

Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

Grado en Ingeniería Informática

Aplicación web para la gestión
de voluntarios, colaboradores
y donaciones de la Fundación
Spínola Solidaria

Ubicación del código fuente:

<https://github.com/joakDA/gestor-ongd-sps>

Trabajo Fin de Grado

Presentado por: Duro Arribas, Joaquín Alejandro

Director: Pedraza Gómara, Luis

Resumen

En este trabajo se describen todos los aspectos referentes al desarrollo de una aplicación web que permite gestionar la información de los voluntarios, colaboradores, donaciones, sedes y la gestión de la seguridad de los datos de una organización sin ánimo de lucro para el desarrollo (ONGD).

La aplicación, ya implantada en la intranet de la “Fundación Spínola Solidaria” funciona en un entorno compuesto por un servidor Internet Information Services (IIS) y Microsoft SQL Server y está desarrollada principalmente en ASP.NET con arquitectura Modelo-Vista-Controlador (MVC), accediendo a determinadas funcionalidades mediante REST API's.

Se ha utilizado SCRUM como metodología ágil de desarrollo para planificar el proyecto y GIT como herramienta de control de versiones del código fuente.

El desarrollo de esta aplicación permite optimizar los procesos de gestión de una ONGD, teniendo centralizada toda la información de la organización y permitiendo generar los certificados de donaciones y las remesas bancarias pulsando un botón en vez de generándolos manualmente uno a uno.

Palabras Clave: ONGD, Aplicación-Web, Voluntarios, Spínola Solidaria

Abstract

This document describes all the aspects related to the development of a web application that allows to manage the information of volunteers, collaborators, donations, headquarters and data security management of a non-profit organization for development (NGDO).

The application, already implemented in the intranet of the "Fundación Spinola Solidaria" works in an environment composed of an Internet Information Services (IIS) server and Microsoft SQL Server and is developed mainly in ASP.NET with Model-View-Controller architecture (MVC), accessing certain functionalities through REST API's.

SCRUM has been used as an agile development methodology to plan the project and GIT as a source code version control tool.

The development of this application allows to optimize the management processes of an NGDO, having centralized all the information of the organization and allowing to generate the certificates of donations and the bank remittances at the push of a button instead of generating them manually one by one.

Keywords: NGDO, Web-Application, Volunteers, Spínola Solidaria

“The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.” (W3C Consortium, 2017)

“Everybody in this country should learn how to program a computer...because it teaches you how to think” by Steve Jobs” (Quigley, 2012)

“There are no problems, only solutions” (John Lennon)

“Sólo tú puedes decidir qué hacer con el tiempo que se te ha dado” (J.R. Tolkien, El Señor de los Anillos)

“Aquellos que tienen el privilegio de saber tienen la obligación de actuar” (Albert Einstein)

Agradecimientos

Primeramente, dar las gracias a la Fundación Spínola Solidaria por ofrecerme la posibilidad de realizar este proyecto de fin de grado.

Agradecer el trabajo y ayuda aportada por mi director Luis Pedraza Gómara durante estos meses, que ha contribuido a que este proyecto haya salido adelante.

Y, por último, quiero dar especialmente las gracias a mis padres por haber sido un ejemplo de vida y enseñarme a ser mejor cada día.

Índice de contenidos

1. Introducción.....	12
1.1. Motivación	12
1.2. Objetivos	13
1.2.1. Objetivo general.....	13
1.2.2. Objetivos específicos.....	13
1.3. Aportación realizada	14
1.4. Metodología de trabajo.....	15
1.5. Estructura del trabajo.....	19
2. Contexto y estudio preliminar	21
2.1. Contexto del Trabajo	21
2.2. Soluciones existentes	23
2.2.1. Soluciones específicas para organizaciones sin ánimo de lucro	23
2.2.2. Soluciones genéricas que pueden ser adaptadas	28
2.2.3. Conclusiones del análisis de soluciones existentes.....	29
3. Identificación de requisitos	32
3.1. Alcance del proyecto	32
3.2. Descripción de los usuarios del sistema	32
3.3. Product Backlog	32
3.4. Perspectiva del producto.....	39
3.5. Requisitos no funcionales.....	39
3.6. Modelo de Dominio	40
3.7. Casos de Uso.....	42
3.7.1. Gestión de Voluntarios.....	42
3.7.2. Gestión de sedes y delegaciones	42
3.7.3. Gestión de colaboradores y donaciones	43
3.7.4. Gestión LOPD	44
3.7.5. Administración y configuración	44
3.7.6. Gestión de usuarios.....	45
4. Descripción del proyecto	45
4.1. Aportación realizada	45
4.2. Plan de trabajo.....	46
4.3. Descripción de la aplicación.....	48

5. Descripción técnica	54
5.1. Arquitectura	54
5.1.1. Capa de Presentación.....	56
5.1.2. Lógica de Negocio o Modelo	57
5.1.3. Capa de Datos	59
5.1.4. Patrones de diseño.....	62
5.1.5. Patrón auxiliar Modelo-Vista-Controlador.....	62
5.1.6. Patrón auxiliar “Singleton”	63
5.1.7. Patrón DTO (Data Transfer Object) u Objeto de Transferencia de Datos.....	63
5.1.8. Patrón DAO (Data Access Object) u Objeto de Acceso a Datos.....	64
5.1.9. Patrón “Active Record”	65
5.1.10. Patrón “Layer Supertype”	65
5.2. Prototipo gráfico	65
5.2.1. Vista de la pantalla listado de colaboradores	65
5.3. Tecnologías utilizadas.....	66
5.3.1. Desarrollo del proyecto con ASP.NET	66
5.3.2. Desarrollo del modelo y el controlador en C# .NET	66
5.3.3. Diseño gráfico y responsive con Bootstrap	67
5.3.4. Optimización del estilo del portal web con CSS.....	67
5.3.5. Mejora de funcionalidades con Javascript y JQuery	67
5.3.6. Manipulación de datos en la base de datos mediante Transact-SQL y Entity Framework	68
5.3.7. Creación del fichero del banco con las remesas bancarias en formato XML.....	69
5.3.8. Generación de las remesas bancarias con PHP.....	69
5.3.9. Intercambio de datos entre las diferentes capas en formato JSON	69
5.4. Herramientas utilizadas.....	70
5.4.1. ArgoUML	70
5.4.2. Microsoft Visual Studio 2015 Enterprise.....	70
5.4.3. Microsoft Visual Studio Team Services	70
5.4.4. Mockaroo	70
5.4.5. Bootstrap Studio.....	70
5.4.6. Microsoft Project 2016 Profesional	71
5.4.7. SQL Server 2016	71
5.4.8. Selenium IDE	71
5.4.9. Apache JMeterTM.....	71

5.4.10. TAW	71
5.5. Implementación.....	72
6. Evaluación y pruebas.....	74
6.1. Plan de pruebas	74
6.1.1. Pruebas unitarias.....	74
6.1.2. Pruebas de regresión	76
6.1.3. Pruebas basadas en requerimientos.....	77
6.2. Usabilidad.....	79
6.3. Accesibilidad	80
7. Conclusiones y trabajo futuro	82
7.1. Conclusiones respecto a los objetivos iniciales	82
7.2. Trabajo futuro.....	83
Bibliografía	85
Listado de acrónimos	88

Índice de Ilustraciones

Ilustración 1. Scrum process flow	16
Ilustración 2. Board Sprint 6 Gestor ONG	18
Ilustración 3. Ejemplo de un listado de contactos de CiviCRM	24
Ilustración 4. Listado de socios y el detalle de uno de ellos en GONG.	26
Ilustración 5. Ejemplo del perfil de un usuario en Sinergia CRM	27
Ilustración 6. Ejemplo del formulario de donación de Microsoft Dynamics CRM.	29
Ilustración 7. Modelo de Dominio	41
Ilustración 8. Diagrama de casos de uso para el módulo "voluntarios"	42
Ilustración 9. Diagrama de casos de uso para el módulo "sedes y delegaciones"	43
Ilustración 10. Diagrama de casos de uso del módulo "colaboradores y donaciones"	43
Ilustración 11. Diagrama de casos de uso para el módulo "LOPD"	44
Ilustración 12. Diagrama de casos de uso del módulo "administración y configuración"	44
Ilustración 13. Validación Usabilidad y Accesibilidad Web.	45
Ilustración 14. Gráfico con los diferentes elementos de trabajo del proyecto.	46
Ilustración 15. Diagrama de Gant del proyecto.	47
Ilustración 16. Página de inicio de la aplicación web.	48
Ilustración 17. Menú secundario al colocar el ratón sobre un elemento del menú principal.	48
Ilustración 18. Implementación de la vista "Listado de Voluntarios"	49
Ilustración 19. Listado de colaboradores filtrado por periodo trimestral	49
Ilustración 20. Formulario de creación de un voluntario.	50
Ilustración 21. Paso 1 del formulario de creación de un nuevo colaborador.	51
Ilustración 22. Paso 2 del formulario de creación de un nuevo colaborador.	51
Ilustración 23. Vista de eliminación de un colaborador y su donación asociada.	52
Ilustración 24. Vista del mensaje informativo al usuario después de realizar una operación.	52
Ilustración 25. Ejemplo de certificado de donaciones realizadas por un colaborador en PDF	53
Ilustración 26. Diagrama de red de la arquitectura "cliente-servidor"	54
Ilustración 27. Arquitectura cliente-servidor 3-tier	55
Ilustración 28. Arquitectura de red del sistema	55
Ilustración 29. Implementación de la vista "Editar nuevo Voluntario"	56
Ilustración 30. Diagrama de clases parcial compuesto por las clases que incluyen otras clases.	57
Ilustración 31. Diagrama de clases de la aplicación.	58
Ilustración 32. Vistas de la base de datos que componen el modelo.	59

Ilustración 33. Modelo Entidad-Relación de la base de datos.	60
Ilustración 34. Modelo de datos utilizando "entity-framework"	61
Ilustración 35. Proyecto ASP.NET con MVC	63
Ilustración 36. Diferencias comunicación patrón "DTO"	64
Ilustración 37. Vista de un listado de la aplicación	66
Ilustración 38. Carpetas que componen el código del proyecto.	72
Ilustración 39. Ejecución del caso de prueba para crear una nueva sede con Selenium IDE.	75
Ilustración 40. Cuando se introducen datos incorrectos, la aplicación muestra errores de validación.	75
Ilustración 41. Prueba de regresión del menú principal con resultado fallido.	76
Ilustración 42. Resultado correcto al ejecutar las pruebas de regresión del módulo de sedes.	77
Ilustración 43. Rendimiento de la aplicación con 5 usuarios simultáneos.	78
Ilustración 44. Media, Mediana, Desviación y Rendimiento de la aplicación con 5 hilos.	78
Ilustración 45. Validación correcta del código HTML.	79
Ilustración 46. Resumen de resultados de la validación de la accesibilidad web.	81

Índice de tablas

Tabla 1. Relación de roles y personas en Scrum	16
Tabla 2. Comparativa de las soluciones similares estudiadas	31
Tabla 3. Product Backlog Items Feature Gestión de Voluntarios	34
Tabla 4. Product Backlog Items Feature Gestión de Colaboradores y Donaciones	35
Tabla 5. Product Backlog Items Feature Gestión de Sedes y Delegaciones	37
Tabla 6. Product Backlog Items Feature Administración y Configuración	37
Tabla 7. Product Backlog Items Feature Gestión LOPD	38
Tabla 8. Relación de la arquitectura con las tecnologías utilizadas	56
Tabla 9. Tabla comparativa de los test de rendimiento con 5, 10 y 20 usuarios.	79

1. Introducción

En el presente capítulo se introducen las motivaciones que han llevado al autor del documento a desarrollar este proyecto, los objetivos que se quieren conseguir, la aportación realizada, así como una descripción de la estructura del documento.

