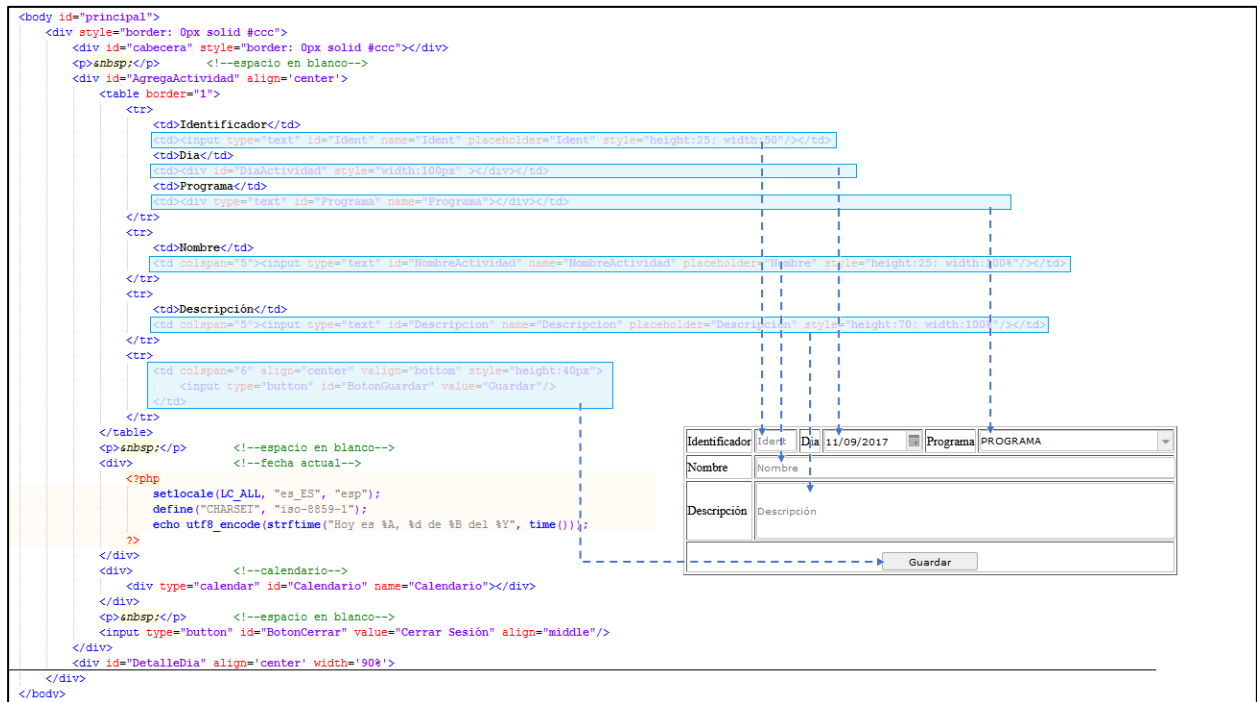


Figura 4-93 interfaceAdmAgregarCalendario.php – 1

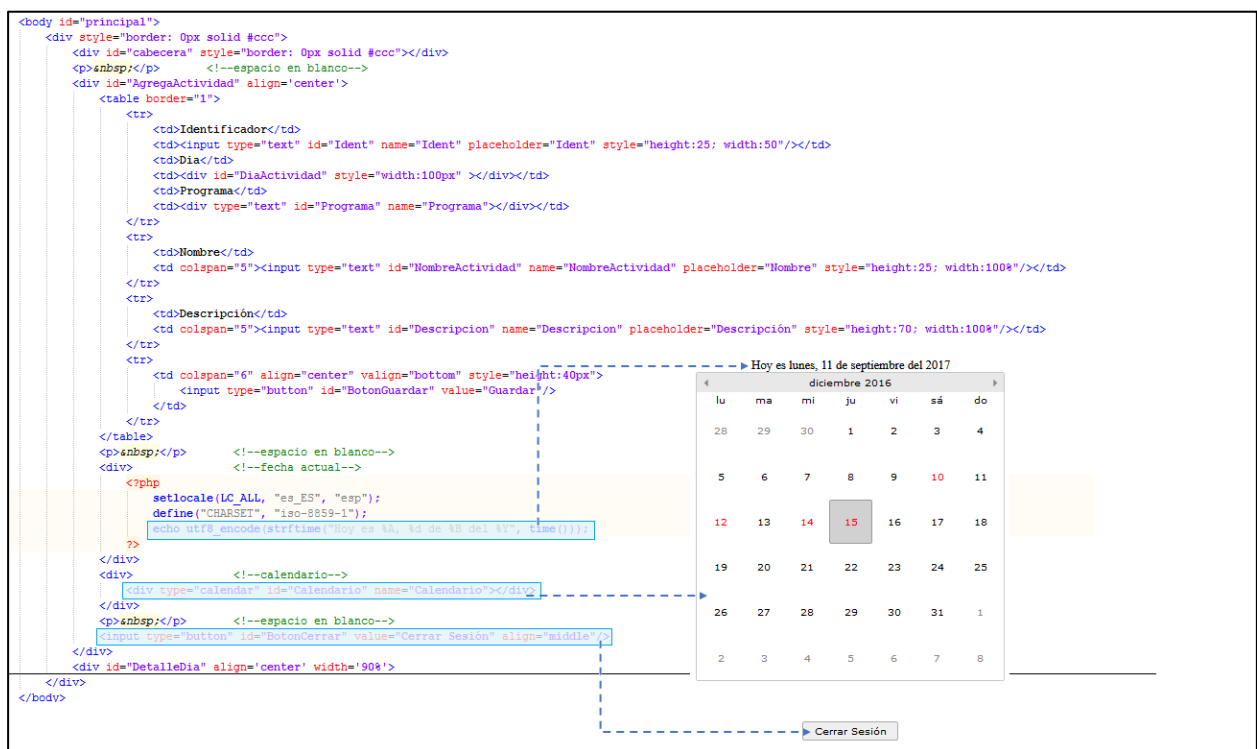


Figura 4-94 interfaceAdmAgregarCalendario.php - 2

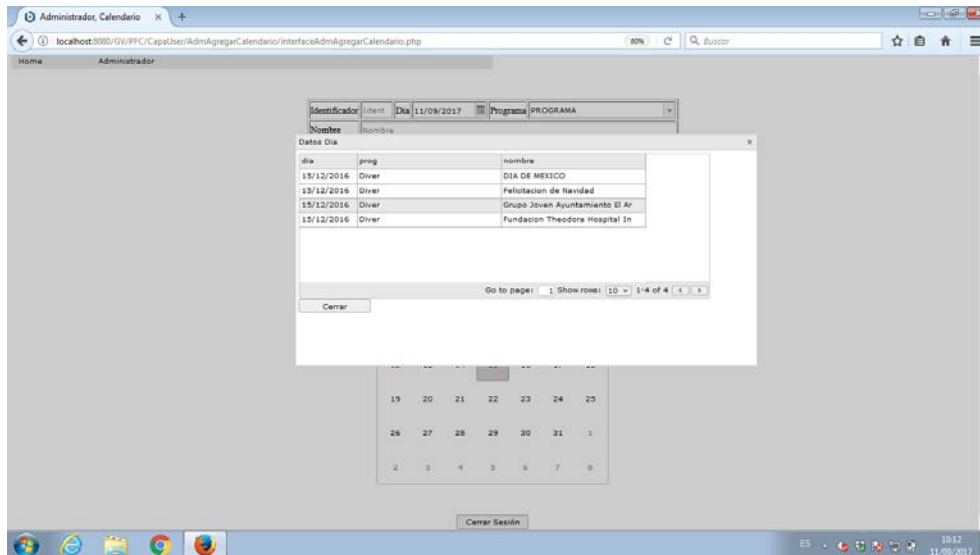


Figura 4-95 Interfaz Agregar Calendario - Detalle

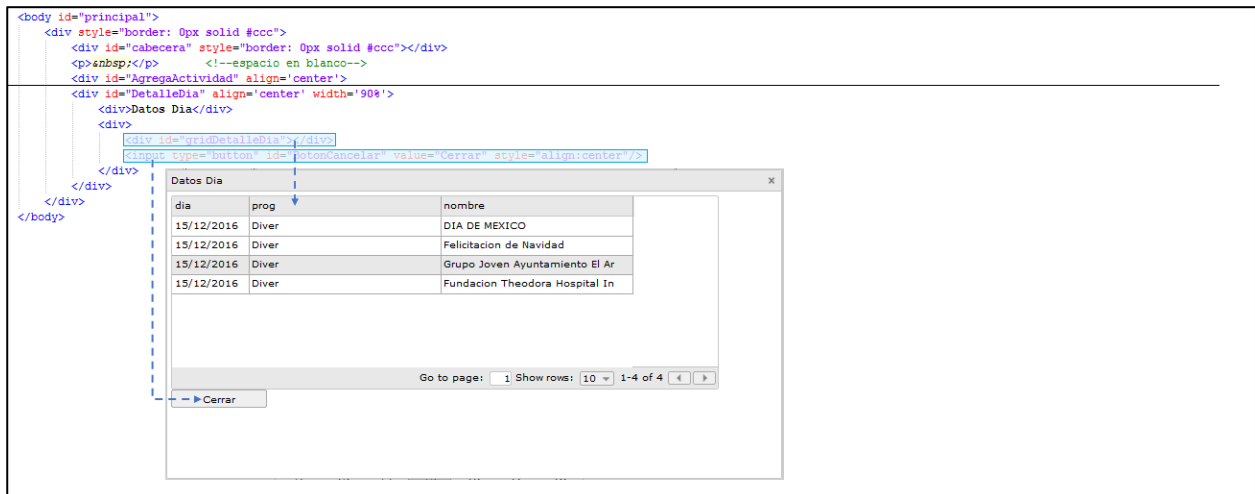


Figura 4-96 interfaceAdmAgregarCalendario.php - 3

#### 4.5.4 Introducir Voluntario [interfaceAdmIntroducirVoluntario.php]

El diseño y los elementos descritos en el apartado Prototipo Administrador – Introducir Voluntario para la página de introducir voluntarios, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\AdmIntroducirVoluntario y que se detalla a continuación:

Datos Voluntario					
DNI 12345678					
Nombre	Administrador	1º apellido	iglesias	2º apellido	cano
Fecha nacimiento	01/01/1980	Sexo	M	Tipo user	ADMINISTRADOR
Dirección	55838		Estado ALTA		
Teléfono	123456789	Correo	5548	User	manglicas
Asociación	ADIMA	Programa	Oiver	Día/Hora	Lunes Mañana
Observaciones	44587				

Figura 4-97 Interfaz Introducir Voluntario

```

<body id="principal">
  <div style="border:0px solid #ccc">
    <div id="cabecera" style="border:0px solid #ccc"></div>
    <p>&nbsp;&nbsp;&nbsp;</p> <!--espacio en blanco-->
    <div id="BuscaUser" align="center">
      <div>
        <table border="0">
          <tr>
            <td>DNI</td>
            <td><input type="text" id="BuscaDNI" name="BuscaDNI" placeholder="DNI" style="height:30; width:100%"/></td>
          </tr>
          <tr>
            <td colspan="2" align="center" valign="bottom" style="height:40px">
              <input type="button" id="BotonComprobar" value="Comprobar"/>
            </td>
          </tr>
        </table>
      </div>
    </div>
    <p>&nbsp;&nbsp;&nbsp;</p> <!--espacio en blanco-->
    <div id="DetalleUser" align="center">
      </div>
  </body>

```

Figura 4-98 interfaceAdmIntroducirVoluntario.php – 1

```

<body id="principal">
  <div style="border:0px solid #ccc">
    <div id="cabecera" style="border:0px solid #ccc"></div>
    <p>&nbsp;</p>
    <div id="BuscaUser" align="center">
      <p>&nbsp;</p>
      <div id="DetalleUser" align="center">
        <!--Datos Voluntario/div-->
        <div>
          <table border="1">
            <tr>
              <td colspan="6" align="center"><legend style="font-family:Verdana; font-size:20; background-color:#444444">Datos Voluntario</legend></td>
            </tr>
            <tr>
              <td colspan="6" align="center">DNI <input type="text" id="DNI" name="DNI" placeholder="DNI" style="height:25; width:75; background:#F3F3F3"/></td>
            </tr>
            <tr>
              <td colspan="6"><Nombre</td>
              <td colspan="6"><input type="text" id="nombre" name="nombre" placeholder="Nombre" style="height:25; width:100%/></td>
            </tr>
            <tr>
              <td colspan="3"><1º apellido</td>
              <td colspan="3"><input type="text" id="apellido1" name="apellido1" placeholder="1º Apellido" style="height:25; width:100%/></td>
            </tr>
            <tr>
              <td colspan="3"><2º apellido</td>
              <td colspan="3"><input type="text" id="apellido2" name="apellido2" placeholder="2º Apellido" style="height:25; width:100%/></td>
            </tr>
            <tr>
              <td colspan="6"><Fecha nacimiento</td>
              <td colspan="6"><div id="fecha nacim"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Sexo</td>
              <td colspan="6"><div type="text" id="sexo" name="sexo" style="width:150px"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Tipo user</td>
              <td colspan="6"><div type="text" id="tipo_user" name="tipo_user" style="width:150px"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Dirección</td>
              <td colspan="6"><input type="text" id="direccion" name="direccion" placeholder="Dirección" style="width:100%/></td>
            </tr>
            <tr>
              <td colspan="6"><Estado</td>
              <td colspan="6"><div type="text" id="estado user" name="estado user" style="width:150px"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Teléfono</td>
              <td colspan="6"><input type="text" id="telefono" name="telefono" placeholder="Teléfono" style="height:25; width:100%/></td>
            </tr>
            <tr>
              <td colspan="3"><Correo</td>
              <td colspan="3"><input type="text" id="correo" name="correo" placeholder="Correo" style="height:25; width:100%/></td>
            </tr>
            <tr>
              <td colspan="3"><User</td>
              <td colspan="3"><input type="text" id="user" name="user" placeholder="User" style="height:25; width:100%/></td>
            </tr>
            <tr>
              <td colspan="6"><Asociación</td>
              <td colspan="6"><div type="text" id="asociacion" name="Asociacion" style="width:150px"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Programa</td>
              <td colspan="6"><div type="text" id="programa" name="Programa" style="width:150px"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Dia/Hora</td>
              <td colspan="6"><div type="text" id="dia hora" name="dia hora" style="width:150px"></div></td>
            </tr>
            <tr>
              <td colspan="6"><Observaciones</td>
              <td colspan="6"><input type="text" id="observaciones" name="observaciones" placeholder="Observaciones" style="height:70; width:100%/></td>
            </tr>
            <tr>
              <td colspan="6" align="center" height="50">
                <input type="button" id="BotonGuardar" value="Guardar"/>
                <input type="button" id="BotonCerrar" value="Cerrar Sesión"/>
              </td>
            </tr>
          </table>
        </div>
      </div>
    </div>
  </div>
</body>

```

Datos Voluntario					
DNI 2145678					
Nombre	Administrador	1º apellido	glesias	2º apellido	caso
Fecha nacimiento	01/01/1980	Sexo	M	Tipo user	ADMINISTRADOR
Dirección	55698			Estado	ALTA
Teléfono	123456789	Correo	4548	User	mariglas
Asociación	ADIMA	Programa	Diver	Dia/Hora	Lunes Mañana
Observaciones	44587				

Figura 4-99 interfaceAdmIntroducirVoluntario.php – 2

### 4.5.5 Modificar Voluntario [interfaceAdmModificarVoluntario.php]

El diseño y los elementos descritos en el apartado Prototipo Administrador – Modificar Voluntario para la página de modificar voluntarios, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\AdmModificarVoluntario y que se detalla a continuación:

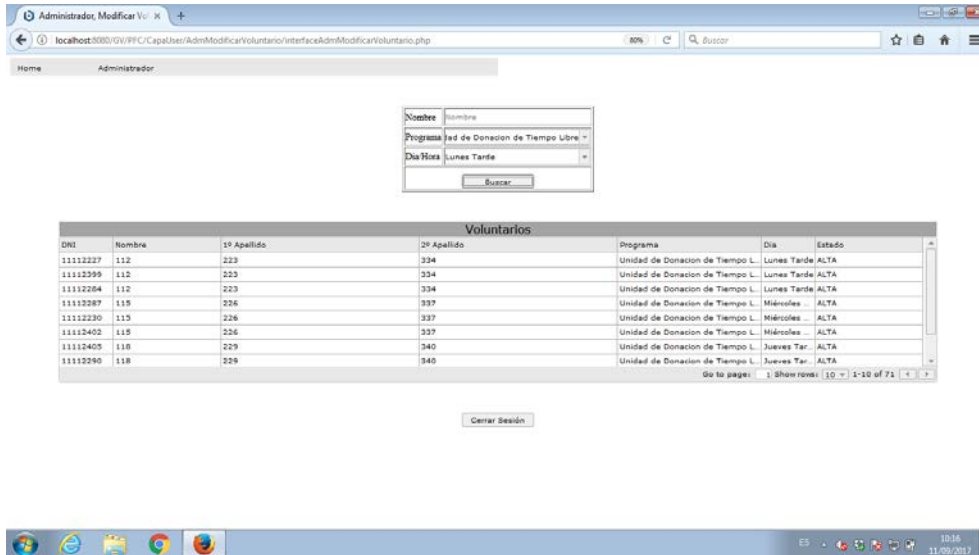


Figura 4-100 Interfaz Modificar Voluntario

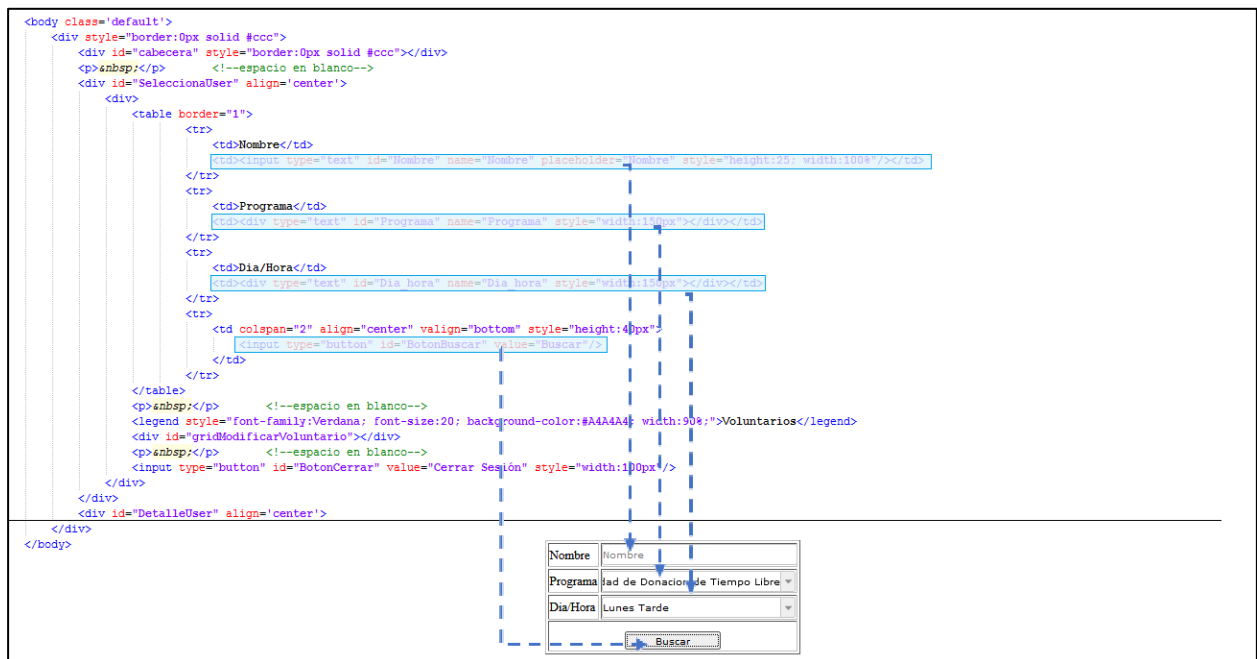


Figura 4-101 interfaceAdmModificarVoluntario.php - 1

```

<body class="default">
  <div style="border:0px solid #ccc">
    <div id="cabecera" style="border:0px solid #ccc"></div>
    <p>&nbsp;&nbsp;&nbsp;</p> <!--espacio en blanco-->
    <div id="SeleccionaUser" align="center">
      <div>
        <table border="1">
          <tr>
            <td>Nombre</td>
            <td><input type="text" id="Nombre" name="Nombre" placeholder="Nombre" style="height:25; width:100%"/></td>
          </tr>
          <tr>
            <td>Programa</td>
            <td><div type="text" id="Programa" name="Programa" style="width:150px"/></td>
          </tr>
          <tr>
            <td>Dia/Hora</td>
            <td><div type="text" id="Dia_hora" name="Dia_hora" style="width:150px"/></td>
          </tr>
          <tr>
            <td colspan="2" align="center" valign="bottom" style="height:40px">
              <input type="button" id="BotonBuscar" value="Buscar"/>
            </td>
          </tr>
        </table>
        <p>&nbsp;&nbsp;&</p> <!--espacio en blanco-->
        <legend style="font-family:Verdana; font-size:20; background-color:#A4A4A4; width:90%;>Voluntarios</legend>
        <div id="gridModificarVoluntario"></div>
        <p>&nbsp;&nbsp;&</p> <!--espacio en blanco-->
        <input type="button" id="BotonCerrar" value="Cerrar Sesión" style="width:100px"/>
      </div>
    </div>
    <div id="DetalleUser" align="center">
  </div>
</body>