1.1. Motivación

Dentro de las diferentes posibilidades de elección de una temática para el proyecto de fin de grado, se ha decidido implementar un gestor web de información de una organización sin ánimo de lucro para el desarrollo (ONGD) como respuesta a una necesidad real de optimizar procesos, automatizando tareas que actualmente se realizan individualmente y de forma manual, como la emisión individual de certificados de colaboradores o la selección de las donaciones mes a mes para emitir las correspondientes remesas bancarias, que se realizan con una hoja de Microsoft Excel o un documento Microsoft Word y en las que se invierte un esfuerzo considerable.

Respecto a la primera funcionalidad, la de emitir certificados de forma manual, actualmente se realiza con un procesador de textos y mediante una plantilla, teniendo el empleado que buscar e introducir de forma manual datos como el NIF/CIF, el nombre y apellidos de cada colaborador o sumar la cantidad total donada a la fundación teniendo en cuenta aspectos como la periodicidad de la donación (mensual, bimensual, trimestral, semestral y anual) o la fecha de alta de la donación, con la posibilidad de que haya errores a la hora de realizar los cálculos. Aunque esta operación sólo se realiza 1 vez de forma anual, si se considera que la generación de cada certificado supone un tiempo medio de 5 minutos, teniendo en cuenta que actualmente la organización tiene unos 300 colaboradores, se tardarían unas 25 horas aproximadamente en generar todos los certificados.

En cuanto a la funcionalidad de emitir las remesas bancarias, actualmente es necesario realizar un procesamiento manual cada mes para poder generar dichas remesas. Este proceso consiste en seleccionar de toda la base de datos de colaboradores (unos 300 registros aproximadamente), generando un archivo Excel con los adeudos correspondientes a cada mes, teniendo en cuenta la periodicidad de las donaciones (mensual, bimensual, trimestral, semestral y anual), lo que supone un arduo trabajo y la posibilidad de cometer errores por parte del empleado de la fundación, además de repercutir en la financiación de la organización. Posteriormente, una vez se ha generado el fichero Excel con los datos necesarios (que consume aproximadamente unas 8-10 horas de tiempo al mes), el empleado debe subir el fichero generado a un programa proporcionado por el banco para que genere el fichero XML en formato SEPA con la información de las remesas bancarias.

Como define la Confederación Española de Cajas de Ahorro (CECA), SEPA son las siglas en inglés de “Single Euro Payments Area”, es decir, Zona única de Pagos en Euros. Su objetivo es establecer instrumentos de pago, infraestructuras y estándares paneuropeos uniformes en toda Europa y eliminar las diferencias entre pagos transfronterizos y pagos nacionales dentro de la zona SEPA. (CECA, 2015, pág. 2)

Otro de los problemas que puede tener una organización de este tipo es la de tener la información de manera centralizada y permitir un uso concurrente de la información. De hecho, actualmente, la información de los colaboradores y los voluntarios se almacena en un fichero Access protegido por contraseña, lo que impide el acceso de varios empleados a la información simultáneamente y supone un riesgo para la seguridad de la información ya que estas contraseñas son fáciles de descifrar.

Además, aunque estos ficheros están protegidos por una contraseña, estos archivos no cumplen con el nivel de seguridad exigido por la Ley de Protección de Datos de Carácter Personal (LOPD), por lo que era necesaria la acometida de este proyecto para cumplir con la legislación vigente antes de la siguiente auditoría de seguridad.

Por último, aunque existen herramientas ya desarrolladas para este tipo de organizaciones (detallados en la sección 2.2, se ha decidido desarrollar un software a medida por tener éstos un precio de licencias bastante considerable o requerir de un gran trabajo de adaptación a las necesidades específicas de la organización.

1.2. Objetivos

Con el desarrollo de este trabajo de fin de grado, se pretenden alcanzar los siguientes objetivos ordenados en función de la prioridad (de más prioritario a menos):

1.2.1. Objetivo general

Desarrollar un sistema basado en web que permita gestionar una base de datos de los voluntarios, colaboradores y donativos de la organización, automatizar los trámites a realizar con las entidades bancarias y las administraciones, así como recopilar datos provenientes de los diferentes sistemas de información ya existentes.

1.2.2. Objetivos específicos

1. Analizar el estado del arte de los sistemas de gestión de organizaciones sin ánimo de lucro o sistemas genéricos que puedan ser adaptados para cubrir las necesidades del proyecto.
2. Implementar un sistema adaptado a las recomendaciones de usabilidad y accesibilidad web según las iniciativas del Consorcio W3C con un nivel mínimo de accesibilidad de doble A (AA) (W3C Consortium, 2017).

3. Ampliar los conocimientos del desarrollador del proyecto en tecnologías Microsoft, como ASP.NET, C# y SQL Server.
4. Utilizar herramientas de control de versiones para gestionar el código fuente.
5. Aplicar conocimientos adquiridos durante el curso de adaptación al grado sobre un proyecto real, aplicando metodologías ágiles para la gestión del proyecto.
6. Analizar “*frameworks*”, “API”s o librerías de terceros (con la licencia apropiada) que ayuden a la implementación de ciertos requisitos o faciliten el desarrollo.
7. Adaptar el software desarrollado, para que sea aplicable a cualquier organización sin ánimo de lucro, permitiendo el cambio de elementos gráficos o cualquier otra personalización que se requiera.

1.3. Aportación realizada

Se ha desarrollado una aplicación web utilizando ASP.NET y C# que conecta con una base de datos Microsoft SQL Server 2016 para almacenar los datos a gestionar por la organización como los datos de los voluntarios, la información de los colaboradores y sus donaciones. La aplicación es una herramienta para la organización que sirve también para registrar las incidencias de seguridad que se pueden producir con los datos de carácter personal, así como para registrar las entradas y salidas de información susceptibles de ser protegidas según marca la Ley Orgánica de Protección de Datos Personales (LOPD).

El tiempo de desarrollo del proyecto ha sido de aproximadamente de 4 meses y medio dividido en 13 “*sprints*” de 2 semanas de duración, habiendo acometido tareas de análisis de la solución, diseño, implementación y pruebas, tras los cuales permitieron realizar la implantación del sistema en producción en las instalaciones de la organización.

Dos de las mejoras sustanciales respecto a cómo se gestionan las donaciones actualmente en la organización de forma manual, son la introducción de la funcionalidad de gestión de remesas bancarias y la emisión de certificados de donaciones para la declaración de la Renta. Éstas suponen un ahorro considerable de tiempo y de recursos para la organización que se pueden emplear en otras actividades.

Como se comenta en la sección 1.1, para la generación de los certificados de la renta se requería buscar toda la información y emplear unas 25 horas aproximadamente al año. Con la aplicación desarrollada, el empleado sólo tiene que pulsar sobre un botón con el rótulo “GENERAR CERTIFICADOS” y la aplicación se encarga de reunir la información y generar los ficheros correspondientes de forma automática, optimizando el esfuerzo dedicado a no más de 1 minuto.

De forma análoga, para la emisión de las remesas bancarias, el sistema le proporciona al empleado de la fundación el fichero XML en 1 minuto aproximadamente, evitando tener que emplear unas 8-10 horas mensuales a dicha tarea.

Respecto al módulo de la “LOPD”, con la aplicación es posible gestionar las incidencias de seguridad y las entradas y salidas de información de carácter personal, permitiendo obtener los datos en formato PDF o generando informes para ser utilizados en el caso de que la organización tenga que realizar una auditoría de cómo se gestionan estos datos.

Otro de los beneficios de esta aplicación es que realiza las operaciones en un tiempo muy reducido, no superando la carga de cada página los 2 segundos, y facilitando la interacción del usuario mediante una interfaz gráfica similar en todas las páginas que componen la aplicación web.

Por último, cabe destacar que el software desarrollado cumple con el nivel “Triple A” de accesibilidad web, tal como se comenta en la sección 6.3

1.4. Metodología de trabajo

La metodología del proyecto utilizada para el desarrollo ha sido SCRUM.

“Scrum es un método de desarrollo de software ágil creado por Jeff Sutherland a principios de los años 90 cuyos principios están basados en los del manifiesto ágil y son usados para guiar las actividades de desarrollo como requisitos, análisis, diseño, evolución y entrega.” (Pressman, 2010)

Existen tres roles en Scrum, denominados “*Scrum team*”, “*Product owner*” y “*ScrumMaster*”:

- ✓ **Scrum tem:** Incluye al *product owen*, al *ScrumMaster* y a los miembros técnicos del equipo, mientras que se denomina “*Scrum delivery team*” al subequipo formado sólo por los miembros técnicos. Su principal misión es abordar un problema (un requisito seleccionado desde la lista sin ordenar denominada “*Product Backlog*” y proponer soluciones innovadoras. Son los que planean, estiman y se comprometen de su trabajo y su propósito es lanzar un entregable del producto en cada *sprint*. (Viscardi, 2013)
- ✓ **Product owner:** Es el responsable del éxito del producto, es la persona que conoce su mercado y las necesidades de los usuarios y su principal misión es guiar al equipo de trabajo a desarrollar una versión del producto *sprint* tras *sprint*. (Viscardi, 2013)
- ✓ **ScrumMaster:** Es el encargado de salvaguardar el proceso y dar lo máximo para que el desarrollo fluya lo máximo posible, evitando interrupciones del equipo y facilitando el desarrollo de las reuniones de *Scrum*. (Viscardi, 2013)

A continuación, en la Tabla 1 se relacionan los diferentes roles presentes en el proyecto con las personas que han desempeñado cada rol:

Tabla 1. Relación de roles y personas en Scrum

Rol	Persona
Product Owner (Cliente)	Directora de la Fundación Spínola Solidaria
Scrum Master	Joaquín Alejandro Duro Arribas
Scrum Team	Joaquín Alejandro Duro Arribas

Un *sprint* es una iteración definida por un comienzo y una fecha de finalización y se inicia con la fase denominada “*sprint planning*” dónde se planifica lo que se va a desarrollar en ese sprint y acaba con el “*sprint review and retrospective*”, en el que se analizan los resultados del sprint que ha finalizado. (Viscardi, 2013)

En la Ilustración 1 se describe el flujo de trabajo de SCRUM, cómo se trasladan los elementos del “sprint backlog”, las reuniones que deben realizarse (de manera diaria y de 15 minutos de duración). Por último, se indica que cada sprint debe dar lugar, al menos, a una nueva funcionalidad.

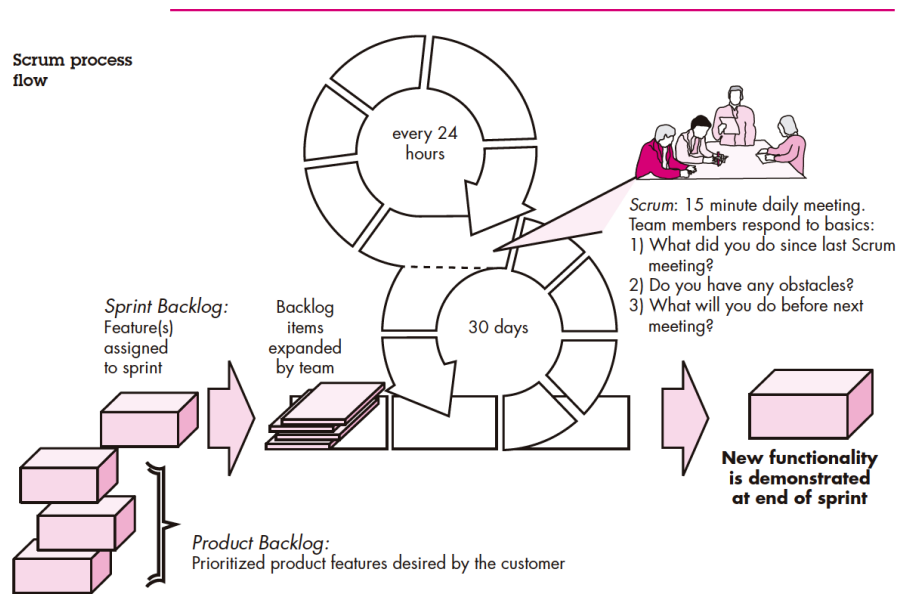


Ilustración 1. Scrum process flow

Al comienzo del proyecto, se definió durante la fase de análisis el “*product backlog*” con las diferentes “*features*” e “*historias de usuarios*” que componían los requerimientos del proyecto (véase Product Backlog) de manera priorizada según la importancia que tiene cada “*product backlog ítem*” en el funcionamiento diario de los empleados de la fundación.

Según la metodología SCRUM, el proyecto debe dividirse en periodos de tiempo fijos conocidos como “*sprints*”. Para el desarrollo del proyecto, se decidió que cada sprint durara

14 días, dónde al principio del sprint se ha hecho una reunión entre el “*Scrum Master*”, el “*Scrum Team*” y el “Product Owner” para decidir todos aquellos PBI (*product backlog ítems*) que se van a desarrollar en el “*sprint*” que comienza.

Durante el “*sprint*”, cada miembro del equipo de trabajo accede a un *tablero compartido por todo el “Scrum Team” (board)*, con las tareas o corrección de “*bugs*” que tiene asignadas y debe realizar en ese “*sprint*”, como se muestra en la Ilustración 2:

Cada día del “*sprint*”, el equipo de trabajo realiza una mini-reunión de no más de 15 minutos para que cada miembro del equipo comunique el estado de sus tareas y ponga en conocimiento del grupo algún problema que haya tenido para que los compañeros le ayuden.

Por último, al final de cada “*sprint*”, se hace una reunión de seguimiento del proyecto donde se analiza cómo se ha desarrollado durante este periodo de tiempo, ver los problemas, aplicar medidas correctoras de los mismos y en el caso de no haber terminado un “*product backlog ítem*”, éste se traslada al siguiente “*sprint*”.

The screenshot displays a Jira Sprint Board for 'GestorONG Team Sprint 6'. The board is organized into columns representing different stages of the sprint cycle: 'To do' (11h), 'New' (2h), 'Approved' (1h), 'In progress' (2h), 'Committed', and 'Done'. Each column contains task cards with titles, assignees (Joaquín Duro), and estimated hours. The 'To do' column has two cards: 'Como propietario de la aplicación web, deseo que todas las páginas del servicio contengan el mismo encabezado de página y el mismo "footer"' (1h) and 'Como empleado quiero dar de alta un nuevo voluntario' (15h). The 'New' column has three cards: 'Cambio Logo actual por logo en blanco' (1h), 'Crear controlador Voluntarios' (4h), and 'Personalizar apariencia del formulario de inserción de un nuevo voluntario' (4h). The 'Approved' column has one card: '404 al abrir la vista "Create" de un voluntario' (1h). The 'In progress' column has one card: 'Probar inserción de nuevo voluntario' (2h). The 'Done' column has four cards: 'Cambio contenido página de inicio' (1h), 'Crear clase para inicializar base de datos con datos de ejemplo' (1h), 'Añadir atributo para no usar el inicializador' (1h), and 'Llamar al inicializador desde el Web.config' (1h). Below the board, there are two additional task cards: 'Como empleado, desearía visualizar un listado de voluntarios' and 'Como empleado deseo poder visualizar y modificar datos de un voluntario'. The top navigation bar includes 'GestorONG', 'Dashboards', 'Code', 'Work', 'Build & Release', and 'Test'. A search bar for work items is also present.

Ilustración 2. Board Sprint 6 Gestor ONG

1.5. Estructura del trabajo

La presente memoria se divide en diferentes apartados, los cuales se describirán brevemente a continuación:

En el primer capítulo “Introducción” se hace una introducción del proyecto, justificando las diferentes motivaciones presentes a la hora de realizar el proyecto y los objetivos tanto generales como específicos a conseguir.

Posteriormente, en el “Contexto y estudio preliminar” se analizan las necesidades que puede tener una ONGD, para luego analizar diferentes soluciones genéricas o desarrollados especialmente para estas organizaciones, comparándolas y obteniendo las conclusiones oportunas que servirán como base para plantear los requisitos funcionales y no funcionales del software, analizados en el siguiente punto “Identificación de requisitos”, planteando el “product backlog” y analizando el sistema mediante casos de uso y sus correspondientes diagramas UML.

En la siguiente sección “Descripción del proyecto” se detallan las aportaciones realizadas al proyecto, una descripción funcional de la aplicación desarrollada, la planificación general del proyecto y por último un detalle con la planificación de uno de los diferentes “sprints” de la fase de implementación del código.

Dentro del capítulo “Descripción técnica”, se expone la arquitectura general del sistema con la descripción de cada una de las capas (tier) presentes en el aplicativo, un prototipo gráfico de las vistas de la aplicación, el modelo de datos implementado, el diagrama de clases del modelo, los diferentes patrones (con ejemplos de uso) utilizados para estructurar el software, las tecnologías, herramientas, librerías y “frameworks” utilizadas en el desarrollo y los detalles de la implementación.

A continuación, en el apartado “Entre sus múltiples comandos se encuentra `git commit -m [mensaje]`”, que sirve para confirmar los cambios de los archivos junto con un mensaje dónde se describen los cambios realizados, y `git push`”, que es el utilizado para introducir dichos cambios en el repositorio.

Evaluación y pruebas”, se describen las diferentes técnicas y resultados de evaluación del software tanto con herramientas automáticas de validación como las pruebas realizadas con usuarios finales, y se exponen las métricas de usabilidad y accesibilidad obtenidas para la aplicación.

En el capítulo “Conclusiones y trabajo futuro” se exponen las diferentes conclusiones obtenidas tras el desarrollo del software y se enumeran las diferentes necesidades u optimizaciones a realizar en posteriores fases de desarrollo del software a partir del “feedback” recibido por parte de los usuarios que han utilizado la aplicación web.

Por último, en las últimas secciones se muestra la bibliografía consultada a la hora de acometer el presente proyecto y un listado con los acrónimos utilizados y su definición.

2. Contexto y estudio preliminar

A continuación, se analizan las necesidades de gestión de información que puede tener una ONGD y se comparan diferentes soluciones existentes genéricas o específicas para estas organizaciones, que puedan ser de interés y sirvan como base para plantear los requisitos funcionales y no funcionales del aplicativo.

2.1. Contexto del Trabajo

La presente memoria contiene toda la documentación acerca del proyecto software que se ha desarrollado como trabajo de fin de grado (TFG) del Curso de Adaptación al Grado en Ingeniería Informática de la Universidad Internacional de La Rioja (UNIR).

El autor del documento, a la hora de elegir el proyecto a realizar, se decantó por implementar un software que diera respuesta a una necesidad existente dentro del mundo de las organizaciones sin ánimo de lucro para el desarrollo (ONGD).

Según se define en la Ley 23/1998 de Cooperación Internacional para el Desarrollo, una ONGD es una organización privada, legalmente constituida en España, sin ánimo de lucro, con estatutos que indiquen que se dedican a la cooperación internacional para el desarrollo y que tienen una estructura suficiente para trabajar en este campo. (Boletín Oficial del Estado (BOE), 1998)

Otra definición que se menciona en el libro “Capital social y organizaciones no lucrativas en España” (Bellostas Pérez-Grueso, Marcuello Servós, Moneva Abadía, & Marcuello Servós, 2007) es la realizada por Zavala en el año 1994 “organizaciones de carácter social, independientes y autónomas, jurídicamente fundadas y que actúan sin finalidad de lucro. Su acción se orienta hacia la Cooperación al desarrollo y hacia la búsqueda de acuerdos de ayudas entre gobiernos con el objetivo de provocar la solidaridad y promover el desarrollo en los pueblos y sociedades del Tercer Mundo. Esta acción se complementa con las actividades de sensibilización y educación para el desarrollo de nuestra sociedad en torno a las realidades de los países del Tercer Mundo y la interdependencia Norte-Sur, y con las actividades de “lobby” o presión política ante los gobiernos y los organismos” (Zavala, 1994)

Las diferentes necesidades que puede tener una ONGD son las siguientes:

- ✓ Gestionar de manera centralizada la información de los datos que se utilizan en su funcionamiento diario o los procesos que lleva a cabo como la gestión de los datos de voluntarios y colaboradores.
- ✓ Normalmente, estas organizaciones gestionan datos de carácter personal, lo que obliga a implantar un sistema robusto con una autenticación de usuarios eficaz y con

sistemas de copias de seguridad, tal como se menciona en el artículo 9 de la Ley de Protección de Datos de Carácter Personal (LOPD) publicada en el BOE: “El responsable del fichero, y, en su caso, el encargado del tratamiento deberán adoptar las medidas de índole técnica y organizativas necesarias que garanticen la seguridad de los datos de carácter personal y eviten su alteración, pérdida, tratamiento o acceso no autorizado, habida cuenta del estado de la tecnología, la naturaleza de los datos almacenados y los riesgos a que están expuestos, ya provengan de la acción humana o del medio físico o natural.” (Jefatura del Estado, 1999).

- ✓ Las ONGD pequeñas cuentan con pocos empleados (muchas veces a tiempo parcial) y se apoyan fundamentalmente en el trabajo de los voluntarios, lo que hace necesario optimizar los procesos automatizando tareas como la gestión de remesas bancarias de las donaciones o la emisión de certificados de donaciones para el IRPF.
- ✓ Generar las remesas bancarias correspondientes a las donaciones mediante la creación de ficheros XML con los datos de las donaciones en formato SEPA¹.

Como demuestra el Informe global sobre tecnología en línea de ONG 2017 (ONG, 2017), estas organizaciones a nivel mundial suelen ser pequeñas (el 56,1 %), están basadas en donaciones de colaboradores y el esfuerzo de los voluntarios (el 25% de estas organizaciones depende exclusivamente de ellos), no teniendo por lo general una persona o departamento de sistemas en plantilla. Además, estas fundaciones u organismos no se pueden permitir un gasto excesivo en licencias de software ni en equipamiento informático (“*hardware*”), decantándose por lo general por soluciones de código abierto o pequeñas aplicaciones personalizadas a sus necesidades básicas.

Aunque actualmente ya existen proyectos cuyo objetivo principal es acercar el uso de las (TIC's) a estas organizaciones mediante programas de donaciones como el Proyecto Dono² con socios como Adobe™, Autodesk™, Bitdefender™, Cisco™, Google™, Microsoft™, Symantec™ o Vodafone™, Google para Organizaciones sin Ánimo de Lucro³ o gestores de contenido web como Wordpress, Joomla o incluso Drupal para crear el sitio web de estas

¹ SEPA: Zona en la que ciudadanos, empresas y otros agentes económicos pueden hacer y recibir pagos en euros http://www.sepaesp.es/sepa/es/secciones/sobre-sepa/Sobre_SEPA.html

² El Proyecto Dono asiste a organizaciones no gubernamentales y entidades caritativas, así como a bibliotecas públicas, ofreciéndoles acceso al programa de donaciones de software y hardware de nuestros socios donantes. Apoyamos a estas organizaciones a mantener sus presupuestos en tecnología y a realizar sus pedidos tecnológicos. (Ilunion Tecnología y Accesibilidad, 2008) <https://dono.discapnet.es>

³ El programa Google para organizaciones sin ánimo de lucro, concede acceso a productos premium de Google que se ofrecen de forma gratuita a este tipo de organizaciones como, por ejemplo, Google Apps. (Google, s.f.) <https://www.google.es/nonprofits/products/>

organizaciones, la mayor parte de los procesos diarios que lleva a cabo la organización se realizan manualmente en herramientas de productividad como Microsoft Office™ o, en casos minoritarios, con una base de datos Access.

Como se analizará más adelante, al estudiar el estado del arte de aplicaciones informáticas de gestión, estos aplicativos son extremadamente complejos o de un coste elevado, por el que se decide apostar por desarrollar un software sencillo que cubra los requerimientos de gestión básicos de una organización sin ánimo de lucro.

Como voluntario de la Fundación Spínola Solidaria⁴ y administrador de sistemas informáticos de la misma, el creador del presente trabajo, ha desarrollado un sistema que, de una manera ágil y sencilla, de respuesta a la solicitud del Patronato de dicha fundación para que se implemente un sistema web que permita gestionar una base de datos de los voluntarios, colaboradores y donativos de la organización, automatizar los trámites a realizar con las entidades bancarias y las administraciones, así como recopilar datos provenientes de los diferentes sistemas de información ya existentes.