```

DNI	Nombre	1º Apellido	2º Apellido	Programa	Dia	Estado
11112227	112	223	334	Unidad de Donacion de Tiempo Libre	Lunes Tarde	ALTA
11112399	112	223	334	Unidad de Donacion de Tiempo Libre	Lunes Tarde	ALTA
11112284	112	223	334	Unidad de Donacion de Tiempo Libre	Lunes Tarde	ALTA
11112287	115	226	337	Unidad de Donacion de Tiempo Libre	Miércoles ...	ALTA
11112230	115	226	337	Unidad de Donacion de Tiempo Libre	Miércoles ...	ALTA
11112402	115	226	337	Unidad de Donacion de Tiempo Libre	Miércoles ...	ALTA
11112405	118	229	340	Unidad de Donacion de Tiempo Libre	Jueves Tar...	ALTA
11112290	118	229	340	Unidad de Donacion de Tiempo Libre	Jueves Tar...	ALTA

Figura 4-102 interfaceAdmModificarVoluntario.php - 2

Figura 4-103 Interfaz Modificar Voluntario - Detalle

```

<body class='default'>
<div style="border:0px solid #ccc">
<div id="cabecera" style="border:0px solid #ccc"></div>
<p>nbsp;</p>
<div id="SeleccionaUser" align="center">
<div id="DetalleUser" align="center">
<div>Datos Voluntario</div>
<div>
<p>nbsp;</p>
<table border="1">
<tr>
<td colspan="6" align="center">DNI<input type="text" id="DetalleDNI" name="DetalleDNI" placeholder="DNI" style="height:25; width:75; background:#f1f3"/></td>
</tr>
<tr>
<td>Nombre</td>
<td colspan="5"><input type="text" id="DetalleNombre" name="DetalleNombre" placeholder="Nombre" style="height:25; width:100%"/></td>
</tr>
<tr>
<td colspan="2" style="width:500px">1º apellido</td>
<td colspan="4"><input type="text" id="DetalleApellido1" name="DetalleApellido1" placeholder="1º Apellido" style="height:25; width:100%"/></td>
</tr>
<tr>
<td colspan="2">2º apellido</td>
<td colspan="4"><input type="text" id="DetalleApellido2" name="DetalleApellido2" placeholder="2º Apellido" style="height:25; width:100%"/></td>
</tr>
<tr>
<td>Fecha nacimiento</td>
<td colspan="5"><input type="text" id="DetalleFecha nacia" style="width:100%"/></td>
</tr>
<tr>
<td>Sexo</td>
<td colspan="5"><div type="text" id="DetalleSexo" name="DetalleSexo" style="width:150px"/></div></td>
</tr>
<tr>
<td>Tipo user</td>
<td colspan="5"><div type="text" id="DetalleTipo user" name="DetalleTipo user" style="width:150px"/></div></td>
</tr>
<tr>
<td>Dirección</td>
<td colspan="5"><input type="text" id="DetalleDireccion" name="DetalleDireccion" placeholder="Dirección" style="width:100%"/></td>
</tr>
<tr>
<td>Estado</td>
<td colspan="5"><div type="text" id="DetalleEstado user" name="DetalleEstado user" style="width:150px"/></div></td>
</tr>
<tr>
<td>Teléfono</td>
<td colspan="5"><input type="text" id="DetalleTelefono" name="DetalleTelefono" placeholder="Teléfono" style="height:25; width:100%"/></td>
</tr>
<tr>
<td>Correo</td>
<td colspan="5"><input type="text" id="DetalleCorreo" name="DetalleCorreo" placeholder="Correo" style="height:25; width:100%"/></td>
</tr>
<tr>
<td>User</td>
<td colspan="5"><input type="text" id="DetalleUser" name="DetalleUser" placeholder="User" style="height:25; width:100%"/></td>
</tr>
<tr>
<td>Asociación</td>
<td colspan="5"><div type="text" id="DetalleAsociacion" name="DetalleAsociacion" style="width:150px"/></div></td>
</tr>
<tr>
<td>Programa</td>
<td colspan="5"><div type="text" id="DetallePrograma" name="DetallePrograma" style="width:150px"/></div></td>
</tr>
<tr>
<td>Dia/Hora</td>
<td colspan="5"><div type="text" id="DetalleDia hora" name="DetalleDia hora" style="width:150px"/></div></td>
</tr>
<tr>
<td>Observaciones</td>
<td colspan="5"><input type="text" id="DetalleObservaciones" name="DetalleObservaciones" placeholder="Observaciones" style="height:70; width:100%"/></td>
</tr>
<tr>
<td colspan="6" align="center" height="50">
<input type="button" id="BotonGuardar" value="Guardar"/>
<input type="button" id="BotonCancelar" value="Cancelar"/>
</td>
</tr>
</table>
<!--espacio en blanco-->
</div>
</div>
</body>

```

Figura 4-104 interfaceAdmModificarVoluntario.php - 3

### 4.5.6 Visualizar Usuarios [interfaceAdmVisualizarUsuarios.php]

El diseño y los elementos descritos en el apartado Prototipo Administrador – Visualizar Usuario para la página de visualizar usuarios, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\AdmVisualizarUsuarios y que se detalla a continuación:

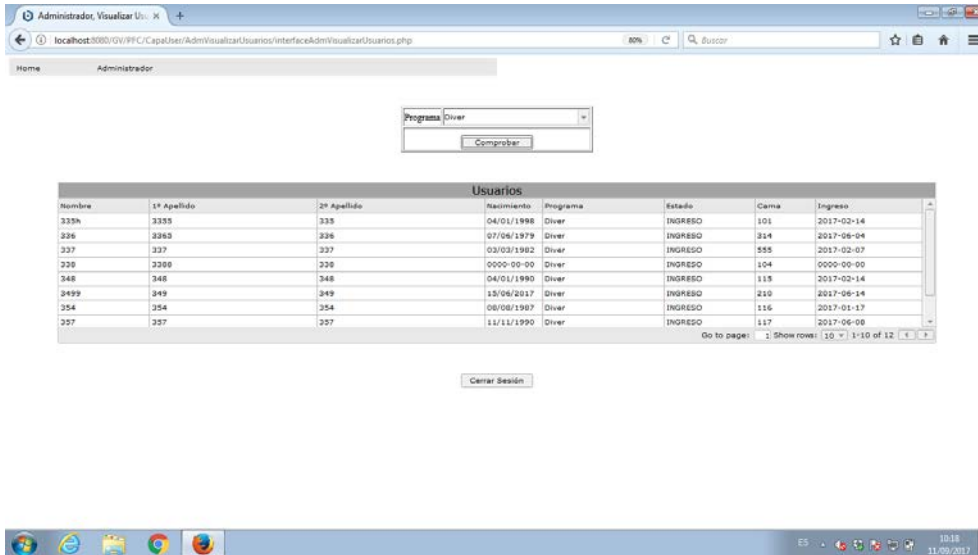


Figura 4-105 Interfaz Visualizar Usuarios

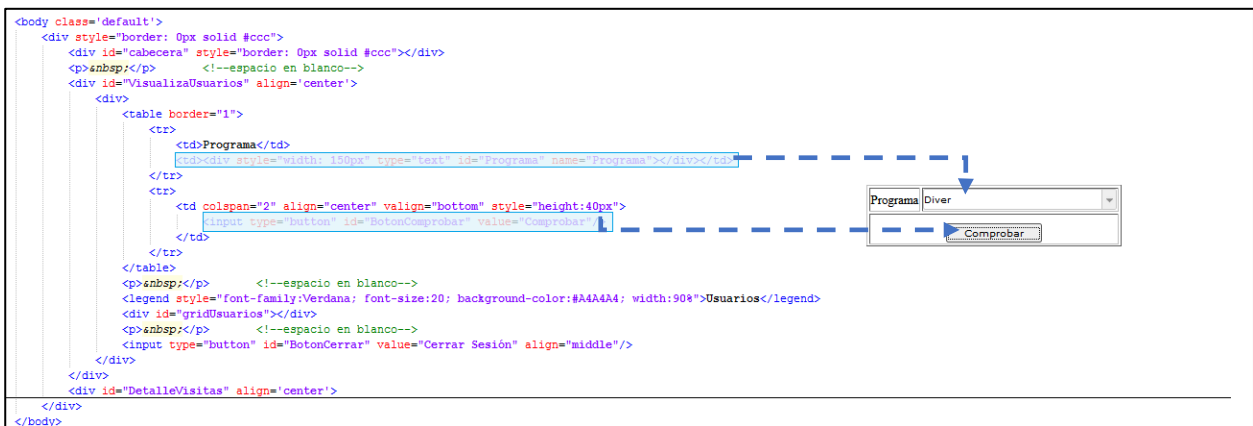


Figura 4-106 interfaceAdmVisualizarUsuarios.php - 1



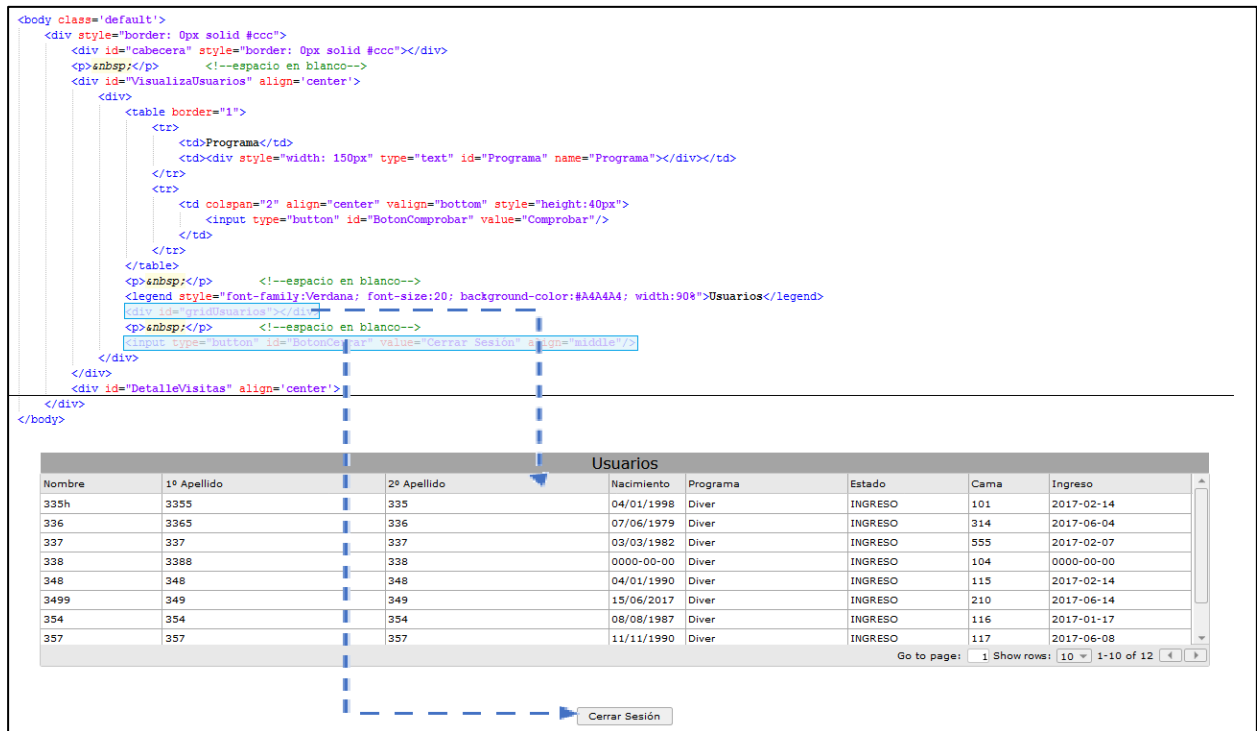


Figura 4-107 interfaceAdmVisualizarUsuarios.php - 2

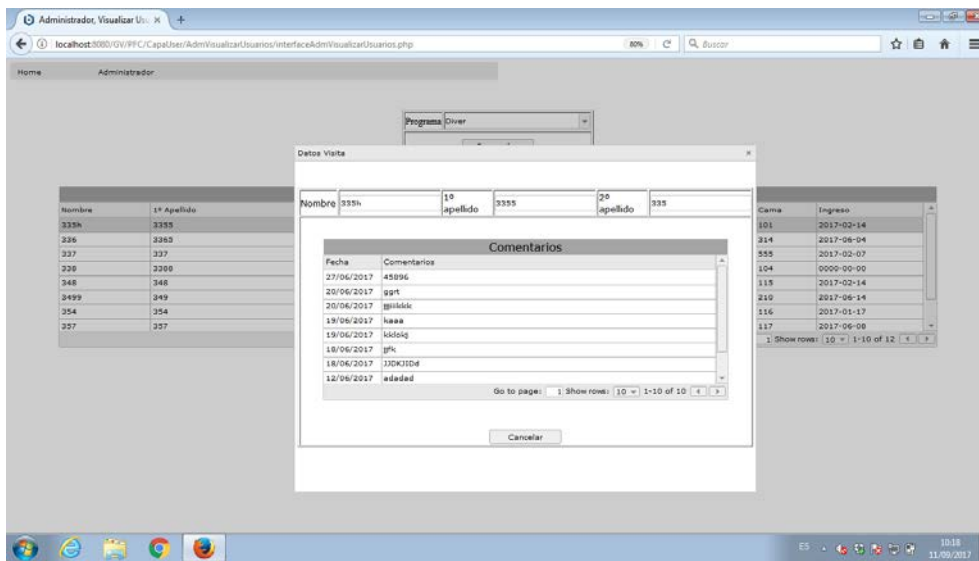


Figura 4-108 Interfaz Visualizar Usuarios - Detalle

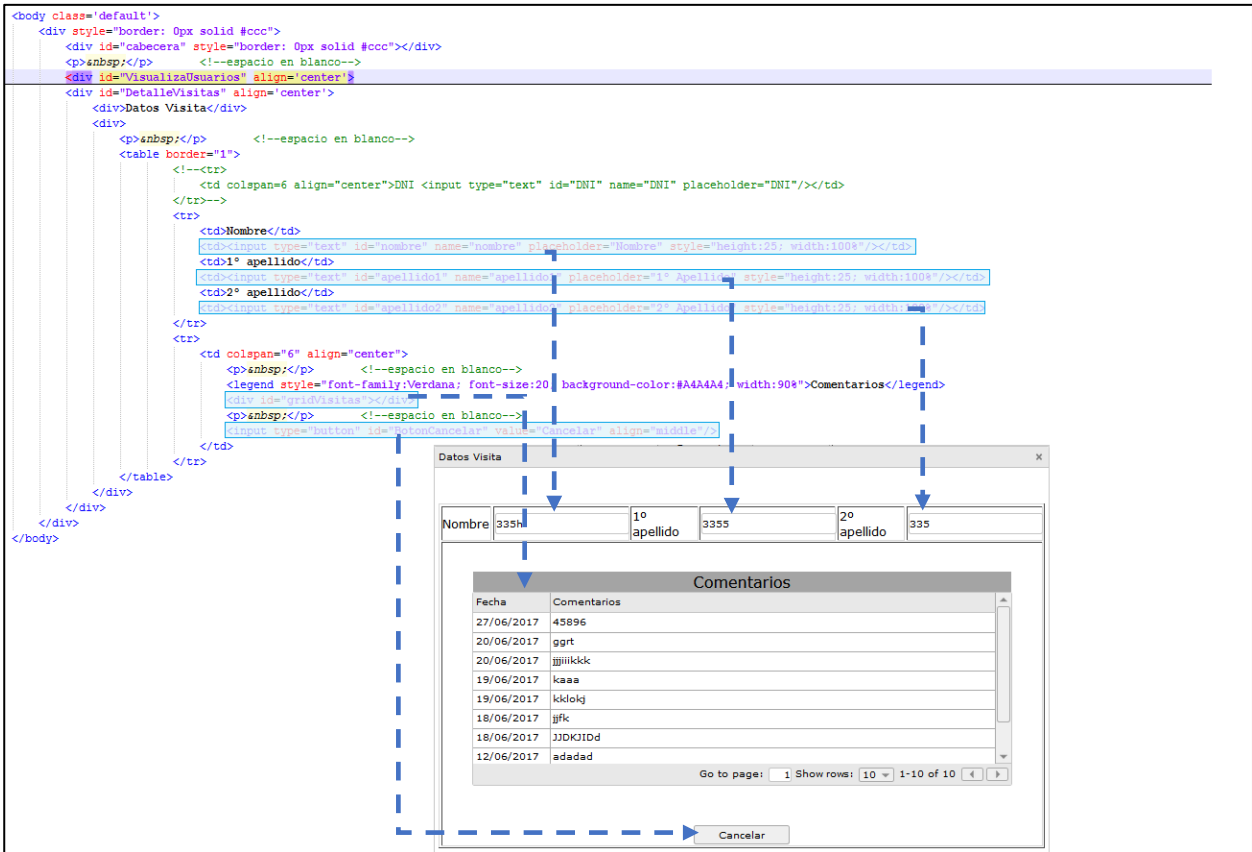


Figura 4-109 interfaceAdmVisualizarUsuarios.php - 3

#### 4.5.7 Introducir Usuarios [interfacePerSolicitarAsistencia.php]

El diseño y los elementos descritos en el apartado Prototipo Personal – Introducir Usuario para la página de introducir usuarios, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\PerSolicitarAsistencia y que se detalla a continuación:

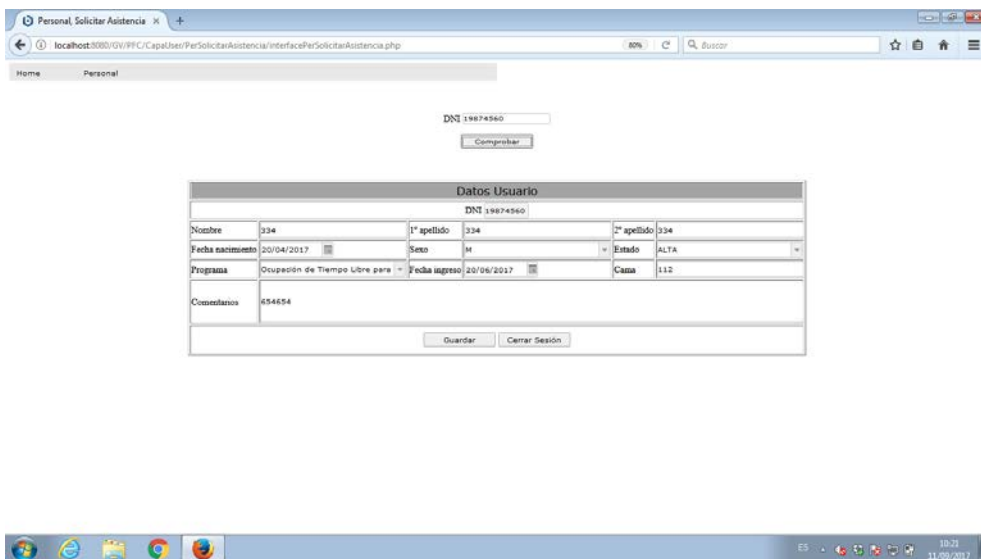


Figura 4-110 Interfaz Introducir Usuarios

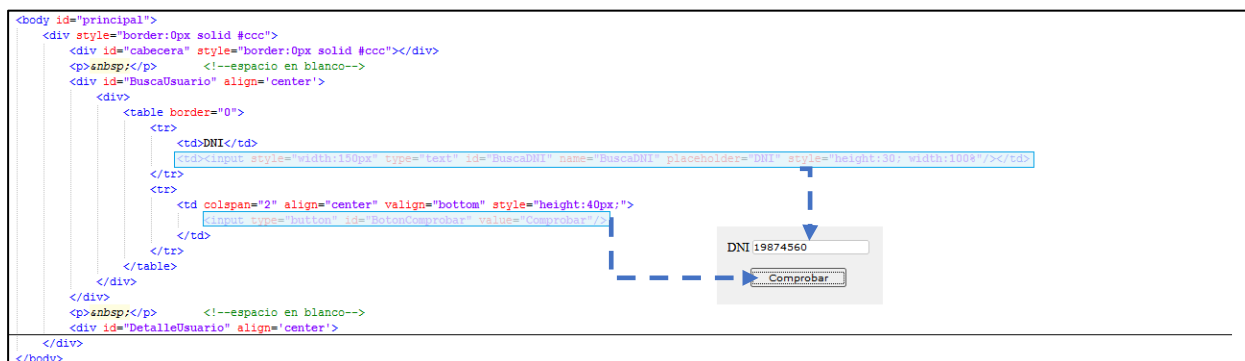


Figura 4-111 interfacePerSolicitarAsistencia.php – 1

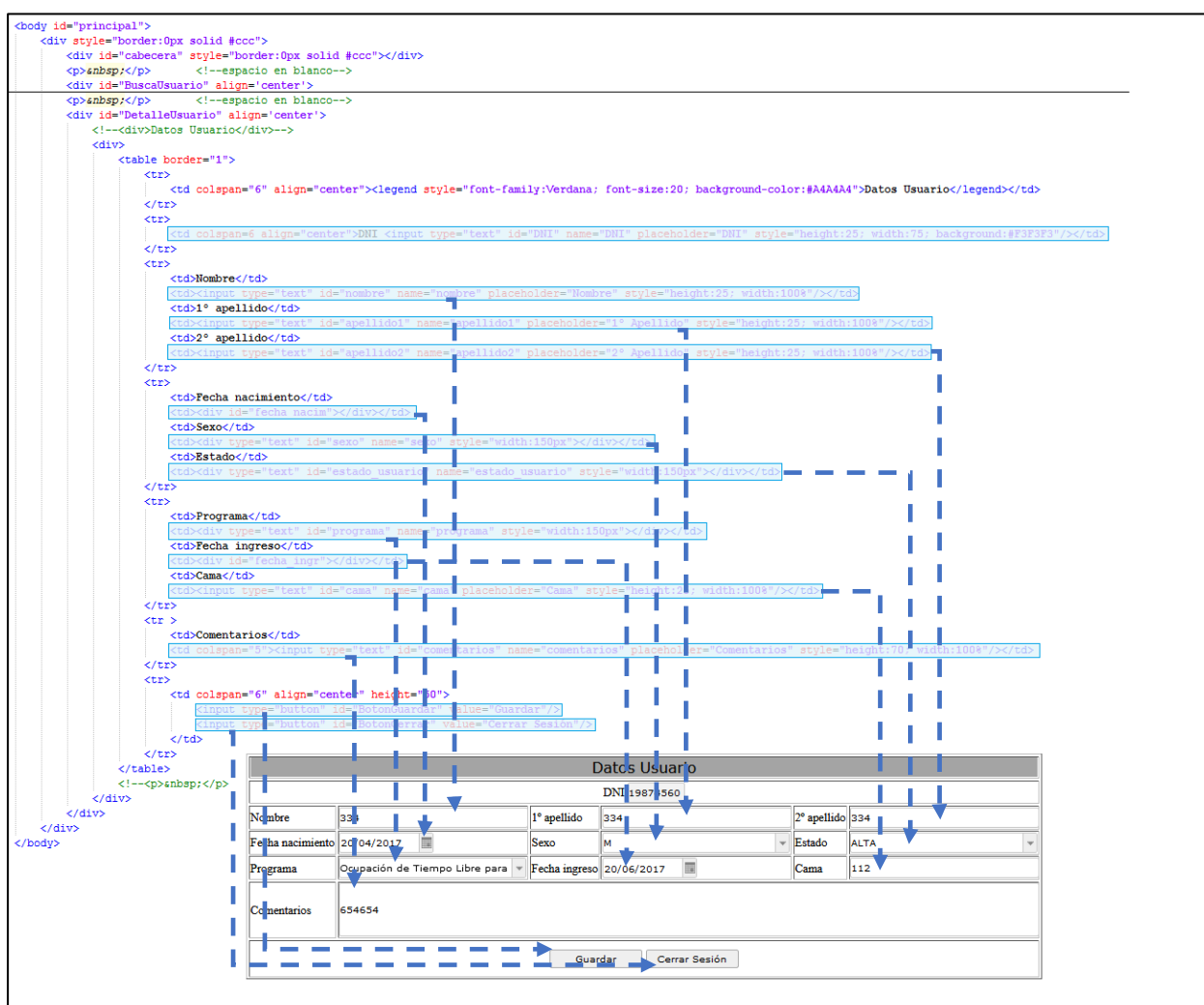


Figura 4-112 interfacePerSolicitarAsistencia.php - 2

### 4.5.8 Modificar Usuarios [interfacePerModificarAsistencia.php]

El diseño y los elementos descritos en el apartado Prototipo Personal – Modificar Usuario para la página de modificar usuarios, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\PerModificarAsistencia y que se detalla a continuación:

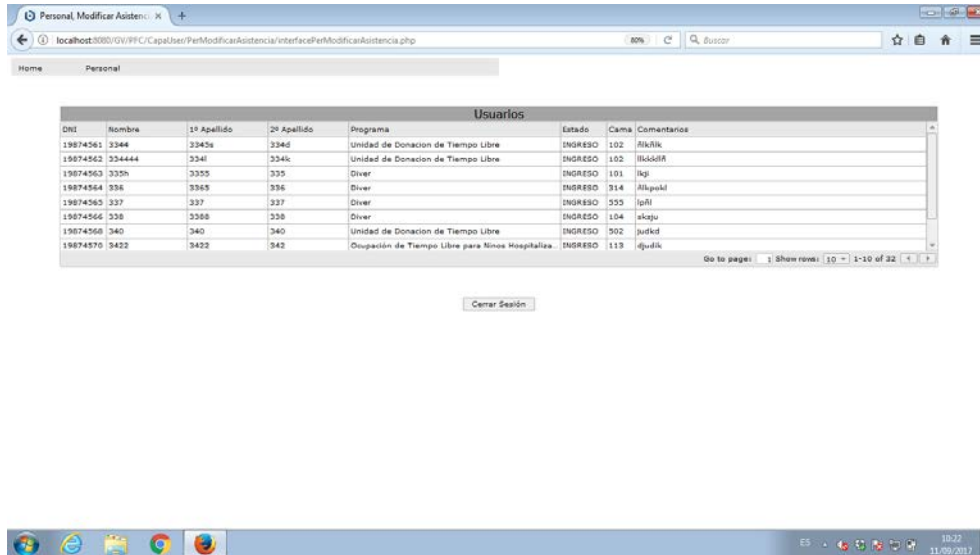


Figura 4-113 Interfaz Modificar Usuarios

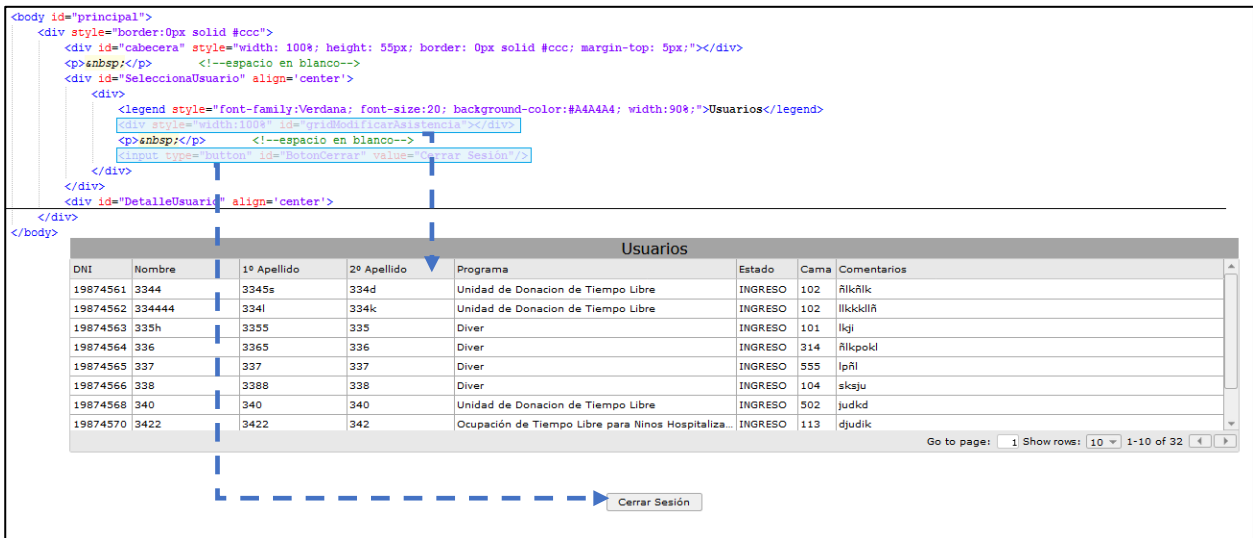


Figura 4-114 interfacePerModificarAsistencia.php - 1

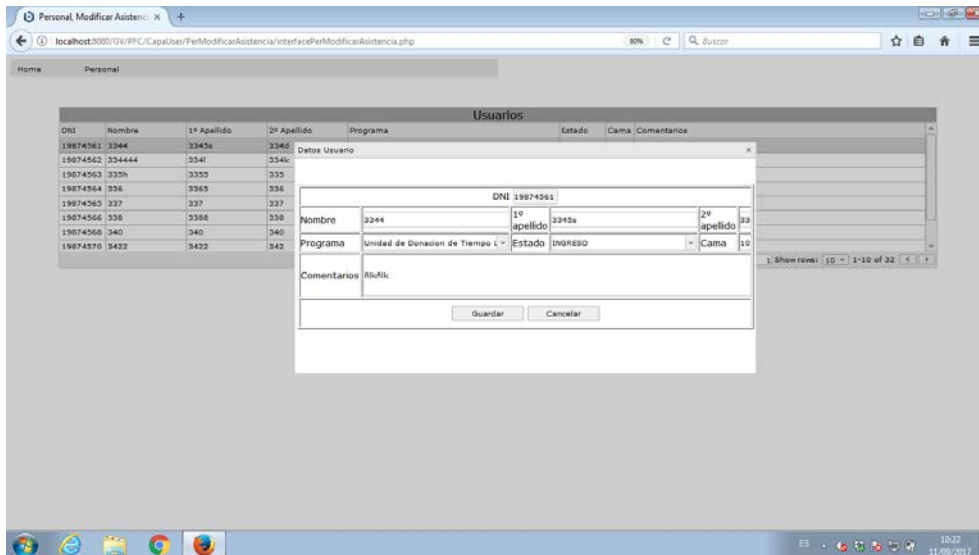


Figura 4-115 Interfaz Modificar Usuarios – Detalle

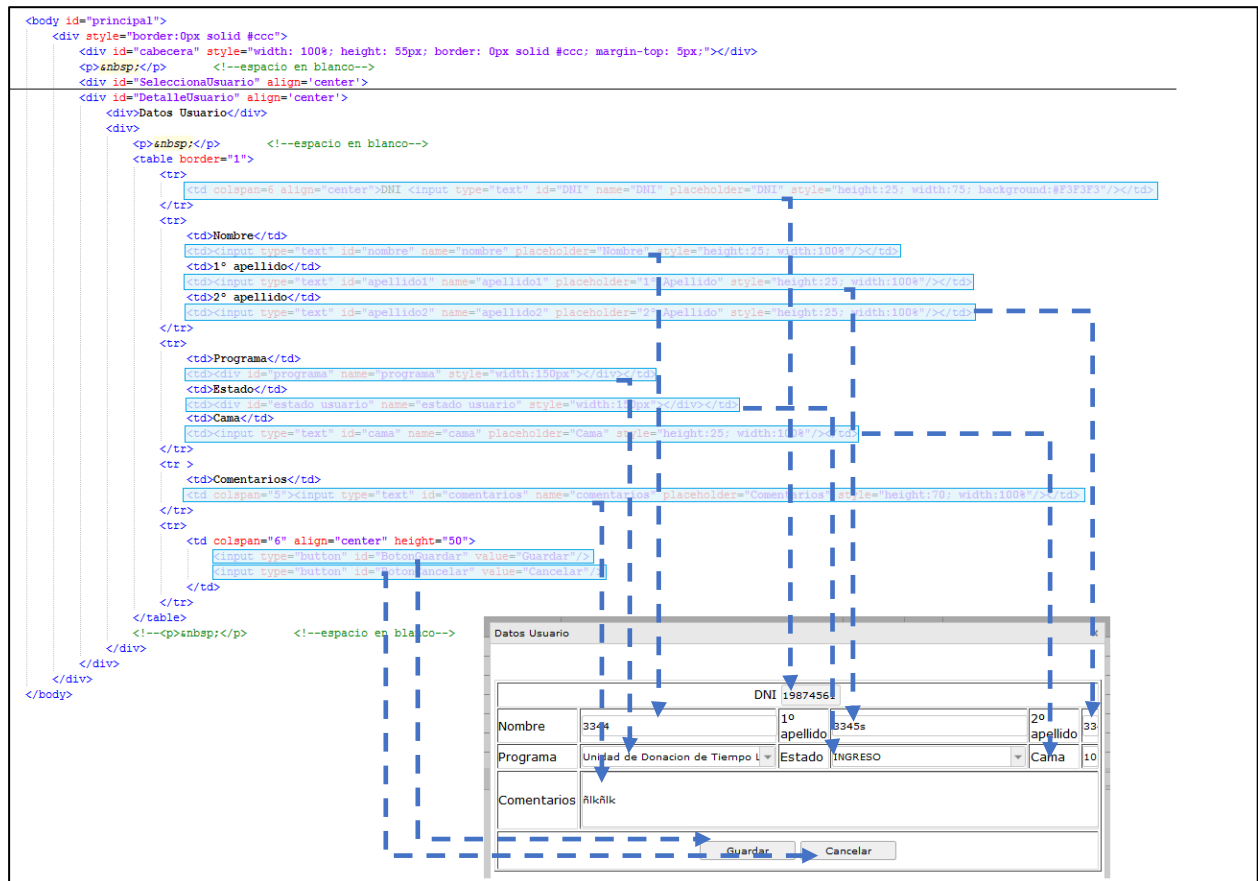


Figura 4-116 interfacePerModificarAsistencia.php - 2

### 4.5.9 Visualizar Calendario [interfaceVolVisualizarCalendario.php]

El diseño y los elementos descritos en el apartado Prototipo Voluntario – Visualizar Calendario para la página de visualizar calendario, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\VolVisualizarCalendario y que se detalla a continuación:

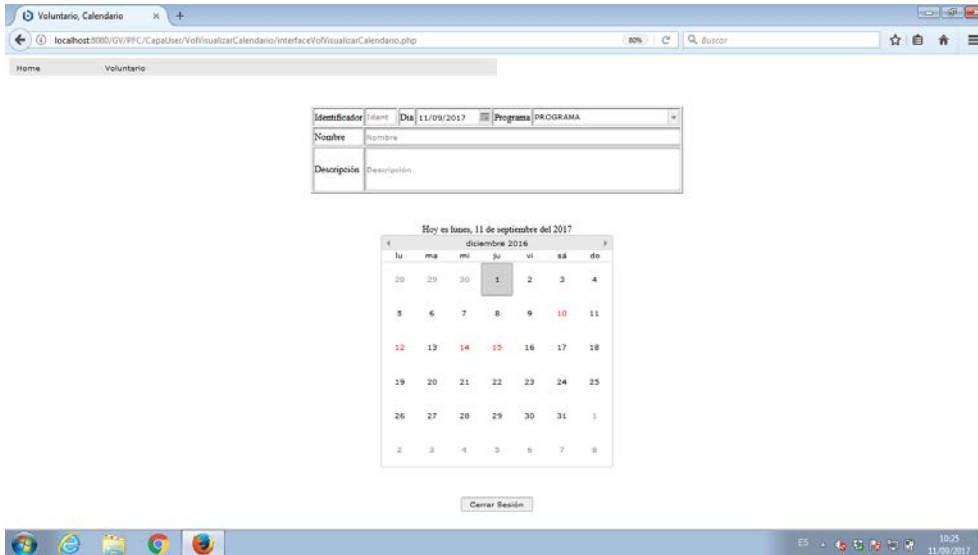


Figura 4-117 Interfaz Visualizar Calendario

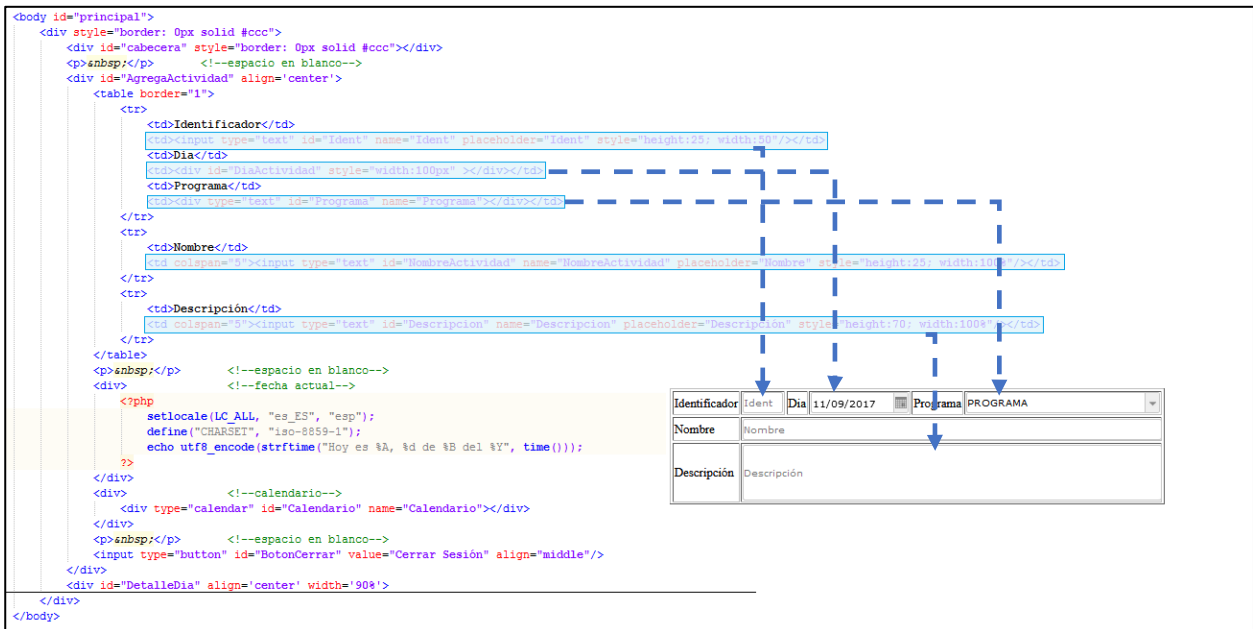


Figura 4-118 interfaceVolVisualizarCalendario.php - 1

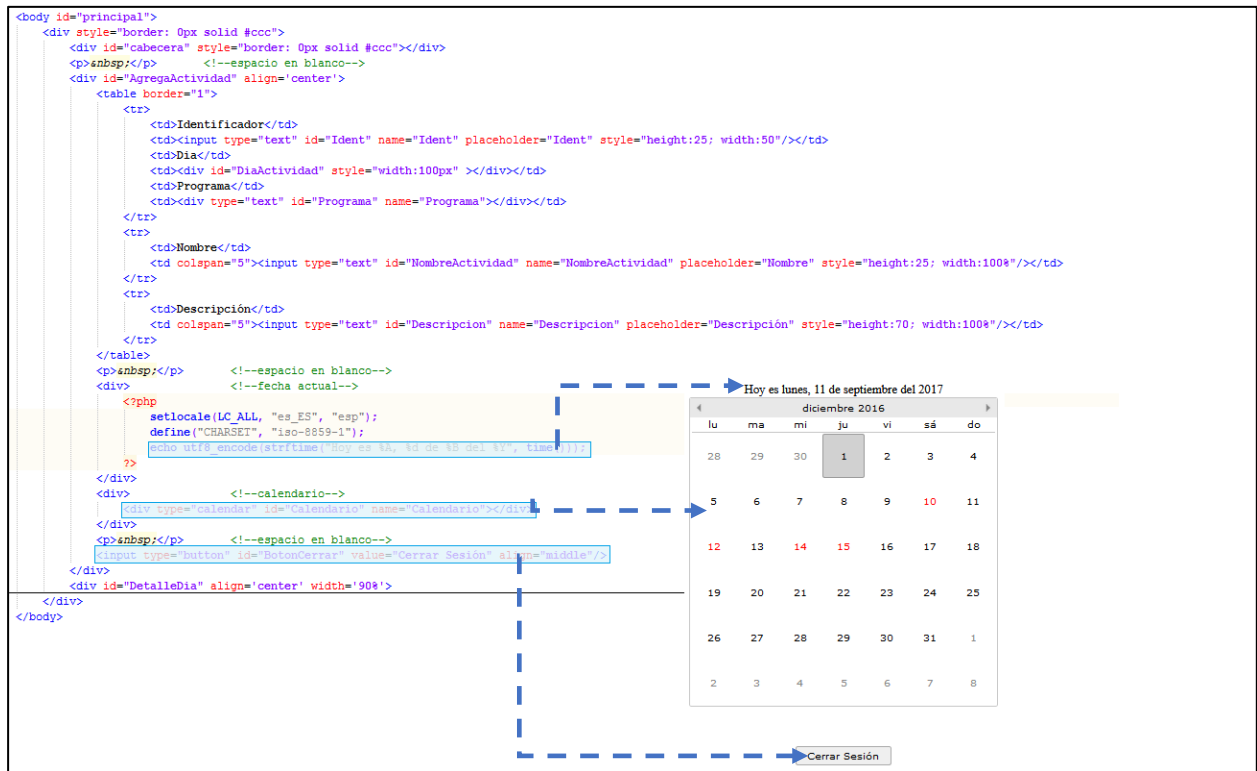


Figura 4-119 interface VolVisualizarCalendario.php - 2

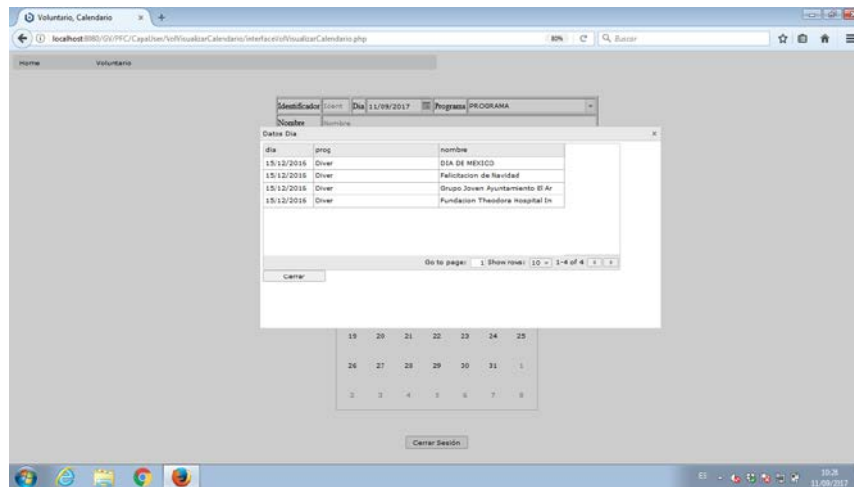


Figura 4-120 Interfaz Visualizar Calendario – Detalle

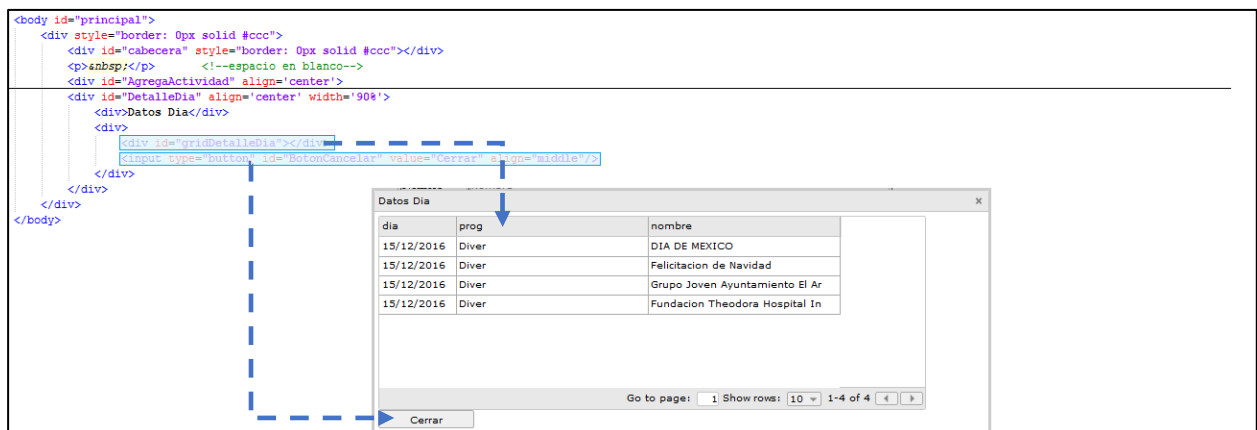


Figura 4-121 interface VolVisualizarCalendario.php - 3

#### 4.5.10 Visualizar Voluntarios del día [interfaceVolVisualizarVoluntariosDia.php]

El diseño y los elementos descritos en el apartado Prototipo Visualizar Voluntario del día para la página de visualizar voluntarios del día, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\VolVisualizarVoluntariosDia y que se detalla a continuación:

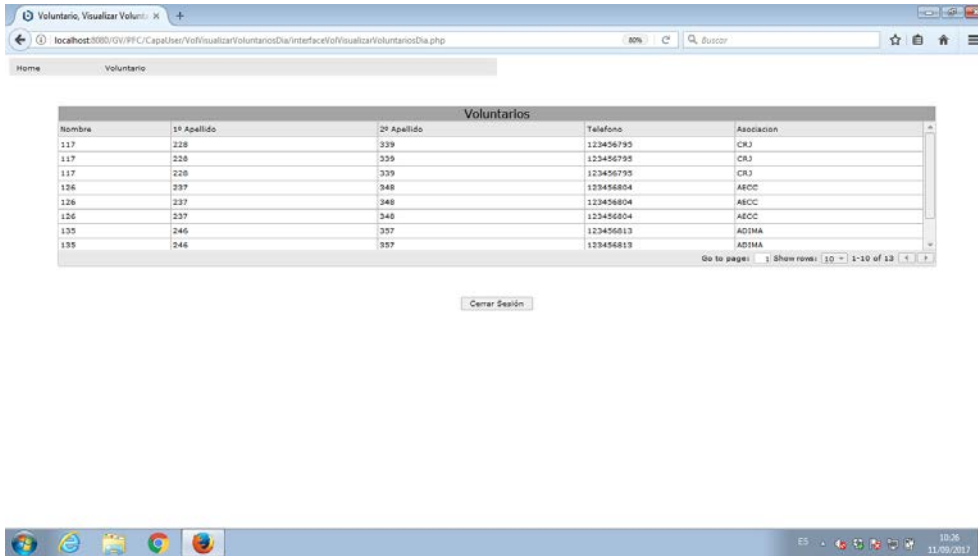


Figura 4-122 Interfaz Visualizar Voluntarios del día

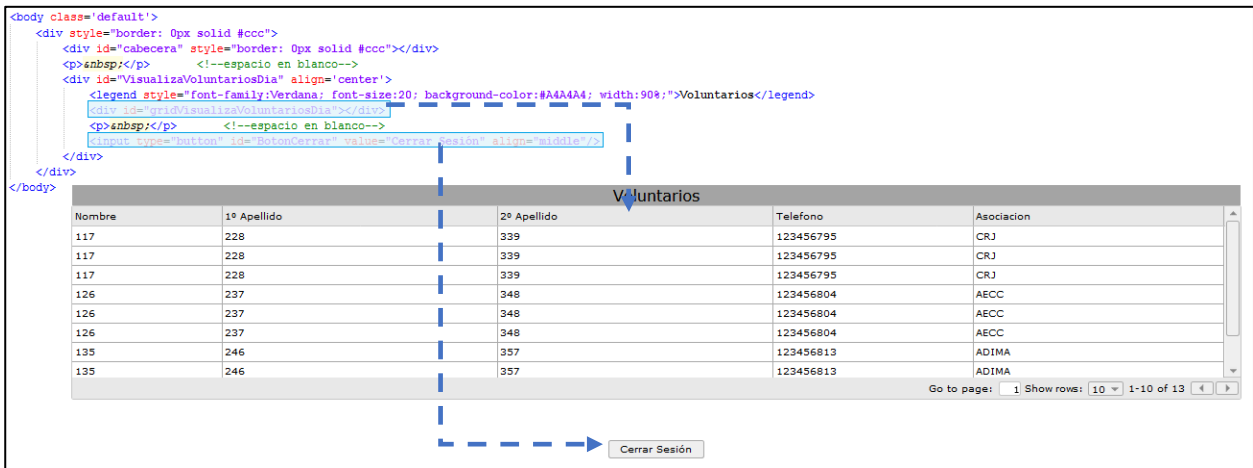


Figura 4-123 interfaceVolVisualizarVoluntariosDia.php



#### 4.5.11 Visualizar Usuarios del día [intefaceVolVisualizarUsuariosDia.php]

El diseño y los elementos descritos en el apartado Prototipo Visualizar Usuario del día para la página de visualizar usuarios del día, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\VolVisualizarUsuariosDia y que se detalla a continuación:

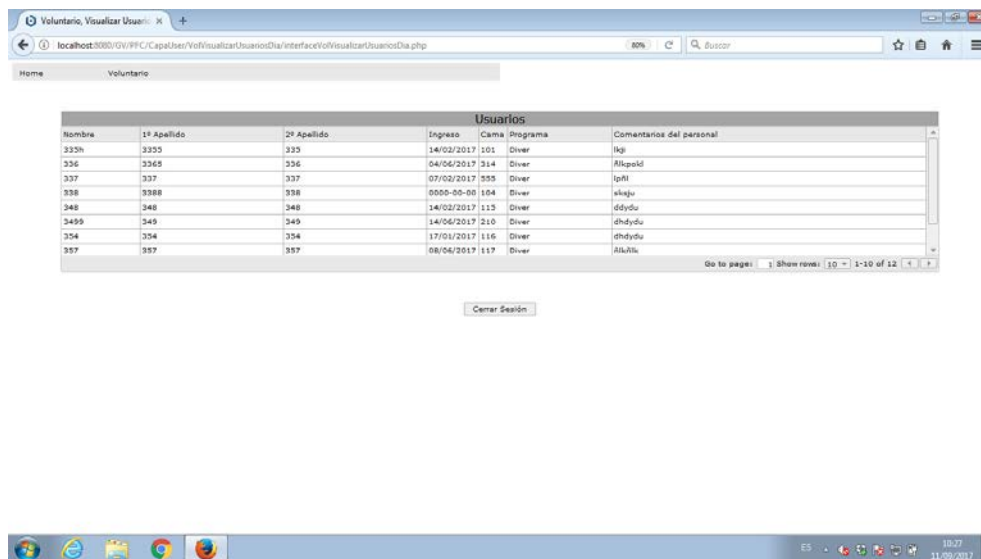


Figura 4-124 Interfaz Visualizar Usuarios del día

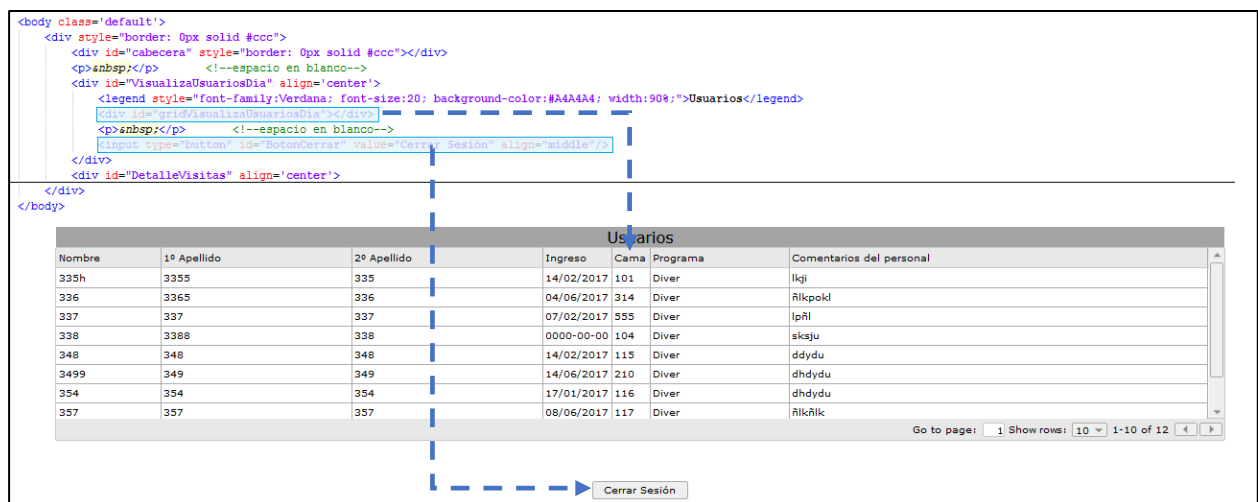


Figura 4-125 intefaceVolVisualizarUsuariosDia.php - 1

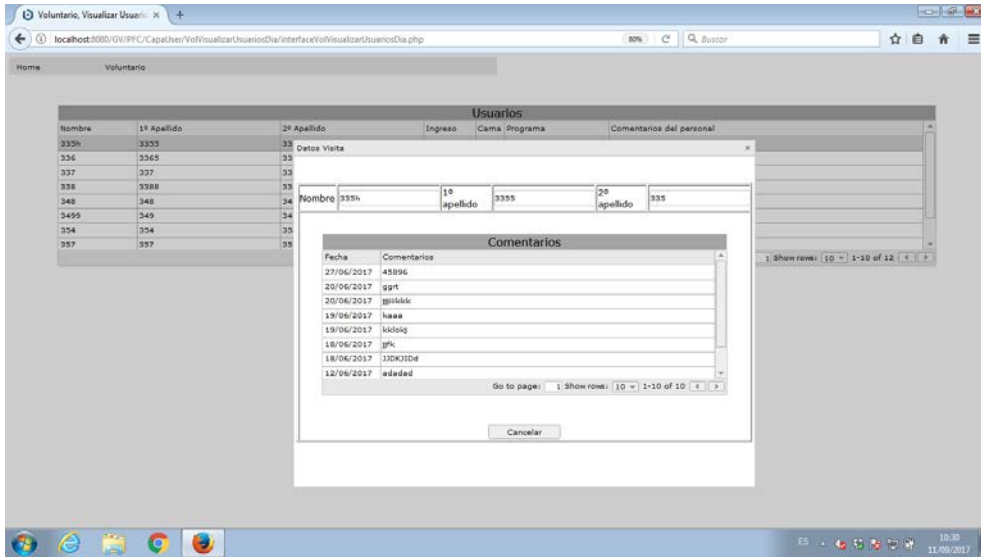


Figura 4-126 Interfaz Visualizar Usuarios del día – Detalle

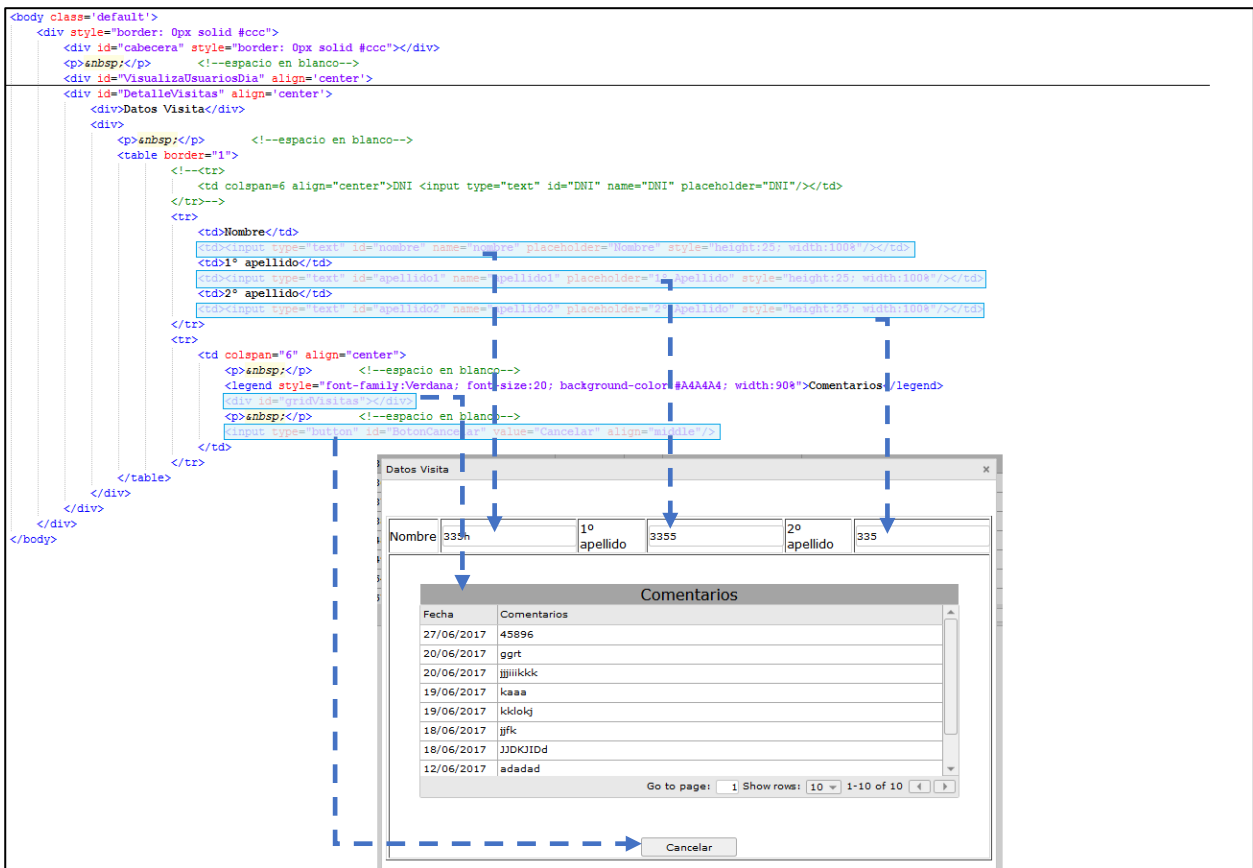


Figura 4-127 interfazVolVisualizarUsuariosDia.php - 2

#### 4.5.12 Registro Actividad [interfaceVolRegistrarActividad.php]

El diseño y los elementos descritos en el apartado Prototipo Voluntario – Registrar actividad para la página de registrar actividad, se ha conseguido mediante el pseudocódigo que está en la dirección \GV\PFC\CapaUser\VolRegistrarActividad y que se detalla a continuación:

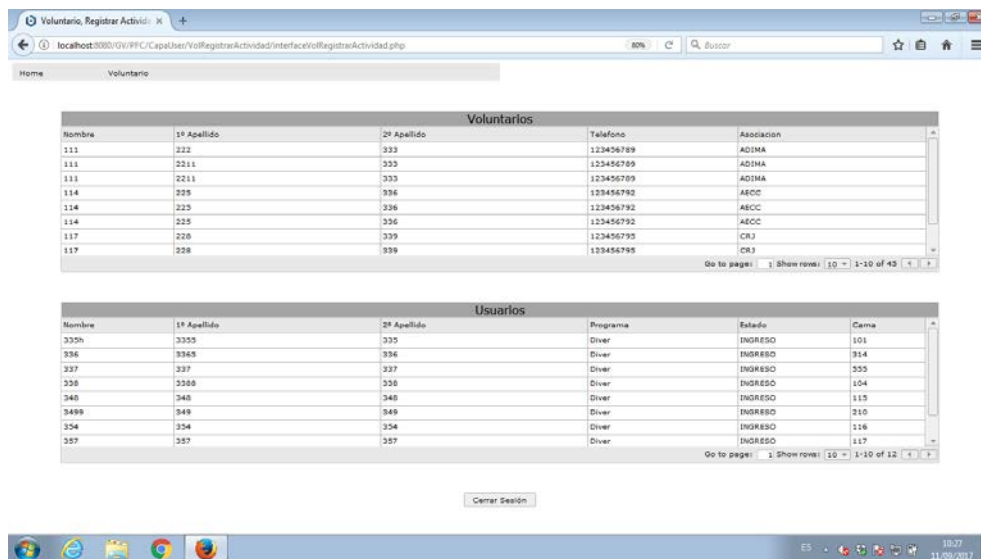


Figura 4-128 Interfaz Registro Actividad

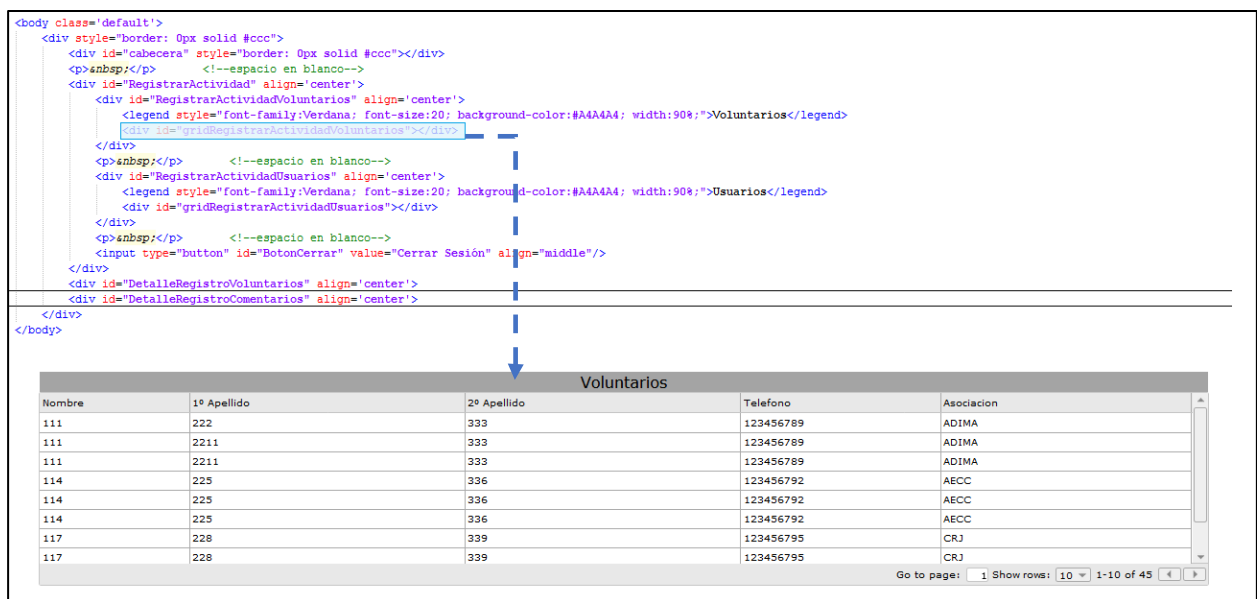


Figura 4-129 interfaceVolRegistrarActividad.php - 1

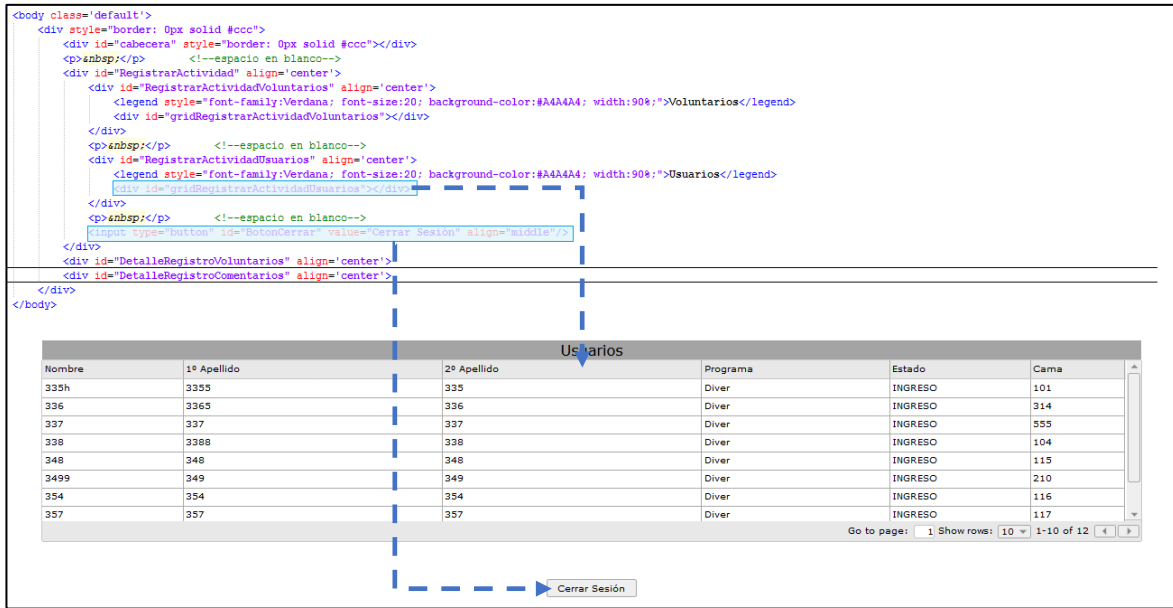


Figura 4-130 interfaceVolRegistarActividad.php – 2

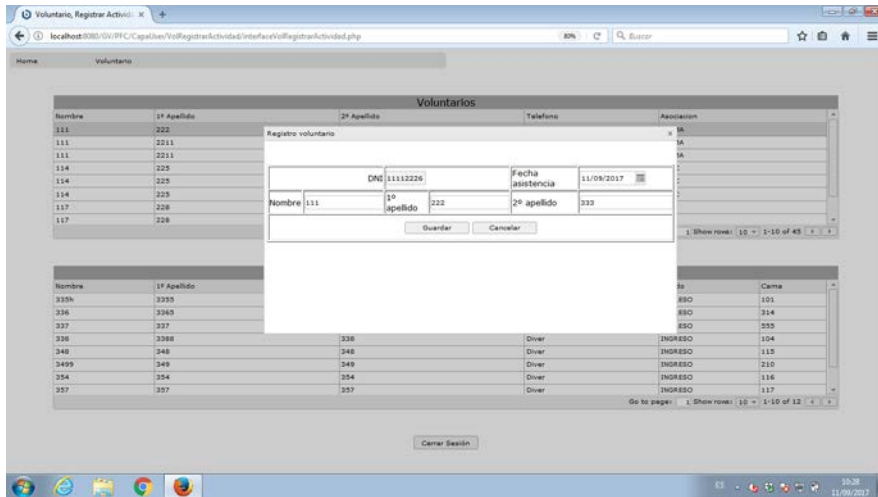


Figura 4-131 Interfaz Registro Actividad – Detalle Voluntarios

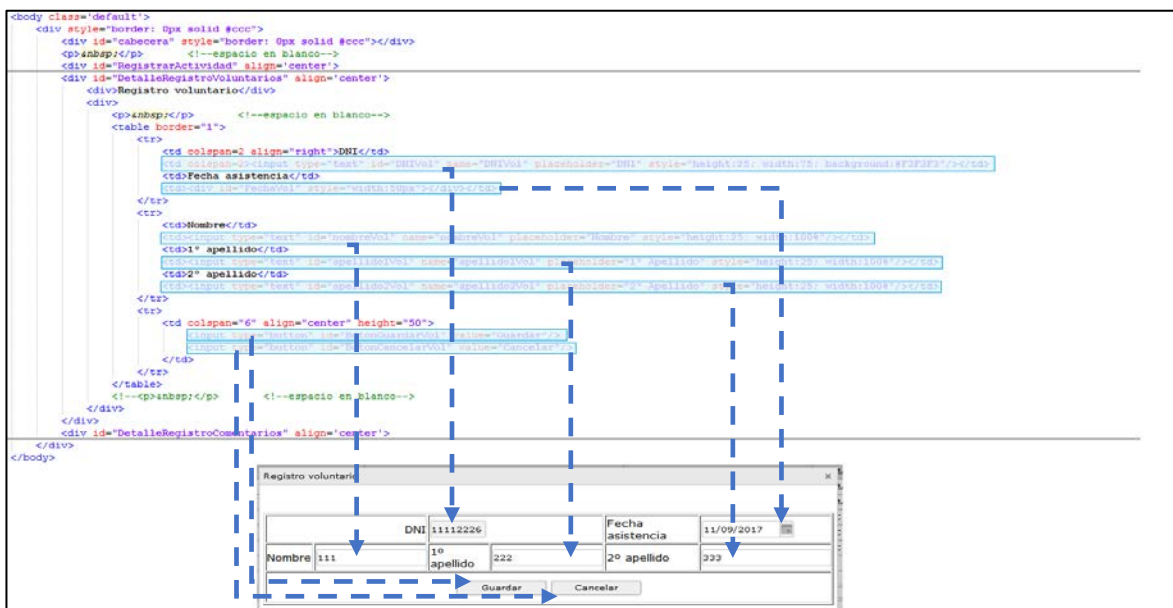


Figura 4-132 interfaceVolRegistarActividad.php - 3

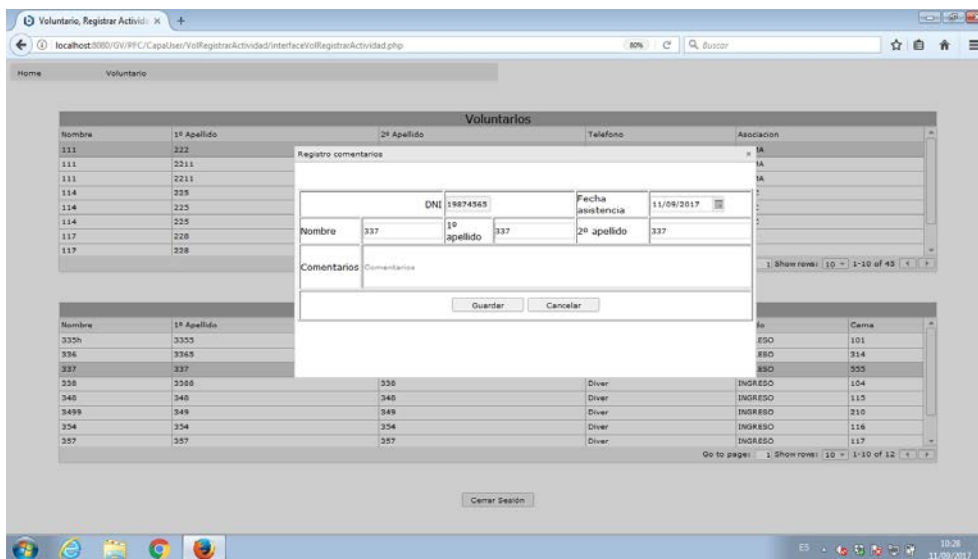


Figura 4-133 Interfaz Registro Actividad – Detalle Usuarios

```

<body class='default'>
  <div style='border: 0px solid #ccc'>
    <div id='cabecera' style='border: 0px solid #ccc'></div>
    <p>&nbsp;</p>
    <div id='RegistrarActividad' align='center'>
    <div id='DetalleRegistroVoluntarios' align='center'>
    <div id='DetalleRegistroComentarios' align='center'>
    <div>Registro comentarios</div>
    <div>
      <p>&nbsp;</p>
      <table border='1'>
        <tr>
          <td colspan=2 align='right'>DNI</td>
          <td colspan=2><input type='text' id='DNI01usa' name='DNI01usa' placeholder='DNI' style='height:25; width:75; background:#F3F3F3'></td>
          <td colspan=2>Fecha asistencia</td>
          <td colspan=2><div id='Fecha01usa' style='width:50px'></div></td>
        </tr>
        <tr>
          <td colspan=2><input type='text' id='nombre01usa' name='nombre01usa' placeholder='Nombre' style='height:25; width:100%'></td>
          <td colspan=2><input type='text' id='apellido01usa' name='apellido01usa' placeholder='1º Apellido' style='height:25; width:100%'></td>
          <td colspan=2><input type='text' id='apellido02usa' name='apellido02usa' placeholder='2º Apellido' style='height:25; width:100%'></td>
        </tr>
        <tr>
          <td colspan=2><input type='text' id='comentarios01usa' name='comentarios01usa' placeholder='Comentarios' style='height:70; width:100%'></td>
        </tr>
        <tr>
          <td colspan=6 align='center' height='5'>
            <input type='button' id='BotonGuardar01usa' value='Guardar' />
            <input type='button' id='BotonCancelar01usa' value='Cancelar' />
          </td>
        </tr>
      </table>
      <p>&nbsp;</p>
    </div>
  </div>
</body>
  
```

Figura 4-134 interfaceVolRegistrarActividad.php - 4

## 4.6 Implementación

A continuación, está detallado el diagrama de clase de la capa de datos del sistema colaborativo, especificando el nombre de la clase y sus atributos. Las operaciones estarán detallar más adelante para facilitar la comprensión, por lo que no se representarán en este diagrama.

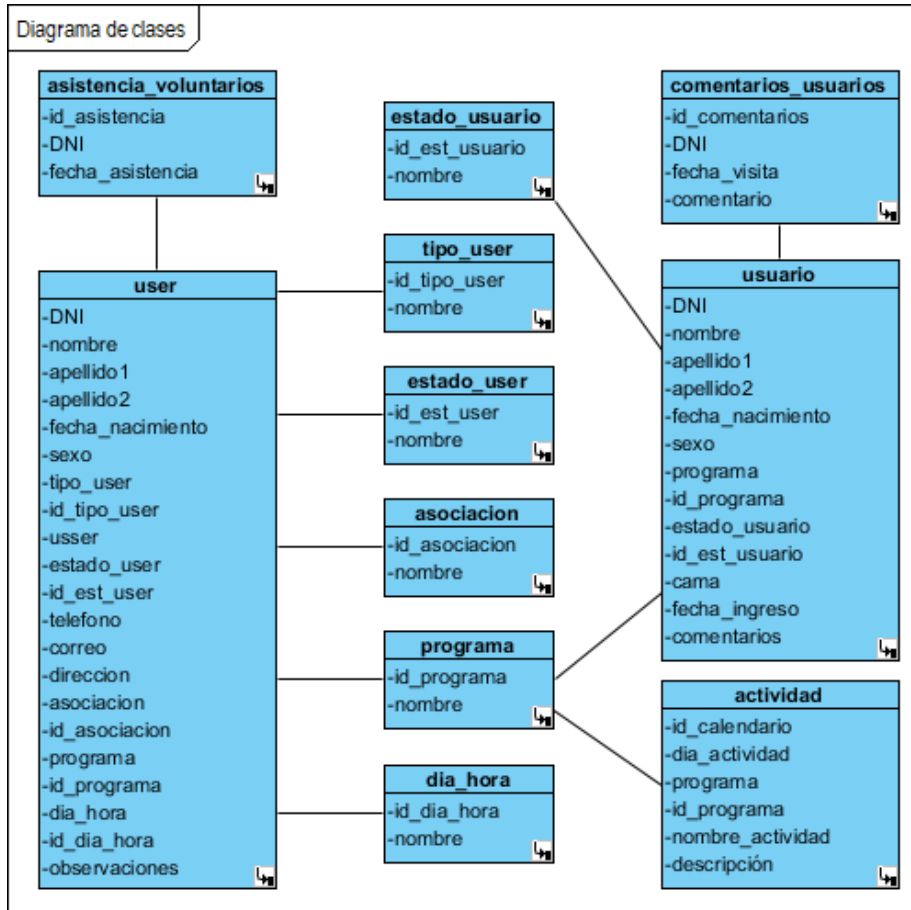


Figura 4-135 Diagrama de clases –Base Datos

### 4.6.1.1 UserBD.php

El diagrama de clase **User** tiene como atributos/elementos: DNI, nombre, apellido1, apellido2, fecha\_nacimiento, sexo, tipo\_user, id\_tipo\_user, usser, estado\_user, id\_est\_user, teléfono, correo, dirección, asociación, id\_asociacion, programa, id\_programa, dia\_hora, id\_dia\_hora, observaciones; y puede realizar las siguientes operaciones/funciones:

- Inicializacion (): coloca los elementos en su estado inicial, sin contenido.
- setUser (): actualiza en bloque todos los elementos de la clase C\_User en la base de datos.
- ActualizaNombreUser (): actualiza el elemento nombre de la clase User.
- ActualizaDNIUser (): actualiza el elemento DNI de la clase User.
- ActualizaLogin (): actualiza los elementos usser y DNI de la clase User.
- ActualizaUser (): actualiza los todos elementos de la clase User.
- ExisteUser (): se conecta con la base de datos y consulta si el user existe en la base de datos.

- `ObtenerUserDNI ()`: accede a la base de datos, consulta el user en el campo DNI de la tabla user y si tiene contenido, obtiene los datos del user.
- `ObtenerUser ()`: accede a la base de datos, consulta el user en el campo usser de la tabla user y si tiene contenido, obtiene los datos del user
- `ObtenerLogin ()`: accede a la base de datos, consulta el user en los campos usser y DNI de la tabla user y si tiene contenido, obtiene los datos del user
- `GuardarUser ()`: se conecta con la base de datos, comprueba si ya existe el user mediante la función `ExisteUser ()`. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- `BuscarUser ()` se conecta con la base de datos, comprueba si ya existe el user mediante la función `ExisteUser ()`. Si existe, obtiene los datos del user en el campo DNI de la tabla user y si tiene contenido, obtiene los datos del user.
- `BorrarUser ()`: obtiene el user mediante la función `ObtenerUser ()` y borra el contenido del user.
- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el user para obtener el listado.
- `Obtener_json_script ()`: obtiene los user mediante la función `Obtener_listado_json ()`.
- `Obtener_listado_json_ModificarVoluntario ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el user para obtener el listado según los campos: `tipo_user`, `nombre`, `programa` y `dia_hora`.
- `Obtener_json_script_ModificarVoluntario ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_ModificarVoluntario ()`.
- `Obtener_json_script_ModificarVoluntario1()` obtiene una cadena con los datos requeridos mediante la función `Obtener_listado_json_ModificarVoluntario ()`.
- `Obtener_listado_json_VoluntariosDia ()` accede a la base de datos, consulta los datos de la base de datos y selecciona el user para obtener el listado según los campos: `tipo_user`, `estado_user`, `programa` y `dia_hora`.
- `Obtener_json_script_VoluntariosDia()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_VoluntariosDia ()`.
- `Obtener_listado_json_RegistroVoluntario ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el user para obtener el listado según los campos: `tipo_user`, `programa` y `dia_hora`.
- `Obtener_json_script_RegistroVoluntario ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_RegistroVoluntario ()`.

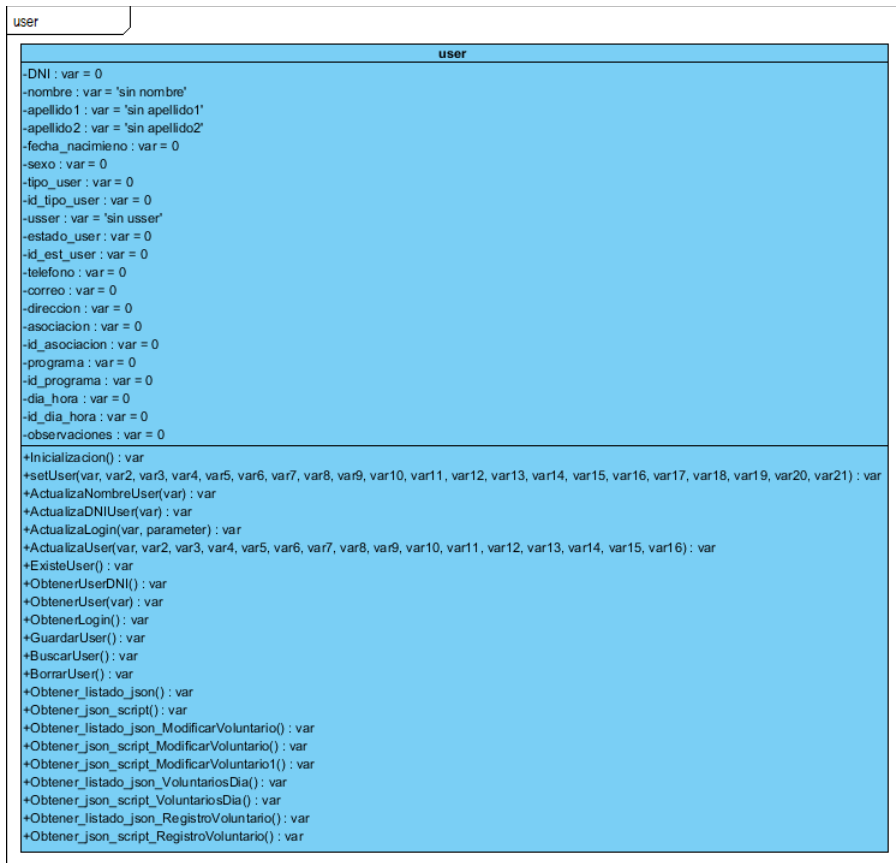


Figura 4-136 Diagrama de clase – userBD

#### 4.6.1.2 UsuarioBD.php

El diagrama de clase **Usuario** tiene como atributos/elementos: DNI, nombre, apellido1, apellido2, fecha\_nacimiento, sexo, programa, id\_programa, estado\_usuario, cama, fecha\_ingreso, comentarios; y puede realizar las siguientes operaciones/funciones:

- **Inicializacion ():** coloca los elementos en su estado inicial, sin contenido.
- **setUsuario ():** actualiza en bloque todos los elementos de la clase C\_Usuario en la base de datos.
- **ActualizaNombreUsuario ():** actualiza el elemento nombre de la clase Usuario.
- **ActualizaDNIUsuario ():** actualiza el elemento DNI de la clase Usuario.
- **ActualizaUsuario ():** actualiza los todos elementos de la clase Usuario.
- **ExisteUsuario ():** se conecta con la base de datos y consulta si el usuario existe en la base de datos.
- **GuardarUsuario ():** se conecta con la base de datos, comprueba si ya existe el usuario mediante la función **ExisteUsuario ()**. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- **BorrarUsuario ():** obtiene el usuario mediante la función **ObtenerUsuario ()** y borra el contenido del usuario.



- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el usuario para obtener el listado.
- `Obtener_json_script ()`: obtiene los usuarios mediante la función `Obtener_listado_json ()`.
- `Obtener_listado_json_AdmVisualizarUsuarios ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el usuario para obtener el listado según los campos: `estado_usuario` y `programa`.
- `Obtener_json_script_AdmVisualizarUsuarios ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_AdmVisualizarUsuarios ()`.
- `Obtener_json_script_AdmVisualizarUsuarios1 ()` obtiene una cadena con los datos requeridos mediante la función `Obtener_listado_json_AdmVisualizarUsuarios ()`.
- `Obtener_listado_json_ModificarAsistencia ()` accede a la base de datos, consulta los datos de la base de datos y selecciona el Usuario para obtener el listado según el campo: `estado_usuario`.
- `Obtener_json_script_ModificarAsistencia ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_ModificarAsistencia ()`.
- `Obtener_listado_json_UsuariosDia ()` accede a la base de datos, consulta los datos de la base de datos y selecciona el Usuario para obtener el listado según los campos: `estado_usuario` y `programa`.
- `Obtener_json_script_UsuariosDia ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_UsuariosDia ()`.
- `Obtener_listado_json_RegistroUsuario ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el Usuario para obtener el listado según los campos: `estado_usuario` y `programa`.
- `Obtener_json_script_RegistroUsuario ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_RegistroUsuario ()`.