Por último, el creador del presente trabajo de fin de grado utilizará las herramientas tecnológicas (que se detallarán a posteriori) para ampliar su formación y demostrar una sólida experiencia en las mismas con el fin de orientar su carrera profesional hacia ese ámbito.

2.2. Soluciones existentes

Aunque no hay muchas soluciones aplicables para dar respuesta a la necesidad existente, en los últimos años, han surgido algunas iniciativas para intentar crear un sistema que aporte las funcionalidades necesarias para este tipo de organizaciones.

El presente estudio del estado del arte se centrará en analizar las distintas soluciones orientadas a organizaciones sin ánimo de lucro o en soluciones genéricas que puedan ser adaptadas:

2.2.1. Soluciones específicas para organizaciones sin ánimo de lucro

A continuación, se analizarán distintas soluciones existentes específicas para dar respuesta a las necesidades generales de una organización sin ánimo de lucro, las cuales han servido

⁴ Fundación sin ánimo de lucro para el desarrollo <https://www.spinolasolidaria.org>

como base a la hora de implementar las diferentes funcionalidades del software desarrollado.

2.2.1.1. CiviCRM

CiviCRM es un CRM “open source” construido por una comunidad de colaboradores y partidarios, coordinado por el “core-team”. Es un software basado en web usado por diversas organizaciones, particularmente ONG. (CIVICRM LLC, 2005)

CiviCRM está pensado especialmente para organizaciones sin ánimo de lucro y provee una plataforma totalmente integrada para dar respuesta a las necesidades básicas de estas organizaciones: “Más que cualquier otra cosa, lo que diferencia a CiviCRM de otros programas competidores es que está diseñado específicamente para satisfacer las necesidades de las organizaciones sin fines de lucro, y proporciona una plataforma bien integrada que aborda todas sus necesidades básicas”. (Murray & P. Shaughnessy, 2011)

Entre sus principales características, destaca que puede ser instalado en el propio servidor de la organización, o ser integrado en CMS como Wordpress, Joomla o Drupal; gestiona todos los datos de los contactos en un único lugar, es totalmente configurable gracias a su licencia “GNU Affero General Public License versión 3 (GNU AGPL v3), permite gestionar miembros y la comunicación con éstos, eventos y la gestión de donaciones. A nivel de informes, permite la generación de los mismos mediante sencillos formularios aplicando filtros y seleccionando los campos a incluir en los mismos.

En la Ilustración 3 se puede observar la vista de una pantalla del sitio de demostración de CIVICRM para Drupal mostrando los datos de los contactos almacenados en el sistema.

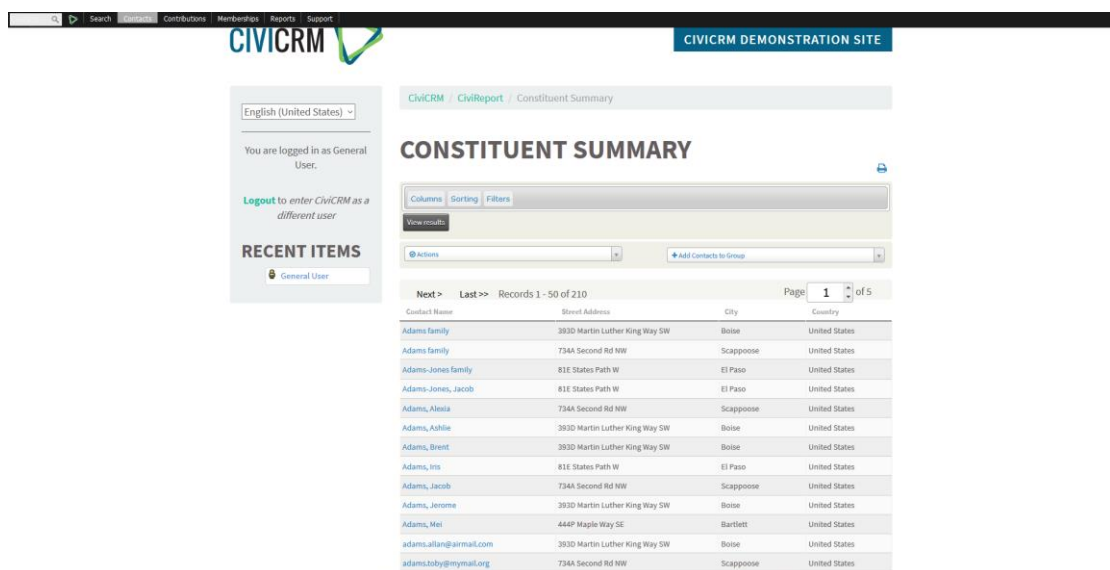


Ilustración 3. Ejemplo de un listado de contactos de CiviCRM

A nivel técnico, está desarrollado con PHP, Javascript (jQuery), HTML y CSS, mientras que almacena los datos en una base de datos MySQL⁵. CiviCRM dispone de una API en PHP para interactuar con el núcleo del sistema desde sistemas externos que está bien documentada.

A nivel de costes, el software es gratuito, por lo que se puede usar sin límites, aunque existen opciones para buscar un proveedor especializado que instale y mantenga el sistema.

Sin embargo, entre sus inconvenientes se encuentra que la curva de aprendizaje tanto a nivel de usuario como de desarrollador es alta, depende de otros sistemas al estar integrado en Drupal, Wordpress o Joomla (de hecho, en la Fundación Spínola Solidaria se está migrando el sitio web de Joomla a Wordpress, por lo que en el caso de haber tenido CiviCRM, se hubieran tenido que mantener ambos sistemas, puesto que CiviCRM no permite la migración de datos de manera sencilla).

2.2.1.2. GONG

GONG es una aplicación Web construida en software libre y cuyo objetivo es ayudar a las ONGD a la gestión de proyectos de cooperación al desarrollo. (semilla de SOFTWARE LIBRE, 2017) con licencia con arreglo a la EUPL⁶.

Una de las principales fortalezas de este proyecto es que está apoyado por instituciones como Cenatic, la Agencia Española de Cooperación Internacional para el Desarrollo (AECID) y Red.es

GONG permite una gestión integral de la información de la organización desde el punto de vista de los proyectos y de los agentes (delegaciones y unidades operativas) e importar y exportar información hacia y desde el sistema. En la Ilustración 4, se muestra una captura de pantalla de listado que ofrece GONG para visualizar los socios, así como un detalle para cada uno de ellos. Se ha podido comprobar que el diseño de la interfaz es limpio y elegante, y su uso resulta sencillo y muy intuitivo.:

⁵ MySQL es un sistema gestor de base de datos propiedad de Oracle. <https://www.mysql.com/>

⁶ European Union Public License

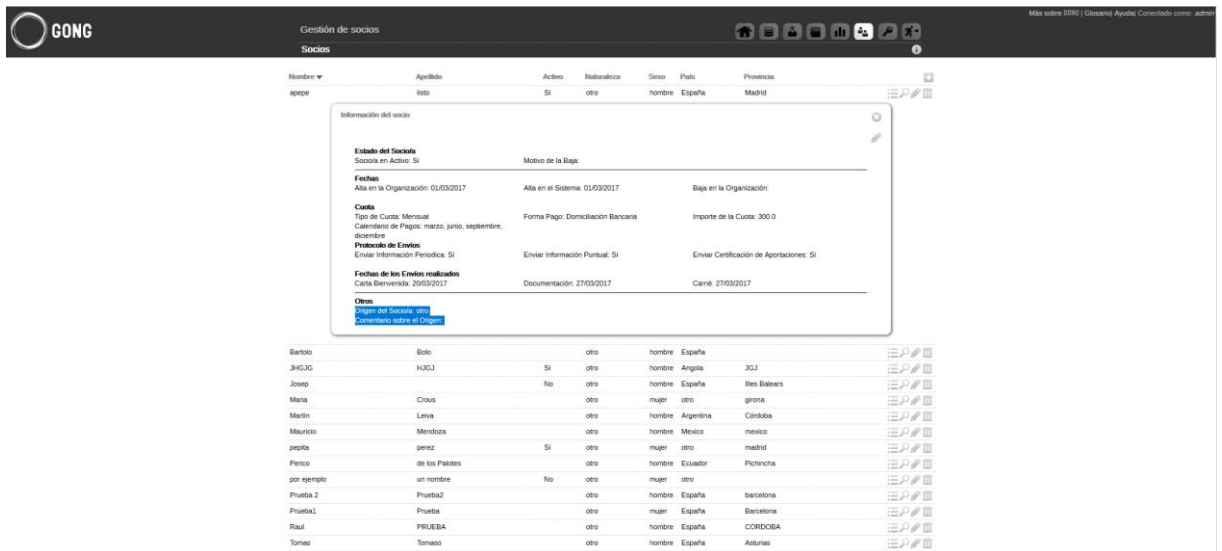


Ilustración 4. Listado de socios y el detalle de uno de ellos en GONG.

GONG puede ser instalado en el propio servidor de la organización si se dispone de personal técnico en la organización o mediante una instalación automática en la Nube.

Se trata de un proyecto desarrollado en “Ruby On Rails” requiriendo una base de datos MySQL (también se puede usar SQLite o Postgre-SQL) y un servidor Apache 2.x para su correcta instalación y funcionamiento.

Según se comenta en el libro de introducción a este lenguaje, “Ruby on Rails” es un “*framework*” de desarrollo basado en el lenguaje de programación “Ruby” y está compuesto por una serie de “*gems*” (gemas) que se han añadido por la comunidad de “Ruby” al “*core*” de este lenguaje: “*Ruby is a programming language. Rails is a development framework. Rails is software code written in the Ruby language. It is a library or collection of gems that we add to the core Ruby language.*” (Kehoe, 2016)

No dispone de una API integrada y los inconvenientes del sistema son su dificultad en la instalación ya que es necesario realizarla por comandos y requiere diversas configuraciones a nivel de servidor para que funcione correctamente. Además, viendo el repositorio dónde se encuentra el código del proyecto, no parece que se lancen actualizaciones de seguridad o mejoras habitualmente.

2.2.1.3. SinergiaCRM

“SinergiaCRM es una solución CRM estándar para el Tercer Sector desde el Tercer Sector” (Asociación SinergiaTIC, s.f.). Se trata de un software de código abierto desarrollado por

SinergiaTIC sobre la base del conocido “SugarCRM Community Edition⁷” liberado bajo licencia AGPL v3.

Entre sus múltiples funcionalidades destaca la posibilidad de gestionar los datos de voluntarios y colaboradores, la gestión de eventos, campañas y proyectos, la generación de informes exhaustivos y permite también generar las remesas bancarias en formato cuaderno 19 / SEPA.

En la Ilustración 5, se puede observar el detalle de un usuario almacenado en el sistema, con una subdivisión en paneles en función del tipo de información (detalles del módulo personas, subpanel de formas de pago y subpanel de histórico de pagos).

Panel de la vista de detalle del módulo Personas

Nombre:	María	Apellidos:	Fernández Alonso
Tipo relación:	<input checked="" type="radio"/> Socio <input type="radio"/> Voluntario	Código:	
Tipo Identificación:	NIF	Nº Identificación:	12345678Z
Fecha Nacimiento:	19/05/1964	Edad:	51
Idioma:	Gallego	Sexo:	Mujer
Teléfono Fijo:	912345678	Móvil:	654789654
Teléfono Alternativo:		Correo electrónico:	maria.fernandez@gmail.com (Principal)
Comentarios:	Asignado a: SinergiaCRM		

Subpanel de Relación Personas en el módulo Personas

Nombre (autom.)	Personas	Tipo relación	Fecha Alta	Fecha de baja	Proyectos	Rol en el proyecto
María Fernández - socio	María Fernández Alonso	Socio	08/07/2014			
María Fernández - voluntario	María Fernández Alonso	Voluntario	17/12/2012		taller de lectura	

Subpanel de Formas de Pago en el módulo Personas

Nombre (autom.)	Tipo de pago	Medio de pago	Periodicidad	Importe	Fecha primer pago	Fecha de baja
María Fernández - cuota socio	Cuota	Domiciliación	Mensual	€50,00	28/07/2015	

Subpanel de Pagos en el módulo Personas

Nombre	Estado	Forma de Pago	Fecha pago	Importe	Última Modificación
María Fernández - cuota socio - 2015-07-28	No Remesada	María Fernández - cuota socio	28/07/2015	€50,00	28/07/2015 18:45

Ilustración 5. Ejemplo del perfil de un usuario en Sinergia CRM

Uno de los puntos a considerar es que sólo puede ser alojada en la nube por lo que los datos no se encuentran en los servidores propios de la organización. En cambio, la organización no debe preocuparse de mantener actualizado el sistema o de realizar la instalación.

⁷ Es un software “open-source” categorizado como CRM para gestionar las relaciones de una organización con sus clientes. <https://www.sugarcrm.com/>

Al basarse en “SugarCRM”, está desarrollado en PHP, por lo que requiere un servidor Apache 2.x o IIS y el sistema gestor de base de datos suele ser MySQL, aunque también admite Microsoft SQL Server.

Aunque “SugarCRM” sí dispone de herramientas de desarrollo, SinergiaCRM al estar alojado en un servidor en la nube ajeno a la organización, no permite la personalización propia o mejora de nuevas funcionalidades, aunque en su página web aseguran que, con el pago de la cuota anual requerida, se cubre el desarrollo de las nuevas funcionalidades.

Aparte del inconveniente que supone el hecho de no tener control del sistema, la implantación de este software requiere el pago de una cuota de alta (desde 200€) y una cuota anual (desde 1.200€), lo que supone un gran inconveniente para organizaciones sin ánimo de lucro de pequeño tamaño que, en algunos casos, no llegan a los 500€ de presupuesto anual en TIC's.

2.2.2. Soluciones genéricas que pueden ser adaptadas

2.2.2.1. Microsoft Dynamics CRM

Microsoft Dynamics CRM es un software propietario desarrollado por Microsoft para gestionar las relaciones de una organización con sus clientes. (en el caso de una organización sin ánimo de lucro podría adaptarse para ser voluntarios y colaboradores).

Entre sus muchas funcionalidades, el software permite acceder a un historial de acciones (llamadas, visitas, cartas, e-mails), administrar toda la información correspondiente a los usuarios, gestionar las tareas diarias y es posible realizar una integración sencilla con sistemas de “*business intelligence*”.

Una de sus múltiples ventajas es la integración con sistemas Windows debido a que en la mayoría de estas organizaciones se utiliza un entorno con este sistema operativo y la integración con herramientas como Microsoft Office o Skype es muy sencilla.

En contra, se tienen dos factores como la necesidad de adquirir licencias de uso por usuario o por dispositivos (aunque existen programas de donación de licencias a un precio reducido a través del Proyecto Dono); y que su implantación en una organización requiere realizar un proyecto con un partner de Microsoft ya que se requiere de personal muy especializado para su instalación, configuración, personalización y mantenimiento del sistema.

En la Ilustración 6, se puede observar una interfaz gráfica de Microsoft Dynamics CRM adaptado a organizaciones sin ánimo de lucro por parte de una empresa (Organon

Consulting⁸) que ha desarrollado una implementación personalizada de este software a este tipo de organizaciones. Concretamente se puede observar el formulario de realización de una nueva donación.

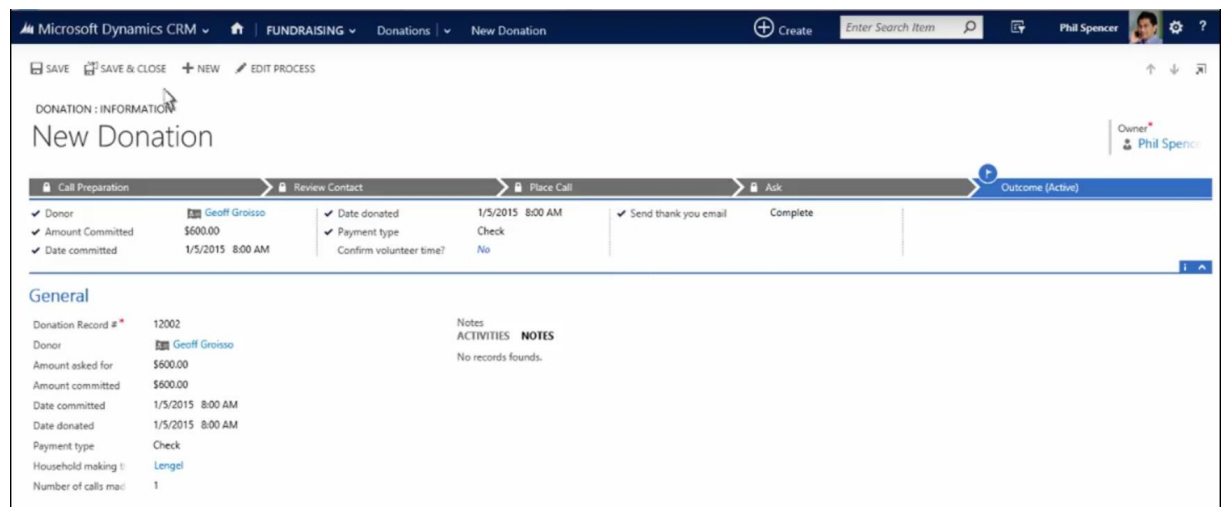


Ilustración 6. Ejemplo del formulario de donación de Microsoft Dynamics CRM.

Técnicamente, Microsoft Dynamics CRM necesita ser instalado en un servidor con Microsoft Windows Server que tenga habilitado Active Directory, tener habilitado el IIS, Microsoft SQL Server 2012 o superior como motor de base de datos (con el complemento de “Reporting Services” habilitado) y SharePoint Server si se desean administrar documentos.

A nivel de programación, se puede utilizar cualquier lenguaje de programación compatible con el “framework” .NET, pero, como ya se ha comentado antes, necesita de personal especializado en el software para que el programa funcione correctamente y se pueda adaptar y personalizar a la organización, lo que supone que el proyecto de implantación tenga un elevado coste debido a las tarifas de consultoría tecnológica de entre 500€ y 800€ diarios.

2.2.3. Conclusiones del análisis de soluciones existentes

En la Tabla 2 se exponen las diferentes soluciones detalladas anteriormente, haciendo hincapié en los puntos más destacados a la hora de elegir una de las soluciones estudiadas o desarrollar un nuevo software.


En cuanto a las soluciones específicas para ONG’s, todas son de código abierto y utilizan principalmente PHP o Ruby On Rails y MySQL como motor de base de datos, mientras que Microsoft Dynamics CRM es de código propietario y utiliza Microsoft SQL Server.

⁸ Para evitar reinventar la rueda, Organon Consulting ha desarrollado una personalización de Microsoft Dynamics CRM para ONG. <http://www.organonconsulting.com.au/industry-crm/not-for-profit>

A nivel de funcionalidades, todas incluyen funcionalidades genéricas, pero otras deben ser desarrolladas específicamente para cumplir las necesidades funcionales de la Fundación Spínola Solidaria, lo que resulta un inconveniente puesto que algunas de ellas no disponen de API, la curva de aprendizaje es muy elevada o están alojadas en la nube y no es posible acceder al código fuente, como es el caso de SinergiaCRM.

Por último, SinergiaCRM y Microsoft Dynamics CRM no pueden ser implantadas en la organización puesto que requieren de una suscripción, pago recurrente de licencias o la implicación de un consultor especializado en el proyecto que la Fundación Spínola Solidaria no puede asumir, mientras que CiviCRM es descartada porque depende de otros sistemas como Wordpress, Joomla o Drupal. De hecho, CiviCRM fue evaluada para implantarse en la organización y se descartó porque iba a realizarse una migración de sistemas web del gestor de contenidos Joomla a Drupal en un futuro no muy lejano, por lo que era un problema a la hora de acometer la migración.

Tabla 2. Comparativa de las soluciones similares estudiadas

				
Tipo de software	Open Source / Web	Open Source / Web	Open Source / Web	Propietario / Web
Requisitos del sistema	Apache 2.x Wordpress, Joomla o Drupal	Apache 2.x	Apache 2.x / IIS	Microsoft Windows Server (AD) con IIS habilitado
Licencia	GNU AGPL v3	EUPL	GNU AGPL v3	-
Tecnologías	PHP, JS, JQuery, CSS	Ruby on Rails	PHP, JS, CSS	.NET
Base de Datos	MySQL	MySQL o SQLite o Postgre-SQL	MySQL / SQL Server	SQL Server 2012 o superior
API	✓	✗	✗	Si se desarrolla
Gestión de contactos	✓	✓	✓	✓
Configurable	✓	✓	✓	✓
Gestión de donaciones	✓	✓	✓	A medida
C19/SEPA	A medida	A medida	✓	A medida
Informes	✓	✓	✓	✓
Usabilidad	3/5	4/5	4/5	4/5
Curva de aprendizaje	Alta	Media	Media	Alta
Precio	Gratis	Gratis	Desde 200€ + 1.200€ (anuales)	364€ + 169€ / usuario

3. Identificación de requisitos

Este apartado tiene como propósito definir de forma clara y precisa las funcionalidades y restricciones que tendrá el sistema software que se desea desarrollar para una organización sin ánimo de lucro, partiendo de las necesidades identificadas con el cliente.

3.1. Alcance del proyecto

El trabajo a desarrollar consistirá en el análisis, diseño, implementación, pruebas y validación de un sistema basado en tecnologías web que ayude a los empleados de una pequeña organización sin ánimo de lucro a gestionar información referente a voluntarios, colaboradores y donativos, automatizar los trámites a realizar con las entidades bancarias como la generación de remesas bancarias o la creación automática de certificados de donaciones para la declaración de la renta.

Así mismo, con el fin de unificar todos los sistemas, se desarrollarán los servicios web necesarios para obtener información sobre las solicitudes de voluntarios y donaciones realizadas a través de la página web de la Fundación.

Por último, además de realizar las pruebas oportunas del código fuente implementado, se realizará una validación funcional de los requisitos del proyecto con los usuarios. Estos usuarios serán los que posteriormente utilizarán la aplicación web en su día a día.

3.2. Descripción de los usuarios del sistema

El sistema está pensado para ser utilizado por los empleados de la fundación (actualmente 3 personas) y tendrá que tener una interfaz gráfica intuitiva y sencilla para que no requiera de ningún tipo de conocimiento informático avanzado. Los usuarios que utilizarán el sistema tienen conocimientos básicos y medios de uso de sitios web y sistema operativo Windows.

Además, existirá un rol “administrador” que tendrá acceso a las mismas funcionalidades que los usuarios del rol “empleado”, además del módulo de administración y configuración. Este rol lo tendrán los usuarios con permisos de administrador de dominio de una red de dominio Windows (departamento de IT).

3.3. Product Backlog

Roger Pressman en su libro “Software Engineering A Practitioner’s Approach” define el “product backlog” como una lista priorizada de requerimientos o características del proyecto que aportan valor al negocio del cliente. Además, estos elementos PBI (“*product backlog items*”) pueden ser añadidos al “product backlog” en cualquier momento. (Pressman, 2010, pág. 83)

A continuación, se muestra el “product backlog” del proyecto de desarrollo del software de gestión para la ONGD “Spínola Solidaria, separando el mismo en diferentes tablas que corresponden a cada una de las características (“features”) del proyecto.