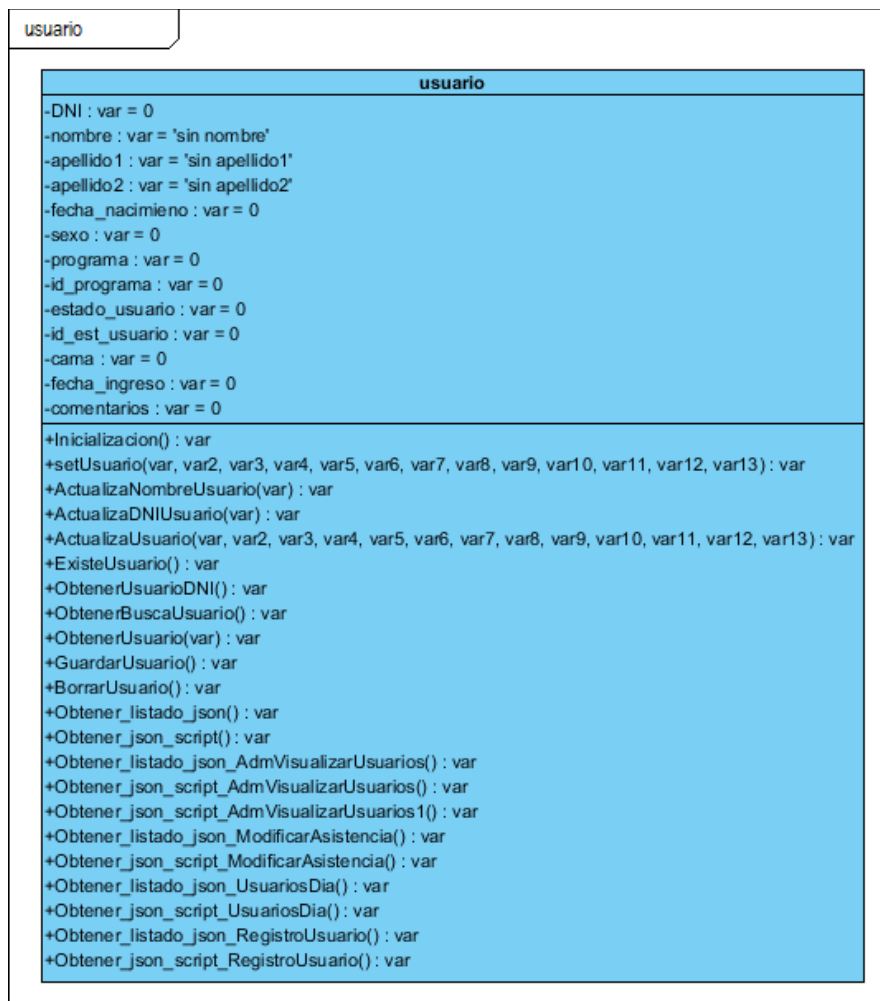


Figura 4-137 Diagrama de clase – usuarioBD

#### 4.6.1.3 Tipo\_userBD.php

El diagrama de clase **Tipo\_user** tiene como atributos/elementos: id\_tipo\_user, nombre; y puede realizar las siguientes operaciones/funciones:

- Inicializacion (): coloca los elementos en su estado inicial, sin contenido.
- Actualiza\_tipo\_user (): actualiza los todos elementos de la clase C\_Tipo\_user.
- Obtener\_tipo\_userid (): accede a la base de datos, consulta el tipo\_user en el campo id\_tipo\_user de la tabla tipo\_user y si tiene contenido, obtiene los datos del tipo\_user.
- Obtener\_tipo\_user (): accede a la base de datos, consulta el tipo\_user en el campo nombre de la tabla tipo\_user y si tiene contenido, obtiene los datos del tipo\_user.
- Guardar\_tipo\_user (): se conecta con la base de datos, comprueba si ya existe el tipo\_user mediante la función Obtener\_tipo\_user (). Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- Borrar\_tipo\_user (): obtiene el tipo\_user mediante la función Obtener\_tipo\_userid ( ) y borra su contenido.
- Obtener\_listado\_json (): accede a la base de datos, consulta los datos de la base de datos y selecciona el tipo\_user para obtener el listado.

- `Obtener_json_script ()`: obtiene los `tipos_user` mediante la función `Obtener_listado_json ()`.

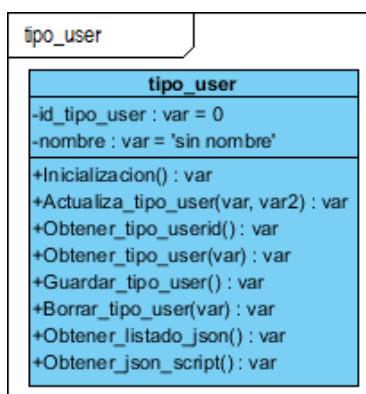


Figura 4-138 Diagrama de clase – `tipo_userBD`

#### 4.6.1.4 Estado\_userBD.php

El diagrama de clase **Estado\_user** tiene como atributos/elementos: `id_est_user`, `nombre`; y puede realizar las siguientes operaciones/funciones:

- `Inicializacion ()`: coloca los elementos en su estado inicial, sin contenido.
- `Actualiza_estado_user ()`: actualiza los todos elementos de la clase `C_Estado_user`.
- `Obtener_estado_userid ()`: accede a la base de datos, consulta el `estado_user` en el campo `id_est_user` de la tabla `estado_user` y si tiene contenido, obtiene los datos del `estado_user`.
- `Obtener_estado_user ()`: accede a la base de datos, consulta el `estado_user` en el campo `nombre` de la tabla `estado_user` y si tiene contenido, obtiene los datos del `estado_user`.
- `Guardar_estado_user ()`: se conecta con la base de datos, comprueba si ya existe el `estado_user` mediante la función `Obtener_estado_user ()`. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- `Borrar_estado_user ()`: obtiene el `estado_user` mediante la función `Obtener_estado_userid ( )` y borra su contenido.
- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el `estado_user` para obtener el listado.
- `Obtener_json_script ()`: obtiene los `estados_user` mediante la función `Obtener_listado_json ()`.

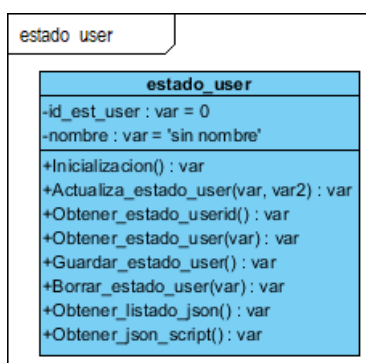


Figura 4-139 Diagrama de clase – `estado_userBD`

#### 4.6.1.5 AsociacionBD.php

El diagrama de clase **Asociación** tiene como atributos/elementos: id\_asociacion, nombre; y puede realizar las siguientes operaciones/funciones:

- Inicializacion (): coloca los elementos en su estado inicial, sin contenido.
- Actualiza\_asociacion (): actualiza los todos elementos de la clase C\_Asociacion.
- Obtener\_asociacionid (): accede a la base de datos, consulta la asociacion en el campo id\_asociacion de la tabla asociacion y si tiene contenido, obtiene los datos del asociacion.
- Obtener\_asociacion (): accede a la base de datos, consulta el asociacion en el campo nombre de la tabla asociacion y si tiene contenido, obtiene los datos del asociacion.
- Guardar\_asociacion (): se conecta con la base de datos, comprueba si ya existe el asociacion mediante la función Obtener\_asociacion (). Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- Borrar\_asociacion (): obtiene el asociacion mediante la función Obtener\_asociacionid ( ) y borra su contenido.
- Obtener\_listado\_json (): accede a la base de datos, consulta los datos de la base de datos y selecciona el asociacion para obtener el listado.
- Obtener\_json\_script (): obtiene las asociaciones mediante la función Obtener\_listado\_json ().

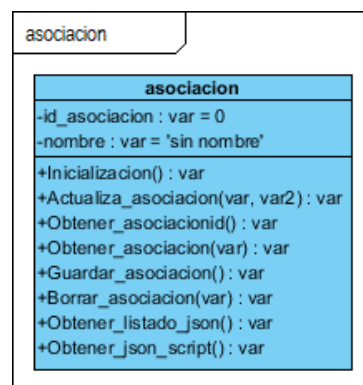


Figura 4-140 Diagrama de clase – asociacionBD

#### 4.6.1.6 ProgramaBD.php

El diagrama de clase **Programa** tiene como atributos/elementos: id\_programa, nombre; y puede realizar las siguientes operaciones/funciones:

- Inicializacion (): coloca los elementos en su estado inicial, sin contenido.
- Actualiza\_programa (): actualiza los todos elementos de la clase C\_Programa.
- Obtener\_programaid (): accede a la base de datos, consulta el programa en el campo id\_programa de la tabla programa y si tiene contenido, obtiene los datos del programa.

- `Obtener_programa ()`: accede a la base de datos, consulta el programa en el campo nombre de la tabla programa y si tiene contenido, obtiene los datos del programa.
- `Guardar_programa ()`: se conecta con la base de datos, comprueba si ya existe el programa mediante la función `Obtener_programa ()`. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- `Borrar_programa ()`: obtiene el programa mediante la función `Obtener_programaid ()` y borra su contenido.
- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el programa para obtener el listado.
- `Obtener_json_script ()`: obtiene las programaes mediante la función `Obtener_listado_json ()`.

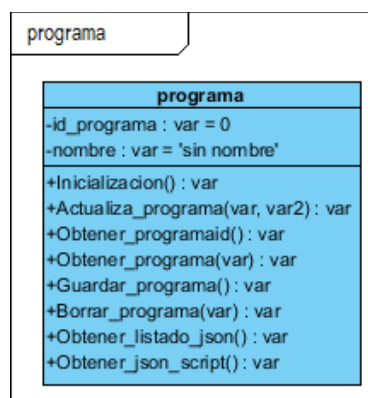


Figura 4-141 Diagrama de clase – programaBD

#### 4.6.1.7 Dia\_horaBD.php

El diagrama de clase **Dia\_hora** tiene como atributos/elementos: `id_dia_hora`, `nombre`; y puede realizar las siguientes operaciones/funciones:

- `Inicializacion ()`: coloca los elementos en su estado inicial, sin contenido.
- `Actualiza_dia_hora ()`: actualiza los todos elementos de la clase `C_Dia_hora`.
- `Obtener_dia_horaid ()`: accede a la base de datos, consulta el dia\_hora en el campo `id_dia_hora` de la tabla `dia_hora` y si tiene contenido, obtiene los datos del `dia_hora`.
- `Obtener_dia_hora ()`: accede a la base de datos, consulta el dia\_hora en el campo nombre de la tabla `dia_hora` y si tiene contenido, obtiene los datos del `dia_hora`.
- `Guardar_dia_hora ()`: se conecta con la base de datos, comprueba si ya existe el `dia_hora` mediante la función `Obtener_dia_hora ()`. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- `Borrar_dia_hora ()`: obtiene el `dia_hora` mediante la función `Obtener_dia_horaid ()` y borra su contenido.
- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el `dia_hora` para obtener el listado.
- `Obtener_json_script ()`: obtiene las `dia_horaes` mediante la función `Obtener_listado_json ()`.

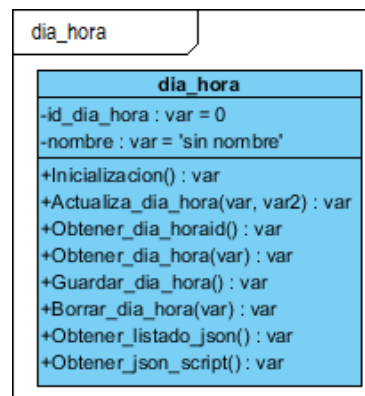


Figura 4-142 Diagrama de clase – dia\_horaBD

#### 4.6.1.8 Estado\_usuarioBD.php

El diagrama de clase **Estado\_usuario** tiene como atributos/elementos: `id_est_usuario`, `nombre`; y puede realizar las siguientes operaciones/funciones:

- `Inicializacion ()`: coloca los elementos en su estado inicial, sin contenido.
- `Actualiza_estado_usuario ()`: actualiza los todos elementos de la clase `C_Estado_usuario`.
- `Obtener_estado_usuarioid ()`: accede a la base de datos, consulta el `estado_usuario` en el campo `id_est_usuario` de la tabla `estado_usuario` y si tiene contenido, obtiene los datos del `estado_usuario`.
- `Obtener_estado_usuario ()`: accede a la base de datos, consulta el `estado_usuario` en el campo `nombre` de la tabla `estado_usuario` y si tiene contenido, obtiene los datos del `estado_usuario`.
- `Guardar_estado_usuario ()`: se conecta con la base de datos, comprueba si ya existe el `estado_usuario` mediante la función `Obtener_estado_usuario ()`. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- `Borrar_estado_usuario ()`: obtiene el `estado_usuario` mediante la función `Obtener_estado_usuarioid ()` y borra su contenido.
- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el `estado_usuario` para obtener el listado.
- `Obtener_json_script ()`: obtiene los `estados_usuario` mediante la función `Obtener_listado_json ()`.

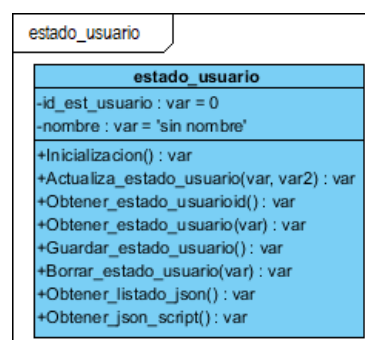


Figura 4-143 Diagrama de clase – estado\_usuarioBD

#### 4.6.1.9 ActividadBD.php

El diagrama de clase **Actividad** tiene como atributos/elementos: `id_calendario`, `dia_actividad`, `programa`, `id_programa`, `nombre_actividad`, `descripcion`; y puede realizar las siguientes operaciones/funciones:

- `Inicializacion ()`: coloca los elementos en su estado inicial, sin contenido.
- `setActividad ()`: actualiza en bloque todos los elementos de la clase `C_Actividad` en la base de datos.
- `ActualizaIdCalendario ()`: actualiza el elemento `idCalendario` de la clase `C_Actividad`.
- `Actualiza_actividad ()`: actualiza los todos elementos de la clase `C_Actividad`.
- `ExisteActividad ()`: se conecta con la base de datos y consulta si la actividad existe en la base de datos.
- `ObtenerActividadid ()`: accede a la base de datos, consulta la actividad en el campo `id_calendario` de la tabla `actividad` y si tiene contenido, obtiene los datos de la actividad.
- `ObtenerActividad ()`: accede a la base de datos, consulta la actividad en el campo `nombre` de la tabla `actividad` y si tiene contenido, obtiene los datos de la actividad.
- `GuardarActividad ()`: se conecta con la base de datos, comprueba si ya existe la actividad mediante la función `ExisteActividad ()`. Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- `BorrarActividad ()`: obtiene la actividad mediante la función `ObtenerActividad ()` y borra el contenido de la actividad.
- `Obtener_listado_json ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona la actividad para obtener el listado.
- `Obtener_json_script ()`: obtiene los actividades mediante la función `Obtener_listado_json ()`.
- `Obtener_listado_json_VisualizarActividad ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona el `Actividad` para obtener el listado.
- `Obtener_json_script_VisualizarActividad ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_VisualizarActividad ()`.
- `Obtener_listado_json_DatosDia ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona la actividad para obtener el listado según el campo: `dia_actividad`.
- `Obtener_json_script_DatosDia1 ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_DatosDia ()`.
- `Obtener_json_script_DatosDia ()`: obtiene una cadena con los datos requeridos mediante la función `Obtener_listado_json_DatosDia ()`.

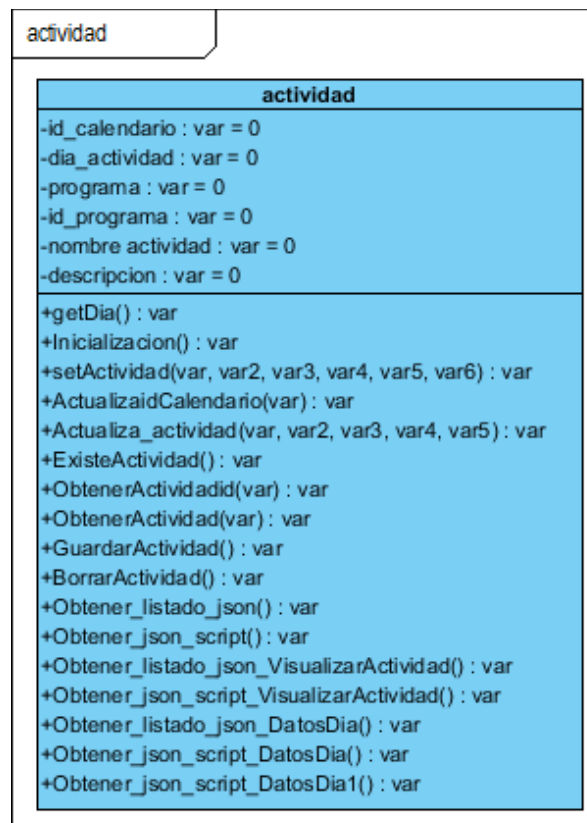


Figura 4-144 Diagrama de clase – actividadBD

#### 4.6.1.10 Asistencia\_voluntariosBD.php

El diagrama de clase **Asistencia\_voluntarios** tiene como atributos/elementos: id\_asistencia,DNI, fecha\_asistencia; y puede realizar las siguientes operaciones/funciones:

- Inicializacion (): coloca los elementos en su estado inicial, sin contenido.
- setAsistencia (): actualiza en bloque todos los elementos de la clase C\_Asistencia en la base de datos.
- ActualizaAsistenciaDNI (): actualiza el elemento DNI de la clase C\_Asistencia.
- ActualizaAsistencia (): actualiza los todos elementos de la clase C\_Asistencia.
- ExisteAsistencia (): se conecta con la base de datos y consulta si la asistencia existe en la base de datos.
- ObtenerAsistenciaid (): accede a la base de datos, consulta la asistencia en el campo id\_asistencia de la tabla asistencia\_voluntarios y si tiene contenido, obtiene los datos de la asistencia.
- GuardarAsistencia (): se conecta con la base de datos, comprueba si ya existe la asistencia mediante la función ExisteAsistencia (). Si existe, se actualiza los datos en la base de datos, sino inserta los nuevos datos.
- BorrarAsistencia (): obtiene la asistencia mediante la función ObtenerAsistenciaid ( ) y borra el contenido de la asistencia.
- Obtener\_listado\_json (): accede a la base de datos, consulta los datos de la base de datos y selecciona la asistencia para obtener el listado.



- `Obtener_json_script ()`: obtiene las asistencias mediante la función `Obtener_listado_json ()`.
- `Obtener_listado_json_AsistenciasDNI ()`: accede a la base de datos, consulta los datos de la base de datos y selecciona la asistencia para obtener el listado según el campo: DNI.
- `Obtener_json_script_AsistenciasDNI ()`: obtiene los datos requeridos mediante la función `Obtener_listado_json_AsistenciasDNI ()`.
- `Obtener_json_script_AsistenciasDNI1 ()`: obtiene una cadena con los datos requeridos mediante la función `Obtener_listado_json_AsistenciasDNI ()`.

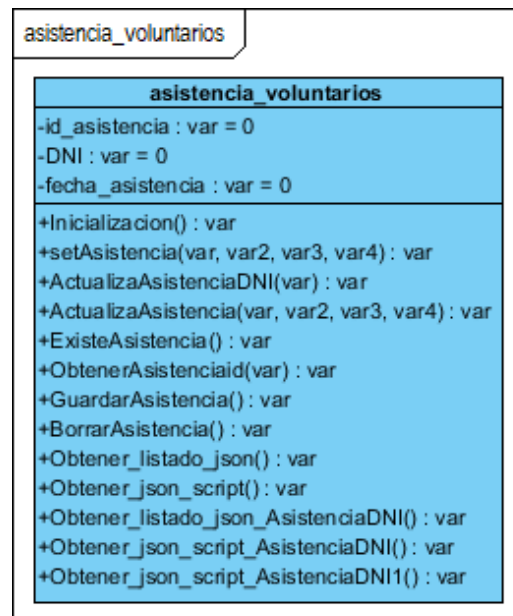


Figura 4-145 Diagrama de clase – asistencia\_voluntariosBD

#### 4.6.1.11 Comentarios\_usuariosBD.php

El diagrama de clase **Comentarios\_usuarios** tiene como atributos/elementos: `id_comentarios`, `DNI`, `fecha_visita`, `comentario`; y puede realizar las siguientes operaciones/funciones:

- `Inicializacion ()`: coloca los elementos en su estado inicial, sin contenido.
- `setComentario ()`: actualiza en bloque todos los elementos de la clase `C_Comentarios` en la base de datos.
- `ActualizaComentarioDNI ()`: actualiza el elemento DNI de la clase `C_Comentarios`.
- `ActualizaComentario ()`: actualiza los todos elementos de la clase `C_Comentarios`.
- `ExisteComentario ()`: se conecta con la base de datos y consulta si el comentario existe en la base de datos.
- `ObtenerComentarioid ()`: accede a la base de datos, consulta el comentario en el campo `id_comentarios` de la tabla `comentarios_usuarios` y si tiene contenido, obtiene los datos de los comentarios.
- `GuardarComentario ()`: se conecta con la base de datos, comprueba si ya existe el comentario mediante la función `ExisteComentario ()`. Si existe, se actualiza los datos en la base de datos, sino inserta

los nuevos datos.

- **BorrarComentario ():** obtiene el comentario mediante la función **ObtenerComentarioid ( )** y borra el contenido del comentario.
- **Obtener\_listado\_json ():** accede a la base de datos, consulta los datos de la base de datos y selecciona el comentario para obtener el listado.
- **Obtener\_json\_script ():** obtiene los comentarios mediante la función **Obtener\_listado\_json ()**.
- **Obtener\_listado\_json\_ComentariosDNI ():** accede a la base de datos, consulta los datos de la base de datos y selecciona el comentario para obtener el listado según el campo: DNI.
- **Obtener\_json\_script\_ComentariosDNI ():** obtiene los datos requeridos mediante la función **Obtener\_listado\_json\_ComentariosDNI ()**.
- **Obtener\_json\_script\_ComentariosDNI1 ()** obtiene una cadena con los datos requeridos mediante la función **Obtener\_listado\_json\_ComentariosDNI ()**.