Estos “product backlog items” se corresponderán a su vez con los diferentes casos de uso a estudiar en los puntos siguientes:

Tabla 3. Product Backlog Items Feature Gestión de Voluntarios

Título	Esfuerzo	Prioridad	Descripción	Criterio de Aceptación
Como empleado quiero dar de alta un nuevo voluntario	20	1	Mediante un formulario con el menor número de campos para que el usuario lo rellene, se debe permitir la introducción de los siguientes datos de un voluntario: * Nombre * Apellidos * Dirección Postal * Código Postal * Localidad * Provincia * País * Teléfono 1 * Teléfono 2 (opcional) * E-mail * Fecha de Nacimiento (opcional) * Voluntariado (Internacional, Sede, Campañas y Sensibilización) * Perfil (Además de Voluntario, puede ser colaborador u otro tipo de usuario : profesor, religiosa, superiora, delegada, director). Además, se cargará desde una REST API un listado genérico de países en un desplegable en formato JSON.	Se deben visualizar los datos del nuevo voluntario en el listado de voluntarios.
Como empleado, desearía visualizar un listado de voluntarios	24	1	En forma de tabla, similar a la siguiente imagen, se deben poder visualizar los datos más representativos de un voluntario, junto con un botón detalles en la parte derecha de la pantalla que permita visualizar y modificar todos los detalles de un voluntario (esta funcionalidad, se definirá en otro "product backlog item"): Los datos más representativos a mostrar en la tabla serán: * Nombre * Apellidos * Dirección Postal * Código Postal * Localidad * Provincia * País * Teléfono 1 * Teléfono 2 (opcional) * E-mail * Fecha de Nacimiento (opcional) * Voluntariado (Internacional, Sede, Campañas y Sensibilización) * Perfil (Además de Voluntario, puede ser colaborador u otro tipo de usuario : profesor, religiosa, superiora, delegada, director)	Se debe visualizar un listado con los datos indicados y unos botones que permitan realizar búsquedas, filtros y modificar o añadir datos.
Como empleado deseo poder visualizar y modificar datos de un voluntario	12	1	Se debe presentar un formulario con los datos completos de un voluntario, con posibilidad de modificar algunos de los campos. En concreto: * Nombre * Apellidos * Dirección Postal * Código Postal * Localidad * Provincia * País * Teléfono 1 * Teléfono 2 (opcional) * E-mail * Fecha de Nacimiento (opcional) * Voluntariado (Internacional, Sede, Campañas y Sensibilización) * Perfil (Además de Voluntario, puede ser colaborador u otro tipo de usuario : profesor, religiosa, superiora, delegada, director)	Si al modificar uno o más datos del voluntario seleccionado, se guardan en la base de datos y aparecen en el listado o en este mismo formulario actualizados.
Como empleado quiero dar de baja a un voluntario	6	2	Para cumplir con la legislación vigente en cuestión de protección de datos, se debe permitir dar de baja a un voluntario eliminando todos sus datos de la base de datos. Esta baja se realizará desde el listado de voluntarios con una confirmación previa para evitar eliminaciones accidentales.	La funcionalidad estará completada si, una vez dado de baja el voluntario, éste ya no aparece en el listado.

Tabla 4. Product Backlog Items Feature Gestión de Colaboradores y Donaciones

Título	Esfuerzo	Prioridad	Descripción	Criterio de Aceptación
Como empleado, desearía dar de alta un colaborador	25	1	<ul style="list-style-type: none"> Se deberá permitir dar de alta a un nuevo colaborador. Los datos del colaborador son muy parecidos a los del voluntario, añadiendo o cambiando algún dato (de hecho, puede ser colaborador y voluntario a la vez). Adicionalmente, a una persona se le denomina colaborador siempre que ésta haga una donación, por lo que el formulario de alta de un colaborador, deberá permitir realizar las siguientes operaciones: * Si los datos de la persona ya están en la base de datos, se deberán solicitar al usuario únicamente aquellos datos nuevos (CIF/NIF y Cuenta Bancaria) y se añadirá en la relación personas-perfiles al usuario con el rol "voluntario" (se deberá permitir buscar o filtrar previamente). Si no están en la base de datos, se activarán o se mostrarán todos los campos para rellenarlos. En ambos casos, habrá un segundo apartado que permitirá introducir los datos de la donación. <p>Campos de un Colaborador: Nombre · Apellidos · Dirección Postal · Código Postal · Localidad · Provincia · País · Teléfono 1 · Teléfono 2 (opcional) · E-mail · Fecha de Nacimiento (opcional) · Perfil (Además de Voluntario, puede ser colaborador u otro tipo de usuario: profesor, religiosa, superiora, delegada, director) CIF/NIF Cuenta Bancaria Campos de las donaciones: - Colaborador. - Importe. - Periodicidad.</p>	Los nuevos datos del colaborador deberán aparecer en el listado de colaboradores y en la vista detalle. Para completar esta funcionalidad, también deberán aparecer las donaciones del mismo.
Como empleado me gustaría ver un listado de colaboradores	24	1	En forma de tabla, similar a la siguiente imagen, se deben poder visualizar los datos más representativos de un colaborador, junto con un botón detalles en la parte derecha de la pantalla que permita visualizar y modificar todos los detalles de un colaborador (esta funcionalidad, se definirá en otro "product backlog item"): Los datos más representativos a mostrar en la tabla serán: * Nombre * Apellidos * Dirección Postal * Código Postal * Localidad * Teléfono 1 * E-mail * Nombre * CIF/NIF * Cuenta Bancaria	Se debe visualizar un listado con los datos indicados y unos botones que permitan realizar búsquedas, filtros y modificar o añadir datos.
Como empleado quiero visualizar y modificar los datos de un colaborador	18	1	Se debe presentar un formulario con los datos completos de un colaborador, con posibilidad de modificar algunos de los campos. En concreto: Nombre · Apellidos · Dirección Postal · Código Postal · Localidad · Provincia · País · Teléfono 1 · Teléfono 2 (opcional) · E-mail · Fecha de Nacimiento (opcional) · Perfil (Además de Voluntario, puede ser colaborador u otro tipo de usuario: profesor, religiosa, superiora, delegada, director) CIF/NIF Cuenta Bancaria	Correcto si al modificar uno o más datos del colaborador seleccionado, se guardan en la base de datos y aparecen en el listado o en este mismo formulario actualizados.
Como empleado debería poder dar de baja a un colaborador	6	2	Para cumplir con la legislación vigente en cuestión de protección de datos, se debe permitir dar de baja a un colaborador eliminando todos sus datos de la base de datos. Esta baja se realizará desde el listado de colaboradores con una confirmación previa para evitar eliminaciones accidentales. Además, se deben dar de bajas todas las donaciones existentes de este colaborador previamente a la eliminación.	La funcionalidad estará completada si, una vez dado de baja el colaborador, éste ya no aparece en el listado.

Como empleado quiero crear una nueva donación	12	1	<ul style="list-style-type: none"> Se presentará un formulario para permitir crear una nueva donación. Una donación implica que la realice una persona, por lo que previamente se tendrá que seleccionar a la persona que realice la donación, bien seleccionando a la misma desde el listado de colaboradores, seleccionando una persona ya existe en el sistema (porque sea voluntario ya) o permitiendo añadir en el mismo formulario de crear donación a un colaborador nuevo (enlace al formulario de alta de colaborador). Los datos característicos de una donación son: - Persona que realiza la donación. - Cantidad. - Periodicidad. 	Cuando se realice el alta una nueva donación, ésta debe aparecer en el detalle del colaborador y en la lista de donaciones.
Como empleado me gustaría listar las donaciones de los colaboradores	24	1	Se debe mostrar un listado con posibilidad de realizar filtros y búsquedas de las donaciones existentes en el sistema. Los datos a mostrar serán: - Nombre y apellidos del colaborador (enlazable a los detalles de ese colaborador). - DNI/CIF del colaborador. - Cantidad de la donación. - Periodicidad.	Se tendrá que visualizar un listado con las donaciones y si se realizan búsquedas o filtros, aparecer los resultados correspondientes.
Como empleado quiero modificar los datos de una donación	6	1	Se debe permitir modificar los datos de una donación. Para ello, sólo será posible modificar los siguientes datos: - Cantidad de la donación. - Periodicidad. Para modificar otros datos, será necesario dar de baja una donación y crear una nueva.	En el listado de donaciones y en los detalles del colaborador, deben aparecer los datos modificados.
Como empleado quiero emitir los certificados para la declaración de la renta	20	1	Se debe generar un documento en formato PDF por cada colaborador con la cantidad total de dinero que ha donado a la fundación en el ejercicio anterior. Este documento servirá al colaborador para realizar la declaración de la renta. El documento a generar será similar al siguiente y se deberán almacenar en una carpeta del servidor dónde esté alojado el sistema gestor de ONG para que el empleado pueda acceder a los documentos generados:	Se tendrá un documento PDF por cada colaborador en la carpeta de exportación del sistema con el formato mostrado en el apartado "Description".
Como empleado y tarea programada me gustaría emitir las remesas bancarias	48	1	Se debe generar un documento en formato XML con los datos necesarios y en el formato concreto para importarlo en el software del banco y poder generar las remesas bancarias. Cada mes, se deberán generar las remesas de las donaciones mensuales, cada tres meses las trimestrales y cada año las donaciones anuales. El sistema tendrá planificadas unas fechas para la realización de esta tarea, aunque el empleado podrá hacerlo manualmente.	Se tendrá en la carpeta de exportación, el fichero en formato XML listo para importar al software del banco.

Tabla 5. Product Backlog Items Feature Gestión de Sedes y Delegaciones

Título	Esfuerzo	Prioridad	Descripción	Criterio de Aceptación
Como empleado quiero crear una nueva sede o delegación	20	1	El usuario con rol empleado, podrá dar de alta una nueva sede o delegación para permitir relacionar a los voluntarios con las sedes a las que pertenecen o en las que colaboran. Los datos a almacenar deben ser: * Nombre de la delegación. * Dirección postal. * Localidad. * Código postal. * Provincia. * País. * Persona de contacto. * Email de contacto. * Teléfono de contacto.	En el listado de sedes y delegaciones y en la tabla de la base de datos correspondiente, debe aparecer la nueva sede o delegación.
Como empleado me gustaría visualizar todas las sedes y delegaciones	10	1	El usuario con rol empleado, podrá visualizar todas las sedes y delegaciones en formato tabla. Los datos más representativos a mostrar serán: * Nombre de la delegación. * Localidad. * País. * Persona de contacto. * Email de contacto. * Teléfono de contacto. El resto de campos se visualizarán en la vista detalle.	Se debe visualizar un listado con los datos indicados y unos botones que permitan realizar búsquedas, filtros y modificar o añadir datos.
Como empleado quiero visualizar y modificar los datos de una delegación	12	1	El usuario con rol empleado, deberá visualizar y modificar los datos de una sede o delegación. Los datos que se mostrarán y se podrán visualizar serán: * Nombre de la delegación. * Dirección postal. * Localidad. * Código postal. * Provincia. * País. * Persona de contacto. * Email de contacto. * Teléfono de contacto.	Al pulsar sobre un registro del listado, se deben poder visualizar todos los datos descritos anteriormente en formato formulario con posibilidad de modificarse. Al modificarse, éstos deberán aparecer actualizados en la base de datos y en el listado.

Tabla 6. Product Backlog Items Feature Administración y Configuración

Título	Esfuerzo	Prioridad	Descripción	Criterio de Aceptación
Como administrador quisiera cambiar el logo del sistema	6	4	Se presentará un formulario con un campo "FILE" para subir la imagen con el nuevo logo, que se almacenará en una tabla de la base de datos y será el que se visualice en la sección superior de la ventana.	Al subir otra imagen, el logo del sistema cambiará y se visualizará en la sección de navegación superior.

Tabla 7. Product Backlog Items Feature Gestión LOPD

Título	Esfuerzo	Prioridad	Descripción	Criterio de Aceptación
Como empleado quiero registrar las entradas y salidas de información	20	4	Según la LOPD, se deben registrar mediante un formulario las entradas y salidas de información. Para las entradas se deberán almacenar: "entradas, el tipo de documento o soporte, la fecha y hora, el emisor, el número de documentos o soportes incluidos en el envío, el tipo de información que contienen, la forma de envío y la persona autorizada responsable de la recepción" Para las salidas de información: "el tipo de documento o soporte, la fecha y hora, el destinatario, el número de documentos o soportes incluidos en el envío, el tipo de información que contienen, la forma de envío y la persona autorizada responsable de la entrega"	El formulario debe permitir introducir los campos mencionados anteriormente y registrar los datos correctamente en la base de datos.
Como empleado quiero registrar una incidencia de seguridad de los datos	24	4	En el registro de incidencias que debe disponer la organización, se registrará cualquier incidente que afecte, o pueda afectar a la seguridad de los ficheros con datos personales. Se deberán tener en cuenta, entre otras, las siguientes incidencias: * Pérdida de información de algún fichero de datos de carácter personal. * Modificación de datos personales por personal no autorizado o desconocido. * Existencia de sistemas de información sin las debidas medidas de seguridad. * Los intentos de acceso no autorizados a ficheros de carácter personal. * El conocimiento por terceros de la clave de acceso al sistema. * El intento no autorizado de salida de un soporte. * La existencia de soportes sin inventariar y que contengan datos personales. * La destrucción total o parcial de un soporte que contenga datos de carácter personal. * La caída del sistema de seguridad informática, que posibilite el acceso a datos personales por personas no autorizadas. Los datos a registrar serán los que aparecen en la siguiente imagen: [¿Qué incidencias deben registrarse de cara a la LOPD?] (17 de mayo de 2013 http://www.lopdmanager.es/que-incidencias-deben-registrarse-de-cara-a-la-lopd/)	Se debe mostrar la incidencia registrar en el listado de incidencias del apartado LOPD.
Como empleado desearía generar informes para auditorías	35	4	Se debe permitir generar informes sobre la seguridad de los datos. Los datos a mostrar serán los siguientes: * Incidencias de seguridad de los datos: * Nº de incidencia * Fecha y Hora * Tipo de incidencia * Descripción * Efectos derivados * Medidas correctoras * Persona que comunica la incidencia * Persona que recibe la notificación * Entrada y salida de información (soportes): * Tipo de documento o soporte * Fecha y hora * Destinatario / Receptor * Número de documentos * Tipo de información que contienen los soportes * Forma de envío * Persona autorizada responsable de la entrega o recepción Se deberá permitir realizar un filtrado de datos por fecha y tipo de informe a realizar principalmente.	Se generará un informe filtrado por las fechas seleccionadas en formato PDF con la información requerida por el usuario.

3.4. Perspectiva del producto

El contexto de implantación del producto será el siguiente:

- ✓ Interfaz de usuario: Será un sitio web basado en una serie de menús de navegación, tablas y formularios.
- ✓ Interfaz hardware: Ratón, teclado y monitor.
- ✓ Interfaz de comunicación: Será necesaria una red de área local (LAN) para que los usuarios puedan acceder mediante interfaz web al sistema instalado en un servidor local con Microsoft Information Services (IIS) y, en casos puntuales, conexión al servidor web externo para importar datos desde los sitios web de la ONG.
- ✓ Operaciones: Se tendrán que realizar copias de seguridad diarias para salvar la información del sistema. Estos “backups” se realizarán directamente del sistema gestor de base de datos en dos ubicaciones diferentes.
- ✓ Requisitos de adaptación: El software se podría instalar en servidores con sistema operativo Microsoft Windows Server 2012 R2 o superior, independientemente de que sean servidores físicos o virtualizados.

3.5. Requisitos no funcionales

A continuación, se detallarán las restricciones que se deben tener en cuenta a la hora de desarrollar el sistema de gestión de una organización sin ánimo de lucro:

- ✓ Limitaciones “hardware”: No existen limitaciones de este tipo puesto que el sistema se ejecutará en un servidor con los recursos suficientes. En caso de ser necesario, se ampliarán estos recursos.
- ✓ Interfaces con otras aplicaciones: El sistema no se comunicará con ninguna aplicación de manera directa.
- ✓ Operaciones en paralelo: Habrá una única instancia de la aplicación corriendo sobre el servidor, pero se permitirán múltiples consultas desde el navegador web de los equipos de la ONG.
- ✓ Requisitos de lenguaje de alto nivel: Se utilizarán tecnologías Microsoft (C#.NET y ASP.NET) para el desarrollo del sistema. Se detallarán las tecnologías utilizadas más adelante.
- ✓ Requisitos de fiabilidad: Los datos deben guardarse correctamente, evitando la duplicidad de información. Habrá que asegurarse de que todas las operaciones de almacenamiento o carga de datos se han completado.
- ✓ Criticidad de la aplicación: Las consultas y operaciones que se realicen en el sistema, no deberán exceder el tiempo de 10 segundos.

3.6. Modelo de Dominio

El modelo de dominio es una representación visual estática del entorno real objeto del proyecto:

El propósito del modelo de análisis de dominio es aportar una descripción de la información requerida, funcional y de comportamiento de un sistema software. (Pressman, 2010, pág. 138)

Es decir, un diagrama con los objetos que existen (reales) relacionados con el proyecto que se va a acometer y las relaciones que hay entre ellos. Pero no son las clases del software (aunque algunas clases del Modelo del Dominio pueden terminar siéndolo).

Su objetivo es ayudar a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar el aplicativo a desarrollar.

En la Ilustración 7. Modelo de Dominio se pueden ver los diferentes módulos que componen el sistema a nivel de funcionalidad como son la gestión de voluntarios, que accede y gestiona toda la información referente a los voluntarios y las sedes o delegaciones a las que pertenecen; la gestión de colaboradores y donaciones, que permite crear, modificar o eliminar la información de los colaboradores y las donaciones que éstos realizan a la organización, así como generar los certificados de la renta y las remesas bancarias correspondientes a dichas donaciones.

Por otro lado, se encuentra el módulo los trámites para cumplir con la Ley Orgánica de Protección de Datos Personales (LOPD) implementando PBI como el registro de incidencias de seguridad de los datos o el registro de entradas y salidas de información de carácter personal de las instalaciones y sistemas de la organización.

Por último, también se puede ver el módulo de administración y configuración que hace referencia a los PBI que permiten cambiar el logo de la aplicación o configurar la información que se muestra en los desplegados que aparecen en los formularios de la aplicación desarrollada.

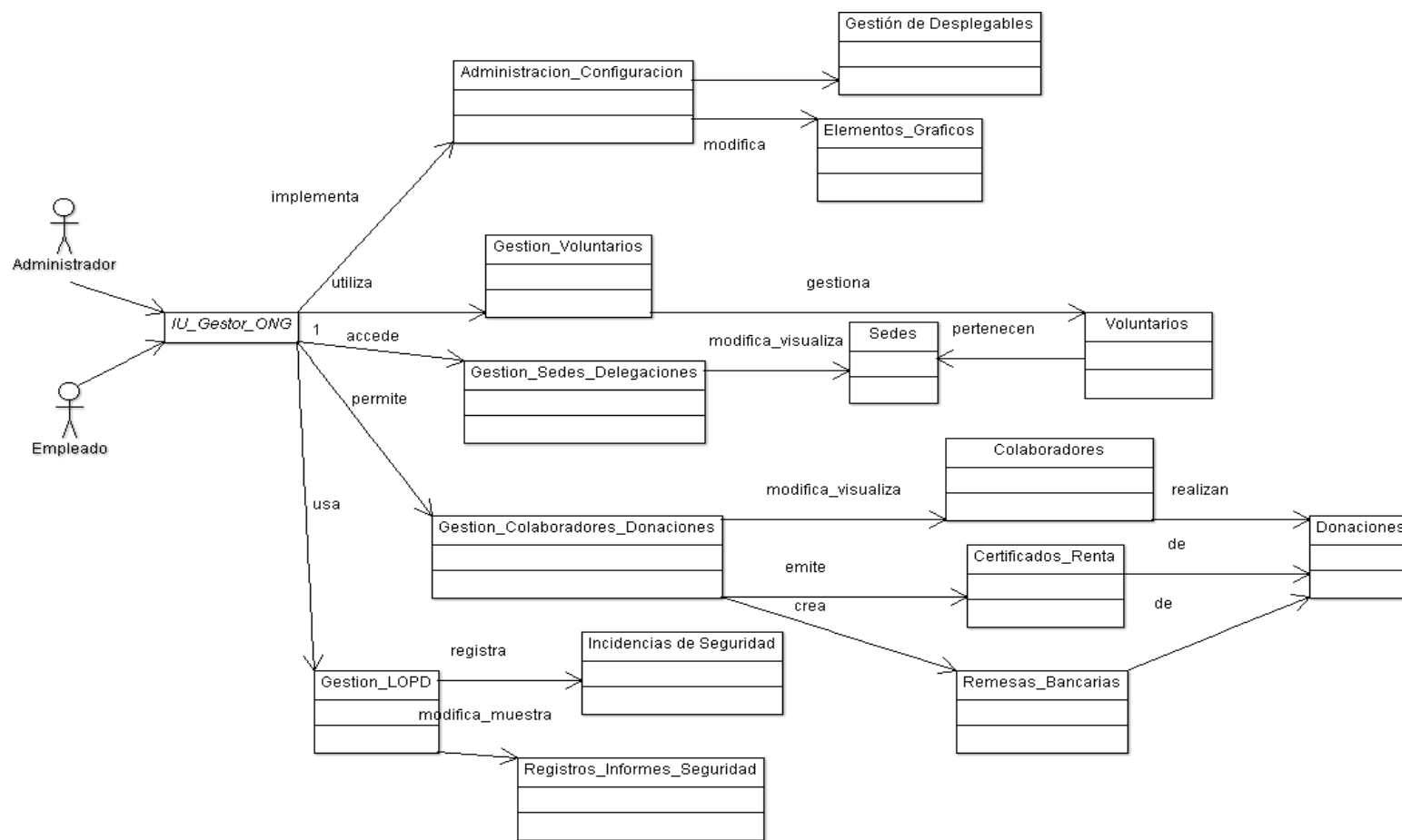


Ilustración 7. Modelo de Dominio

3.7. Casos de Uso

A continuación, se analizarán los diferentes requerimientos del sistema mediante diagramas UML, identificando los casos de uso correspondientes, es decir, mostrando las principales funcionalidades que debe tener la aplicación, los actores que tienen acceso a éstas y la conexión con sistemas externos.

Cada caso de uso se corresponde a un PBI según se menciona en el apartado Product Backlog.

3.7.1. Gestión de Voluntarios

En la Ilustración 8. se pueden ver los diferentes casos de uso que forman parte del módulo de voluntarios. En él, hay dos actores, el empleado de la fundación que puede interactuar con todas las funcionalidades y una tarea programada que únicamente importará las solicitudes de voluntariado recibidas desde la web. A la derecha de la imagen se muestran los sistemas externos con los que se comunicará el sistema para recibir o modificar datos vía servicios web.

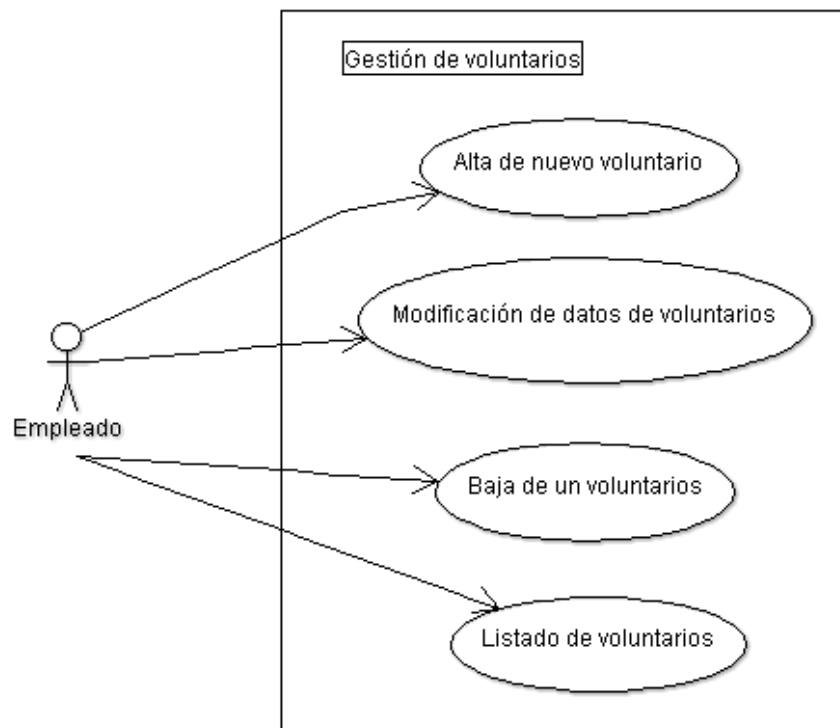


Ilustración 8. Diagrama de casos de uso para el módulo "voluntarios"

3.7.2. Gestión de sedes y delegaciones

En el módulo de gestión de sedes y delegaciones (Ilustración 9) se podrán realizar las funcionalidades típicas de un software de gestión: alta, modificación y listado.