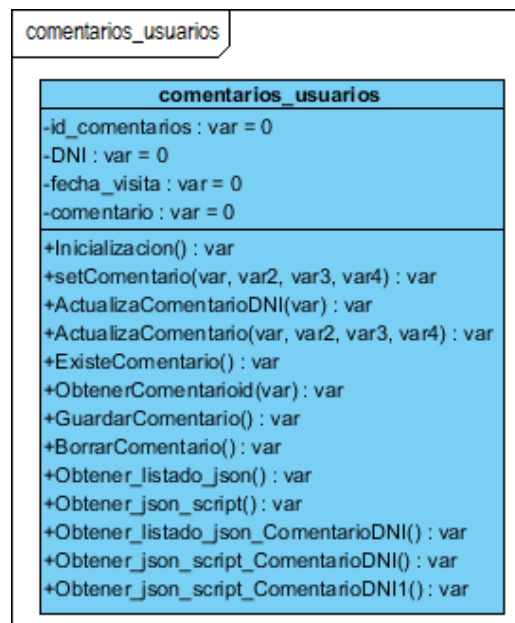


Figura 4-146 Diagrama de clase – comentarios\_usuariosBD

# 5 MANUAL DE USUARIO

En el manual de uso, se detallará la información necesaria para el uso de la aplicación. Como se explicó anteriormente, existen tres tipos de usuarios del sistema o user: administrador, personal y voluntario, teniendo limitadas las acciones según el tipo que presenten. Puesto que cada uno tiene acceso a unas acciones distintas, habrá que detallar el manual de usuario para los tres tipos de user.

El manual de usuario de la aplicación web consta de dos partes:

- Instalación y configuración de la base de datos.
- Instrucciones de uso.

La primera parte de instalación y configuración de la base de datos es común para los tres tipos de user, por lo tanto, sólo será necesario explicarlo una vez. Las instrucciones de uso si serán necesarias detallarlas para los tres tipos de user.

La instalación se explicará desde la configuración previa que es necesaria que presente el ordenador, partiendo de que no hay nada instalado con anterioridad. Una vez explicada la instalación y configuración, se detallará el funcionamiento de la aplicación desde la página inicial hasta pasar por todas las diferentes páginas de las que disponga cada tipo de user en su menú.

## 5.1 Instalación y configuración la base de datos

### 5.1.1 Base de datos del sistema

Lo primero que habrá que hacer para comenzar a utilizar la aplicación es instalar y configurar el servidor web, para ello los pasos a seguir son:

1. Descargar el paquete de instalación de XAMPP, en el siguiente enlace está disponible para Windows, Linux y OS X: <https://www.apachefriends.org/es/>.
2. Seleccionar XAMPP para el sistema de nuestra máquina y descargar software.
3. Ejecutar el archivo .exe y seguir las instrucciones de instalación.

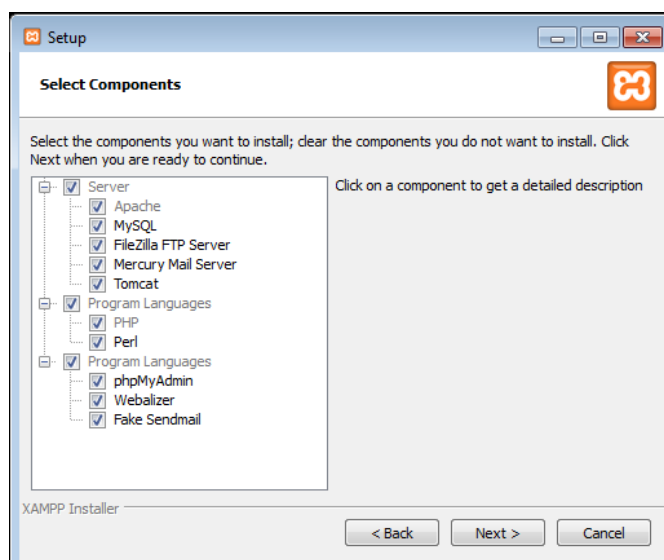


Figura 5-1 Instalación XAMPP

### 5.1.2 Ficheros de la aplicación

Una vez instalado el servidor web, será necesario cargar la base de datos en la herramienta phpMyAdmin, ubicado en la dirección <http://localhost/phpmyadmin>. Para ello deberemos seguir una serie de pasos detallados a continuación:

1. Abrir en el navegador web la dirección de phpMyAdmin.
2. Seleccionar la pestaña 'Base de Datos'; que mostrará una nueva ventana para crear nuevas bases de datos. Para ello bastará con escribir su nombre, en este caso 'gv' y colocar el cotejamiento deseado, en nuestro caso 'utf8\_spanish\_c'.
3. Con la nueva base de datos seleccionada en el árbol de la izquierda, seleccionar la pestaña 'Importar'
4. Seleccionar el archivo sql a importar, indicando que el conjunto de caracteres del archivo es utf-8.
5. Mantener el resto de campos como aparece por defecto y continuar con la importación.

El nombre de la base de datos que se utilice en los archivos php para hacer las consultas siempre tiene que ser el mismo que el creado en phpMyAdmin, en este caso 'gv', en caso contrario la aplicación no podrá conectarse a la base de datos. Así mismo, el nombre de las tablas de dicha base de datos también tiene que corresponderse en la base de datos y en los archivos php para que las consultas den resultado.

Después de cargar la base de datos habrá que conectarla a la aplicación web, para ello deberemos cargar las carpetas de la aplicación que contiene los archivos php y html en el directorio donde esté instalado la herramienta phpMyAdmin. En este caso el directorio donde se subirán todos los archivos de la aplicación es: C:\Bitnami\redmine-3.1.0-0\apache2\htdocs\GV\PFC.

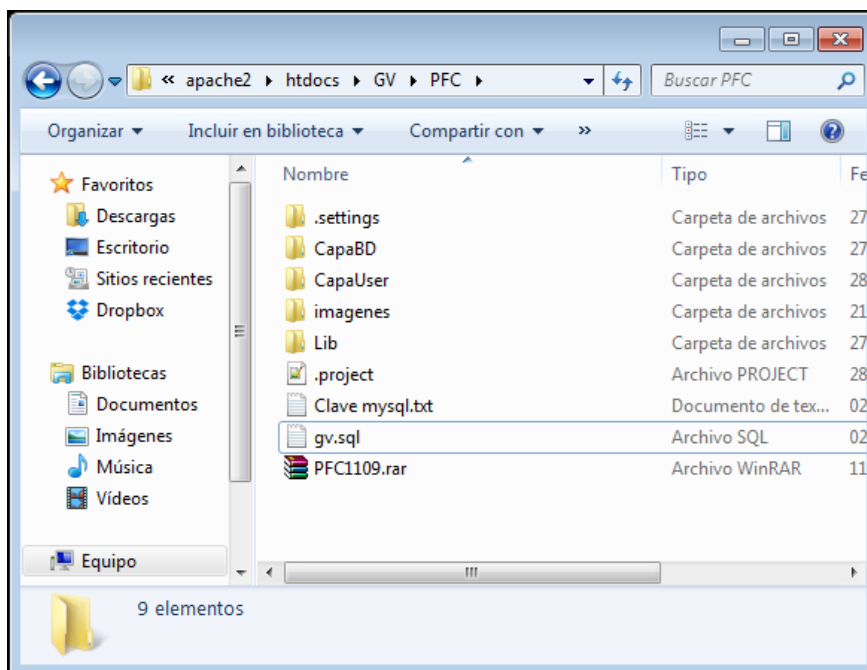


Figura 5-2 Archivos de la aplicación

## 5.2 Instrucciones de uso

Una vez cargada la base de datos y subido los archivos de la aplicación al directorio de phpMyAdmin, podremos acceder a la aplicación abriendo el navegador de internet. Algunos de los navegadores que podemos usar son: Internet Explorer, Google Chrome o Mozilla Firefox.

Habrá que escribir la dirección: <http://localhost:8080/GV/PFC/CapaUser/login/acceso.php> en el navegador, con lo que nos dará acceso a la página inicial. El paso inicial es común a los tres tipos de user.

En la parte central de la página inicial hay dos casillas para que el usuario de la aplicación o user introduzca su nombre de usuario de la aplicación y su contraseña para poder acceder a la página principal.

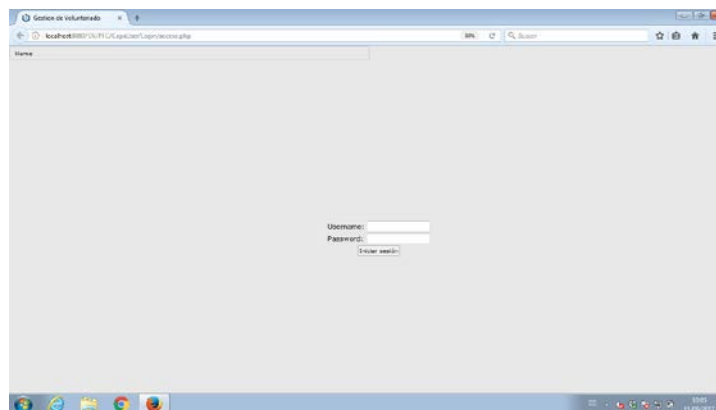


Figura 5-3 Página acceso al sistema

A continuación, se detallarán las acciones que pueden realizar cada user, agrupándolas según el tipo de user registrado en la aplicación.

### 5.2.1 Administrador

#### 5.2.1.1 Página principal del Administrador

En la parte superior aparece el menú con las distintas opciones de que dispone el administrador, este menú se mantendrá para todas las páginas a las que tiene acceso el administrador. Las acciones que puede realizar son:

- Agregar actividad.
- Introducir voluntario.
- Modificar voluntario.
- Visualizar usuarios.

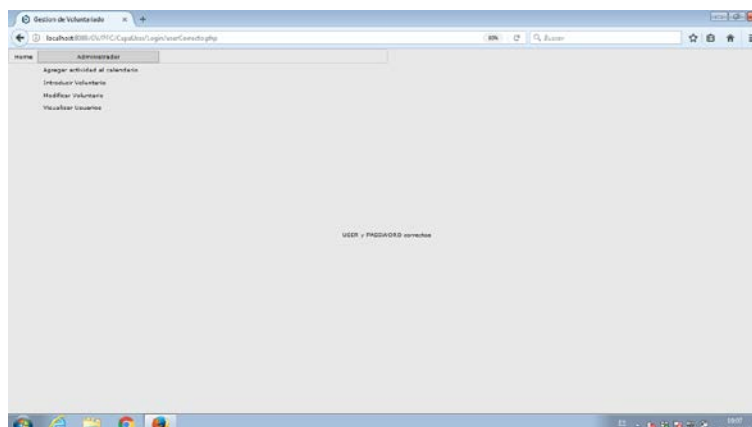


Figura 5-4 Página principal de Administrador

### 5.2.1.2 Agregar actividad

Para gestionar las actividades del voluntariado, el user debe ser tipo Administrador, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Agregar Actividad del menú.
- Introducir una nueva actividad:
  2. Rellenar los campos.
    - Día.
    - Programa.
    - Nombre.
    - Descripción.
    - El campo identificador no hay que rellenarlo, se autocompleta sólo al guardar.
  3. Pulsar sobre el botón 'guardar'.

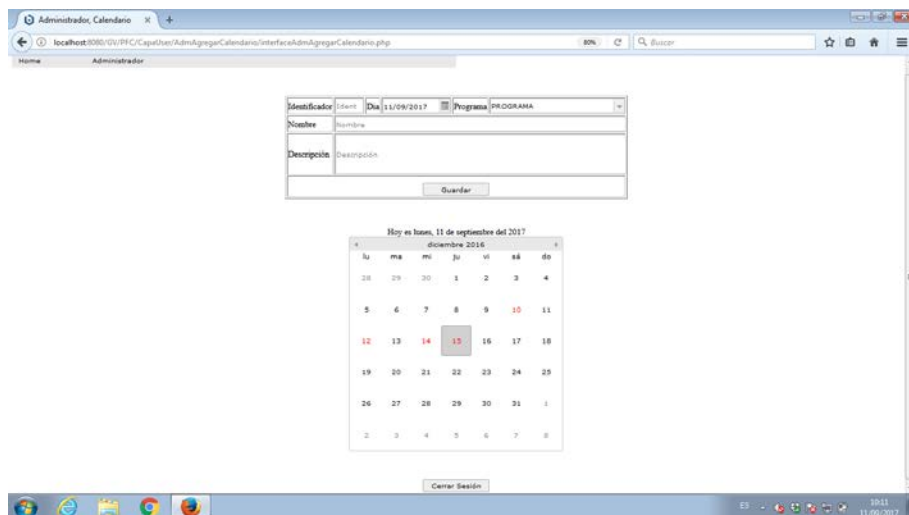


Figura 5-5 Página para gestionar las actividades

- Visualizar una actividad ya registrada:
  2. Seleccionar día en el calendario con doble clic.
  3. Seleccionar actividad en la lista de actividades con doble clic.

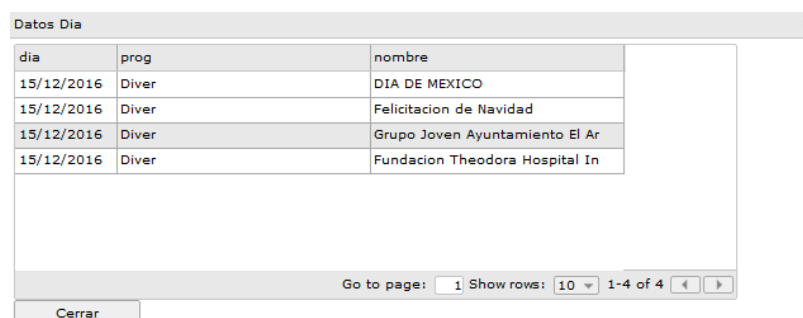


Figura 5-6 Página para seleccionar la actividad

- Modificar una actividad ya registrada (previa visualización de la actividad ya registrada):
  4. Realizar los cambios necesarios en los campos ya rellenos.
    - Día.
    - Programa.
    - Nombre.
    - Descripción.
    - El campo identificador no hay que modificarlo.
  5. Pulsar sobre el botón 'guardar'.

### 5.2.1.3 Introducir voluntario

Para gestionar a los voluntarios, el user debe ser tipo Administrador, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Introducir Voluntario del menú.
2. Rellenar el campo DNI de la parte superior.
3. Pulsar sobre el botón 'comprobar'.
4. Rellenar los campos.
  - Nombre.
  - 1º Apellido.
  - 2º Apellido.
  - Fecha de nacimiento.
  - Sexo (M/F).
  - Tipo user (Administrador/Personal/Voluntario).
  - Dirección.
  - Estado (ALTA/BAJA).
  - Teléfono.
  - Correo.
  - User.
  - Asociación.
  - Programa.
  - Día/Hora.
  - Observaciones.
5. Pulsar sobre el botón 'guardar'.

Es posible introducir un nuevo voluntario sin comprobar previamente si existe en la base de datos, saltándose los pasos 1, 2 y 3, y realizando sólo el 4 y 5, pero no es recomendable pues puede ocasionar errores por duplicar los datos.

DNI: 12345678

Comprobar

Datos Voluntario						
DNI: 12345678						
Nombre	Madministrador	1º apellido	gieras	2º apellido	caso	
Fecha nacimiento	01/01/1990	Sexo	M	Tipo user	ADMINISTRADOR	
Dirección	35638	Estado	ALTA			
Teléfono	123456789	Código	5548	User	marjolca	
Asociación	ADOMA	Programa	Diver	Día/Hora	Lunes Mañana	
Observaciones	44587					

Guardar Cerrar Sesión

Figura 5-7 Página para gestionar a los voluntarios

#### 5.2.1.4 Modificar voluntario

Para modificar a los voluntarios, el user debe ser tipo Administrador, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Agregar Actividad del menú.
2. Rellenar los campos (uno o varios).
  - Nombre.
  - Programa.
  - Día/Hora.
3. Pulsar sobre el botón 'buscar'.
4. Seleccionar voluntario en la lista de voluntarios con doble clic.

Nombre: nombre

Programa: Unidad de Donación de Tiempo Libre

Día/Hora: Lunes Tarde

Buscar

DNI	Nombre	1º Apellido	2º Apellido	Programa	Día	Estado
11112227	112	223	334	Unidad de Donación de Tiempo L...	Lunes Tarde	ALTA
11112228	112	223	334	Unidad de Donación de Tiempo L...	Lunes Tarde	ALTA
11112229	112	223	334	Unidad de Donación de Tiempo L...	Lunes Tarde	ALTA
11112230	112	223	337	Unidad de Donación de Tiempo L...	Miércoles	ALTA
11112231	112	226	337	Unidad de Donación de Tiempo L...	Miércoles	ALTA
11112232	112	226	337	Unidad de Donación de Tiempo L...	Miércoles	ALTA
11112233	112	229	340	Unidad de Donación de Tiempo L...	Jueves Tar	ALTA
11112234	112	229	340	Unidad de Donación de Tiempo L...	Jueves Tar	ALTA

Go to page: 1 Show rows: 10 of 71

Cerrar Sesión

Figura 5-8 Página para buscar y seleccionar voluntarios



5. Realizar los cambios necesarios en los campos ya rellenos.
  - Nombre.
  - 1º Apellido.
  - 2º Apellido.
  - Fecha de nacimiento.
  - Sexo (M/F).
  - Tipo user (Administrador/Personal/Voluntario).
  - Dirección.
  - Estado (ALTA/BAJA).
  - Teléfono.
  - Correo.
  - User.
  - Asociación.
  - Programa.
  - Día/Hora.
  - Observaciones.

The screenshot shows a web form titled "Datos Voluntario" with a close button (x) in the top right corner. The form contains the following fields and values:

DNI 11112287				
Nombre	115	1º apellido	226	2º apellido
Fecha nacimiento	01/05/1980	Sexo	M	Tipo user
Dirección	55702			Estado
Teléfono	123456793	Correo	5552	User
Asociación	ADIMA	Programa	Unidad de Donacion de Tiempo L	Dia/Hor.
Observaciones	44591			
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>				

Figura 5-9 Página para modificar a los voluntarios

#### 5.2.1.5 Visualizar usuarios

Para visualizar a los usuarios del voluntariado, el user debe ser tipo Administrador, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Visualizar Usuarios del menú.
2. Rellenar el campo Programa de la parte superior.

3. Pulsar sobre el botón 'comprobar'.
4. Seleccionar usuario en la lista de usuarios con doble clic.

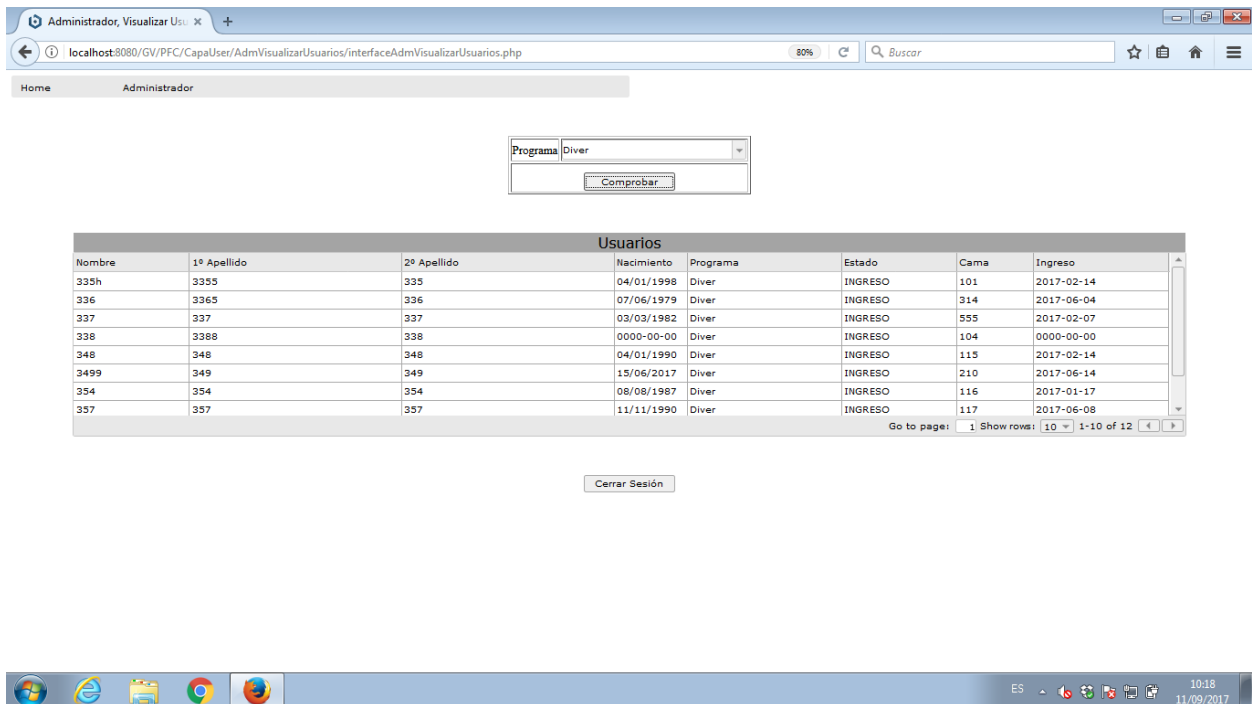


Figura 5-10 Página para buscar y seleccionar usuarios

5. Visualizar los comentarios del usuario seleccionado.



Figura 5-11 Página para buscar usuarios

6. Pulsar sobre el botón 'cancelar'.

## 5.2.2 Personal

### 5.2.2.1 Página principal del Personal

En la parte superior aparece el menú con las distintas opciones de que dispone el personal, este menú se mantendrá para todas las páginas a las que tiene acceso el administrador. Las acciones que puede realizar son:

- Solicitar asistencia.
- Modificar asistencia.

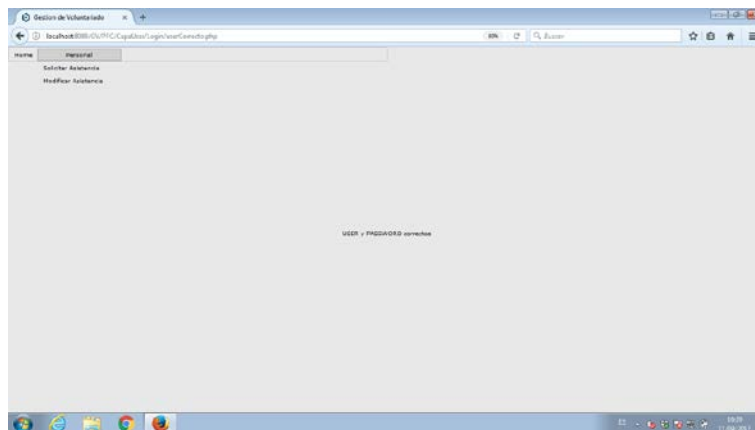


Figura 5-12 Página principal de Personal

### 5.2.2.2 Solicitar asistencia

Para solicitar un nuevo acompañamiento, el user debe ser tipo Personal, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Solicitar Asistencia del menú.
2. Rellenar el campo DNI de la parte superior.
3. Pulsar sobre el botón 'comprobar'.
4. Rellenar los campos.
  - Nombre.
  - 1º Apellido.
  - 2º Apellido.
  - Fecha de nacimiento.
  - Sexo (M/F).
  - Estado (INGRESO/ALTA).
  - Programa.
  - Fecha de ingreso.
  - Cama.
  - Comentario.

### 5. Pulsar sobre el botón 'guardar'.

Es posible solicitar una nueva asistencia sin comprobar previamente si el usuario existe en la base de datos, saltándose los pasos 1, 2 y 3, y realizando sólo el 4 y 5, pero no es recomendable pues puede ocasionar errores duplicar los datos.

Personal Solicitar Asistencia x

localhost:8080/GV/PFC/Capitulos/PerSolicitarAsistencia/interfacePerSolicitarAsistencia.php

Home Personal

DNI 19874560

Comprobar

Datos Usuario					
DNI 19874560					
Nombre	334	1º apellido	334	2º apellido	334
Fecha nacimiento	20/04/2017	Sexo	M	Estado	ALTA
Programa	Ocupación de Tiempo Libre para	Fecha ingreso	20/06/2017	Cama	112
Comentarios	654654				

Guardar Cerrar Sesión

ES 11:00 11/09/2017

Figura 5-13 Página para solicitar un nuevo acompañamiento

### 5.2.2.3 Modificar asistencia

Para modificar los acompañamientos, el user debe ser tipo Personal, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Modificar Asistencia del menú.
2. Seleccionar usuario en la lista de usuarios con doble clic.

Personal Modificar Asisten... x

localhost:8080/GV/PFC/Capitulos/PerModificarAsistencia/interfacePerModificarAsistencia.php

Home Personal

DNI	Nombre	1º Apellido	2º Apellido	Programa	Estado	Cama	Comentarios
19874561	3344	3345a	334d	Unidad de Donación de Tiempo Libre	INGRESO	102	AluAlu
19874562	334444	334f	334e	Unidad de Donación de Tiempo Libre	INGRESO	102	AluAlu
19874563	3356	3355	335	Diver	INGRESO	101	luj
19874564	326	3265	326	Diver	INGRESO	314	Riquelme
19874565	337	337	337	Diver	INGRESO	555	luj
19874566	338	3388	338	Diver	INGRESO	104	luj
19874568	340	340	340	Unidad de Donación de Tiempo Libre	INGRESO	502	judik
19874570	3422	3422	342	Ocupación de Tiempo Libre para Niños Hospitaliza.	INGRESO	113	judik

Cerrar Sesión

Go to page: 1 Show rows: 10 1-10 of 32

ES 11:00 11/09/2017

Figura 5-14 Página para seleccionar usuarios

3. Realizar los cambios necesarios en los campos ya rellenos.

- Nombre.
- 1º Apellido.
- 2º Apellido.
- Programa.
- Estado (INGRESO/ALTA).
- Cama.
- Comentarios.

Datos Usuario				
DNI 19874561				
Nombre	3344	1º apellido	3345s	2º apellido
Programa	Unidad de Donacion de Tiempo L	Estado	INGRESO	Cama
Comentarios	ñkñk			
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>				

Figura 5-15 Página para modificar a los usuarios

## 5.2.3 Voluntario

### 5.2.3.1 Página principal del Voluntario

En la parte superior aparece el menú con las distintas opciones de que dispone el voluntario, este menú se mantendrá para todas las páginas a las que tiene acceso el administrador. Las acciones que puede realizar son:

- Visualizar calendario.
- Visualizar voluntarios del día.
- Visualizar usuarios del día.
- Registrar actividad.

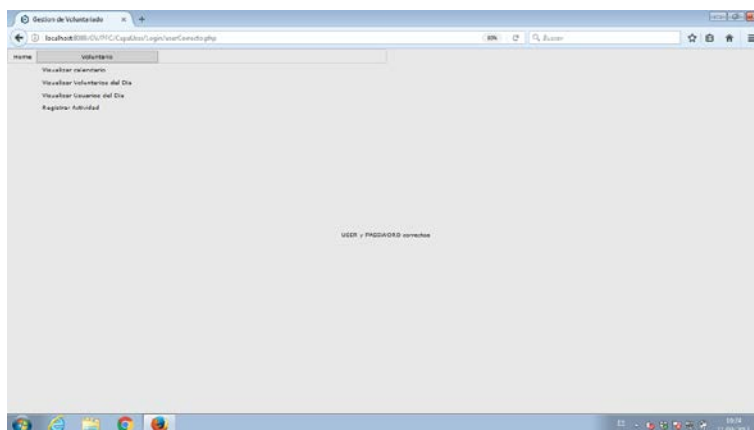


Figura 5-16 Página principal de Voluntario

### 5.2.3.2 Visualizar actividad

Para visualizar las actividades del voluntariado, el user debe ser tipo Voluntario, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Visualizar Calendario del menú.
2. Seleccionar día en el calendario con doble clic.

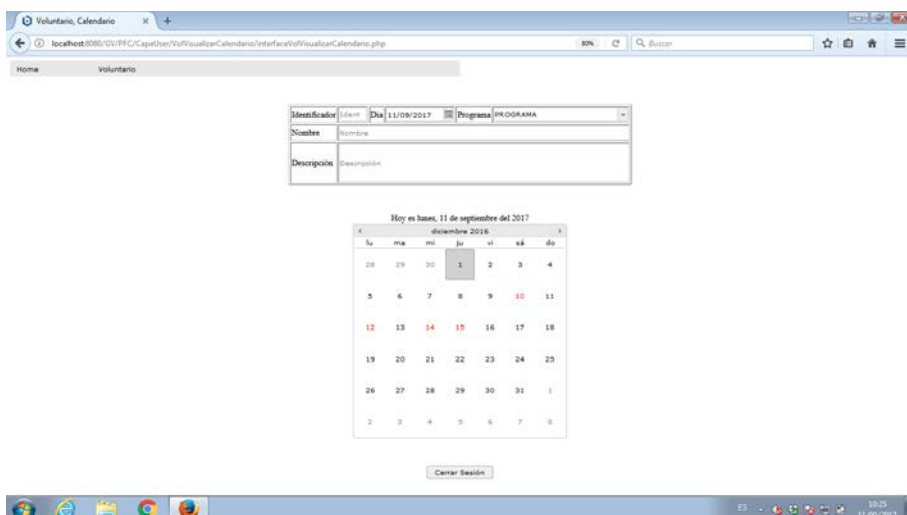


Figura 5-17 Página para visualizar las actividades

3. Seleccionar actividad en la lista de actividades con doble clic.

dia	prog	nombre
15/12/2016	Diver	DIA DE MEXICO
15/12/2016	Diver	Felicitation de Navidad
15/12/2016	Diver	Grupo Joven Ayuntamiento El Ar
15/12/2016	Diver	Fundacion Theodora Hospital In

Figura 5-18 Página para seleccionar la actividad

4. Visualizar los datos de la actividad seleccionada.

### 5.2.3.3 Visualizar voluntaries del día

Para visualizar a los voluntarios del día, el user debe ser tipo Voluntario, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Visualizar Voluntarios del día del menú.

Nombre	1º Apellido	2º Apellido	Telefono	Asociacion
117	228	329	123456795	CRJ
117	228	329	123456795	CAJ
117	228	329	123456795	CRJ
126	237	348	123456804	ABCC
126	237	348	123456804	ABCC
135	246	357	123456813	ADIMA
135	246	357	123456813	ADIMA

Figura 5-19 Página para visualizar los voluntarios del día

2. Visualizar los voluntarios que asisten el día.

### 5.2.3.4 Visualizar usuarios del día

Para visualizar a los usuarios del día, el user debe ser tipo Voluntario, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:

1. Seleccionar la opción de Visualizar Usuarios del día del menú.
2. Seleccionar usuario en la lista de usuarios con doble clic.

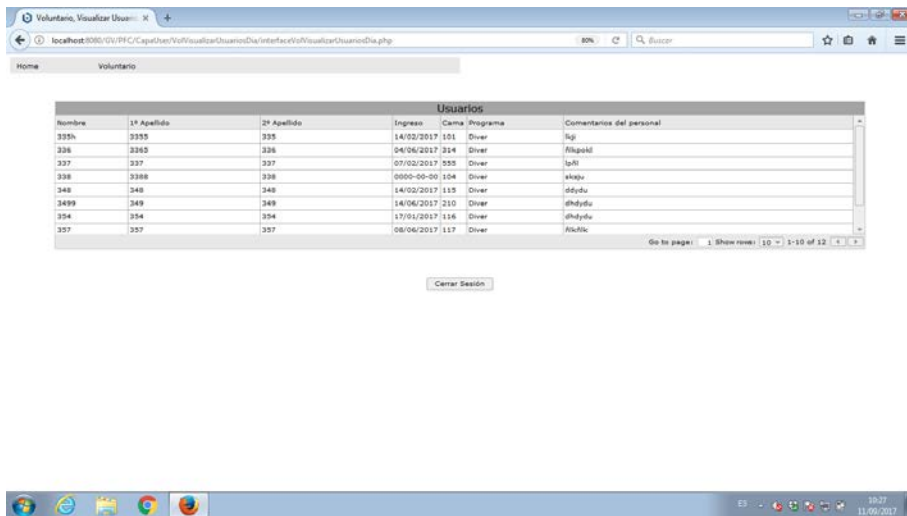


Figura 5-20 Página para seleccionar usuarios

3. Visualizar los comentarios del usuario seleccionado.



Figura 5-21 Página para buscar usuarios

4. Pulsar sobre el botón 'cancelar'.

### 5.2.3.5 Registrar actividad

Para registrar la actividad realizada por el voluntariado, el user debe ser tipo Voluntario, el resto de tipos de user no tendrán acceso para realizar esta acción.

Una vez se ha accedido al sistema, se deben seguir los siguientes pasos:



1. Seleccionar la opción de Registrar Actividad del menú.

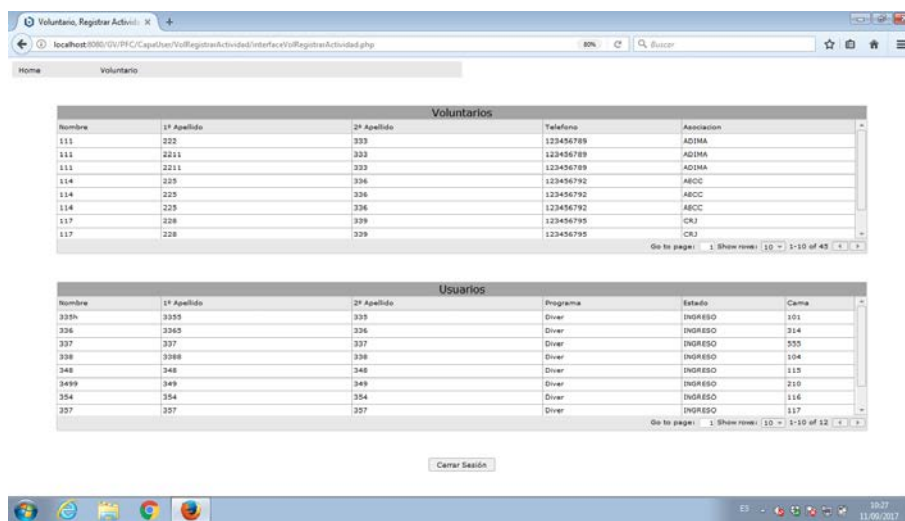


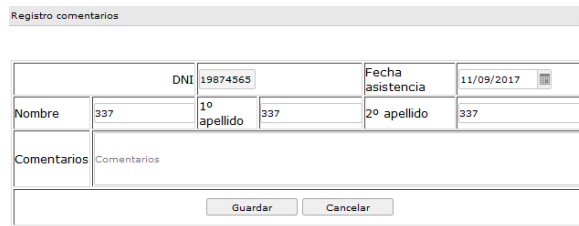
Figura 5-22 Página para registrar la actividad realizada

- Registrar actividad realizada por el voluntario:
  2. Seleccionar el voluntario en la lista de voluntarios con doble clic.
  3. Rellenar los campos:
    - Fecha de asistencia.
    - El resto de campos no hay que rellenarlos ni modificarlos.

DNI		11112226		Fecha asistencia		11/09/2017	
Nombre	111	1º apellido	222	2º apellido	333		
Guardar				Cancelar			

Figura 5-23 Página para registrar las actividades realizadas por los voluntarios

4. Pulsar sobre el botón 'guardar'.
- Registrar acompañamiento al usuario:
    2. Seleccionar el usuario en la lista de usuarios con doble clic.
    3. Rellenar los campos:
      - Fecha de asistencia.
      - Comentarios.
      - El resto de campos no hay que rellenarlos ni modificarlos.



Registro comentarios					
DNI		19874565		Fecha asistencia	11/09/2017
Nombre	337	1º apellido	337	2º apellido	337
Comentarios	Comentarios				
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>					

Figura 5-24 Página para registrar los acompañamientos a los usuarios

4. Pulsar sobre el botón 'guardar'.

Los pasos 2, 3 y 4 (tanto para los voluntarios como los acompañamientos), se realizarán tantas veces como sea necesario para registrar la actividad de todos los voluntarios y todos los acompañamientos.

# 6 CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

---

En este trabajo se ha obtenido un sistema de información colaborativo que hace posible una mejor gestión de los distintos programas de voluntariados presentes en los centros hospitalarios mediante el uso de los recursos digitales, gracias a que estos sustituyen al formato en papel consiguiendo mayor flexibilidad a la hora de organizar a los voluntarios, las actividades que estos realizan y a los usuarios que acompañan durante su estancia en el hospital. Por todo ello, los objetivos conseguidos serían:

- Disminución del número de hojas de inscripción del voluntariado, haciendo que la inscripción sea estándar y más sencilla.
- Disminución del número de hojas de actividad del voluntario, haciendo que el registro de actividades realizadas posibilite una mejor atención futura.
- Mejora de la asignación de recursos humanos a los turnos de actividad.
- Mejora de los acompañamientos a los usuarios recién ingresados, ya que las solicitudes de acompañamiento se realizarán de forma continua en vez de forma puntual.
- Mejora de la atención en el acompañamiento a los usuarios que han estado ingresados anteriormente.
- Mejora de la visibilidad de actividades no cotidianas realizadas en el hospital mediante el calendario de actividades.

Las futuras líneas de trabajo para obtener mejoras del sistema serán:

- Adaptar la aplicación a una plataforma móvil, considerando los distintos tamaños de estos dispositivos y la protección de datos personales tanto de voluntarios como de usuarios.
- Ampliar la aplicación para poder realizar informes periódicos que muestren las actividades realizadas y las asistencias de voluntarios y usuarios.
- Ampliar la aplicación para que los usuarios puedan solicitar ellos mismos los acompañamientos, inscribiendo sus datos a la espera de asignación de programa de voluntariado específico de acompañamiento.
- Mejorar la interfaz para que sea más estética de cara a los usuarios.
- Mejorar las búsquedas para que permita más opciones de filtrado.



# REFERENCIAS

---

- [1] Ciclo de vida del desarrollo software [09-10-2017]. Disponible en Web: <[https://www.tutorialspoint.com/es/software\\_engineering/software\\_development\\_life\\_cycle.htm](https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm)>
- [2] Que es la metodología orientada a objetos – Introducción UML, Alex Quintanilla [20-09-2017]. Disponible en web: <[https://www.academia.edu/15943563/Que\\_es\\_la\\_Metodolog%C3%ADa\\_orientada\\_a\\_objetos\\_-\\_Introducci%C3%B3n\\_a\\_UML](https://www.academia.edu/15943563/Que_es_la_Metodolog%C3%ADa_orientada_a_objetos_-_Introducci%C3%B3n_a_UML)>
- [3] NDT-Suite, una solución práctica para el uso de NDT [20-09-2017]. Disponible en web: <[https://www.researchgate.net/publication/319109607\\_NDT-Suite\\_una\\_solucion\\_practica\\_para\\_el\\_uso\\_de\\_NDT](https://www.researchgate.net/publication/319109607_NDT-Suite_una_solucion_practica_para_el_uso_de_NDT)>
- [4] Especificación de requerimientos, Departamento de Ciencias de la Computación e I.A., Universidad de Granada [20-09-2017]. Disponible en web: <<http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>>
- [5] Requisitos, Ingeniería del Software I 3º I.T.I. Gestión Miguel A. Laguna [20-09-2017]. Disponible en web: <<https://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>>
- [6] 3. Metodología orientada al objeto [20-09-2017]. Disponible en web: <[ftp://www.dlsi.ua.es/people/jaime/apuntes/isi\\_tema3.1.pdf](ftp://www.dlsi.ua.es/people/jaime/apuntes/isi_tema3.1.pdf)>
- [7] Documentación de requisitos mediante casos de uso, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla [20-09-2017]. Disponible en web: <<http://www.lsi.us.es/docencia/get.php?id=2008>>
- [8] Diagramas de navegación en aplicaciones web.pdf [21-09-2017]. Disponible en web: <<http://revistavinculos.udistrital.edu.co/files/2013/09/Diagramas-de-navegaci%C3%B3n-en-aplicaciones-web.pdf>>
- [9] Página oficial Apache [22-09-2017]. Disponible en Web: <<https://www.apachefriends.org/es/index.html>>
- [10] Página oficial phpMyAdmin [22-09-2017]. Disponible en Web: <<https://www.phpmyadmin.net/>>
- [11] Página sobre phpMyAdmin [22-09-2017]. Disponible en Web: <<http://paradigmasiut.blogspot.com.es/2012/11/phpmyadmin.html>>
- [12] Arquitectura 3 capas [22-09-2017]. Disponible en Web: <<https://es.slideshare.net/Decimo/arquitectura-3-capas>>
- [13] Página oficial Adobe. Aspectos básicos de las aplicaciones web [21-09-2017]. Disponible en Web: <<https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>>
- [14] Enciclopedia Wikipedia. Ref. *HTML* [22-09-2017]. Disponible en Web: <<https://es.wikipedia.org/wiki/HTML>>
- [15] Pagina PHP [20-09-2017]. Disponible en Web: <<http://php.net/manual/es/intro-what-is.php>>
- [16] Enciclopedia Wikipedia. Ref. *PHP* [22-09-2017]. Disponible en Web: <<https://es.wikipedia.org/wiki/PHP>>
- [17] Enciclopedia Wikipedia. Ref. *CSS* [22-09-2017]. Disponible en Web: <[https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)>
- [18] Enciclopedia Wikipedia. Ref. *Javascript* [22-09-2017]. Disponible en Web: <<https://es.wikipedia.org/wiki/JavaScript>>

- [19] Enciclopedia Wikipedia. Ref. *MySQL* [22-09-2017]. Disponible en Web: <<https://es.wikipedia.org/wiki/MySQL>>
- [20] Los 10 Frameworks\_Librerías más populares de Javascript [16-10-2017]. Disponible en Web: <<https://www.genbetadev.com/javascript/los-10-frameworks-librerias-mas-populares-de-javascript>>
- [21] Los 10 mejores editores gratuitos de HTML, CSS y JavaScript [10-10-2017]. Disponible en Web: <<https://www.campusmvp.es/recursos/post/Los-10-mejores-editores-gratuitos-de-HTML-CSS-y-JavaScript.aspx>>
- [22] Página oficial Notepad++ [10-10-2017]. Disponible en Web: <<https://notepad-plus-plus.org/>>
- [23] Página oficial KompoZer [10-10-2015]. Disponible en Web: <<http://www.kompozer.net/>>
- [24] Enciclopedia Wikipedia. Ref. *Kompozer* [10-10-2017]. Disponible en Web: <<https://es.wikipedia.org/wiki/KompoZer>>
- [25] Enciclopedia Wikipedia. Ref. *Dreamweaver* [10-10-2017]. Disponible en Web: <[https://es.wikipedia.org/wiki/Adobe\\_Dreamweaver](https://es.wikipedia.org/wiki/Adobe_Dreamweaver)>