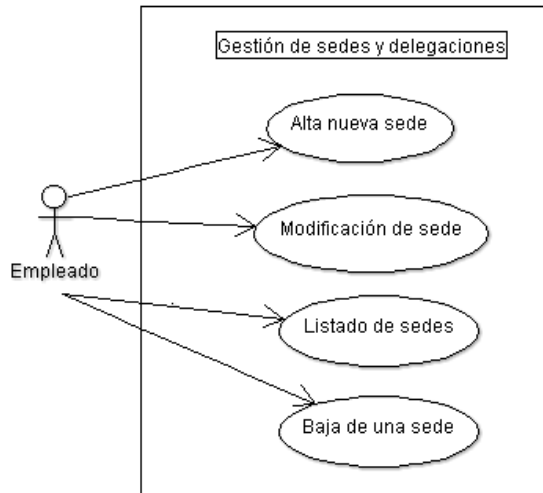


Ilustración 9. Diagrama de casos de uso para el módulo "sedes y delegaciones"

3.7.3. Gestión de colaboradores y donaciones

En la Ilustración 10, aparte de las funcionalidades de alta, modificación (y visualización) y listado con filtros de colaboradores y donaciones, se incluyen dos casos de uso adicionales como la emisión de certificados de la renta y la creación de las remesas bancarias SEPA.

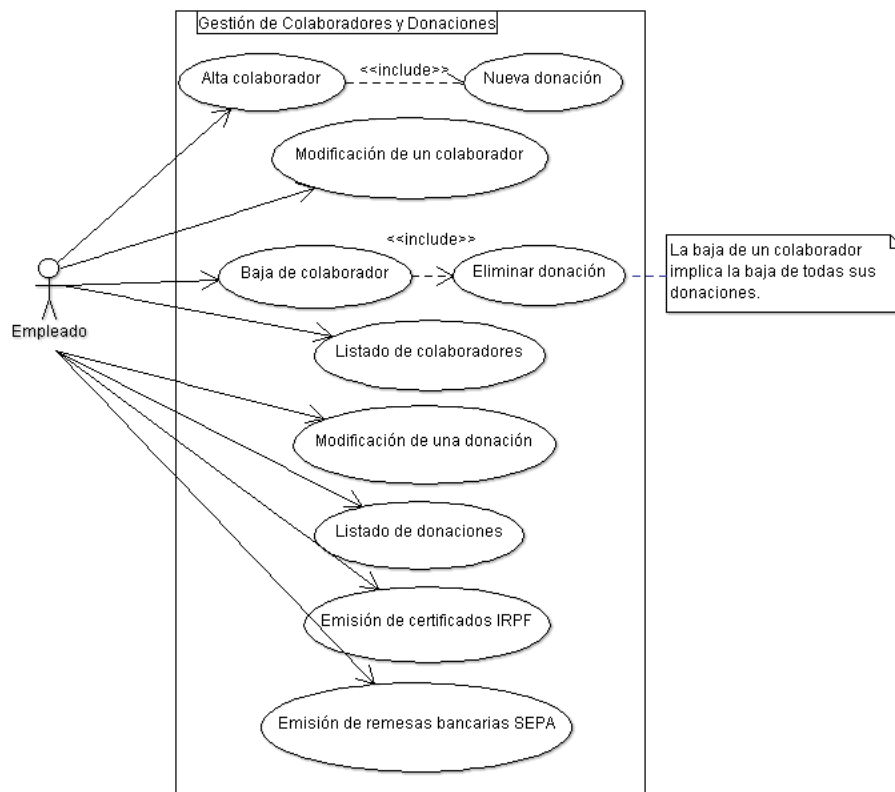


Ilustración 10. Diagrama de casos de uso del módulo "colaboradores y donaciones"

3.7.4. Gestión LOPD

En la Ilustración 11. se puede observar como los usuarios con rol “empleado”, podrán registrar cierta información relacionada con la normativa que exige la LOPD, dando lugar a tres casos de uso correspondientes al registro de los datos y su posterior tratamiento y generación de informes.

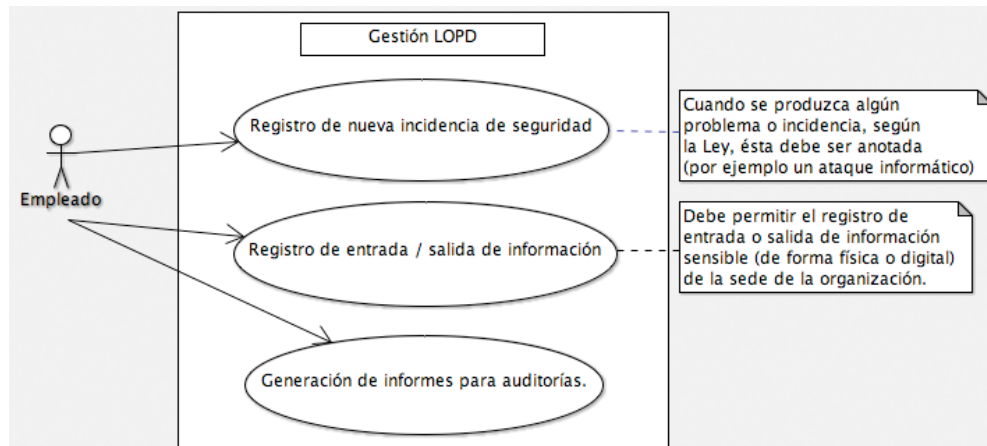


Ilustración 11. Diagrama de casos de uso para el módulo “LOPD”

3.7.5. Administración y configuración

En la Ilustración 12 se puede ver cómo un empleado podrá cambiar el logo de la aplicación, así como gestionar (crear, modificar o eliminar) toda la información referente a los desplegables presentes en los diferentes formularios de la aplicación como los perfiles o las periodicidades de las donaciones (mensual, trimestral, semestral, anual, etc.)

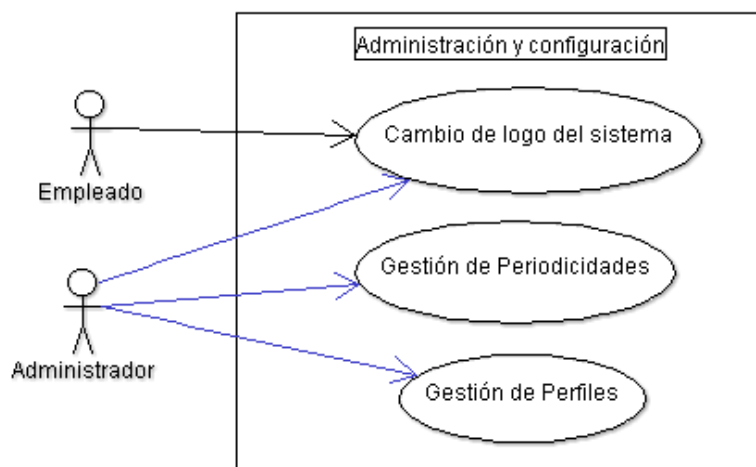


Ilustración 12. Diagrama de casos de uso del módulo “administración y configuración”

3.7.6. Gestión de usuarios

Respecto a la gestión de usuarios, no habrá ningún caso de uso asociado puesto que la gestión se realizará mediante la autenticación de “Active Directory”, que ya está implantado en un servidor de la Fundación y que está soportado en el “core” de ASP.NET MVC.

4. Descripción del proyecto

4.1. Aportación realizada

Se ha desarrollado una aplicación web intuitiva y sencilla en la que mediante formularios y listados es posible crear, modificar y consultar todos los datos referentes a los voluntarios, colaboradores, donaciones y registros de incidencias de seguridad, así como de entradas y salidas de información.

Para favorecer la usabilidad, todas las pantallas de la aplicación web tienen el mismo diseño gráfico compartiendo un “layout” común compuesto por una barra de navegación con el menú principal del sitio web (*header*), un cuerpo dónde se muestran los diferentes formularios y listados (*body*) y un *footer* formado por unos enlaces a otros sitios web de la organización como son el sitio web, el acceso a la plataforma de correo electrónico y un enlace a la plataforma de formación de la fundació. En este último contenedor también aparecen los logos que confirman que la aplicación web tiene validados correctamente el código HTML, CSS y Javascript y sigue las directivas del tercer nivel de accesibilidad web marcado por el W3C⁹, como se puede observar en la Ilustración 13.



Ilustración 13. Validación Usabilidad y Accesibilidad Web.

Respecto al cumplimiento de la LOPD, la aplicación es capaz de obtener informes filtrados por fecha sobre las incidencias de seguridad de los datos o salidas y entradas de información de los sistemas y archivos de la organización.

Todos estos formularios e informes comentados se podrán ver mediante imágenes ilustrativas en el siguiente apartado Descripción de la aplicación.

⁹ Logos de conformidad con las guías de accesibilidad del consorcio W3C <https://www.w3.org/WAI/WCAG2-Conformance>

4.2. Plan de trabajo

Respecto al plan de trabajo, como ya se ha mencionado en la sección Metodología de trabajo, se ha utilizado *SCRUM*, por lo que se ha dividido el desarrollo del proyecto en 13 “*sprints*” con una duración del mismo de exactamente dos semanas.

A continuación, se puede ver en la Ilustración 14 un gráfico con la distribución de tipos de elementos de trabajo, habiendo un total de 153 elementos distribuyéndose en 111 tareas, 24 PBI, 13 errores (“*bugs*”) solucionados y 5 “*features*” que corresponden a los 5 módulos presentes en el “*product backlog*”:

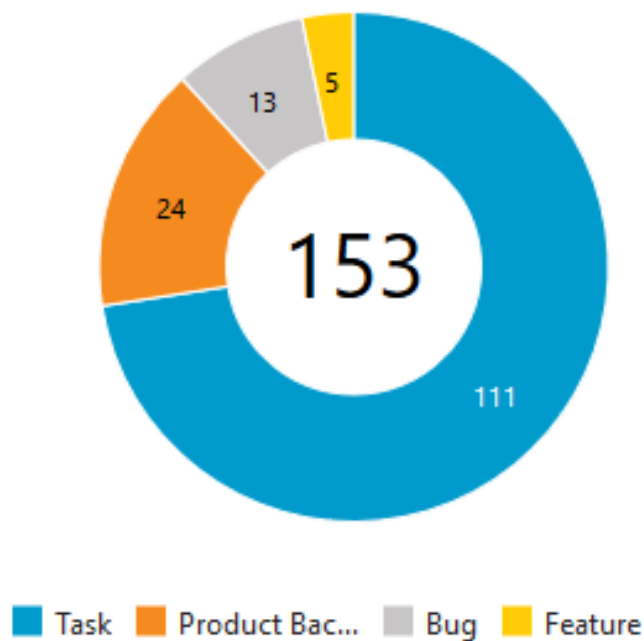


Ilustración 14. Gráfico con los diferentes elementos de trabajo del proyecto.

En la Ilustración 15, se puede ver la distribución de las tareas del proyecto en los diferentes “*sprints*” (parte izquierda de la imagen), mientras que en la parte derecha se muestra el diagrama de *Gantt* correspondiente a la planificación inicial que se hizo del proyecto. “Se define un diagrama de *Gantt* como la forma básica de un diagrama de barras que consiste en una lista de actividades en el lado izquierdo, cada una de las cuales tiene asociada una barra horizontal de longitud proporcional a su duración.” (González Marcos, Alba Elías, & Ordieres Meré, 2014, pág. 380)

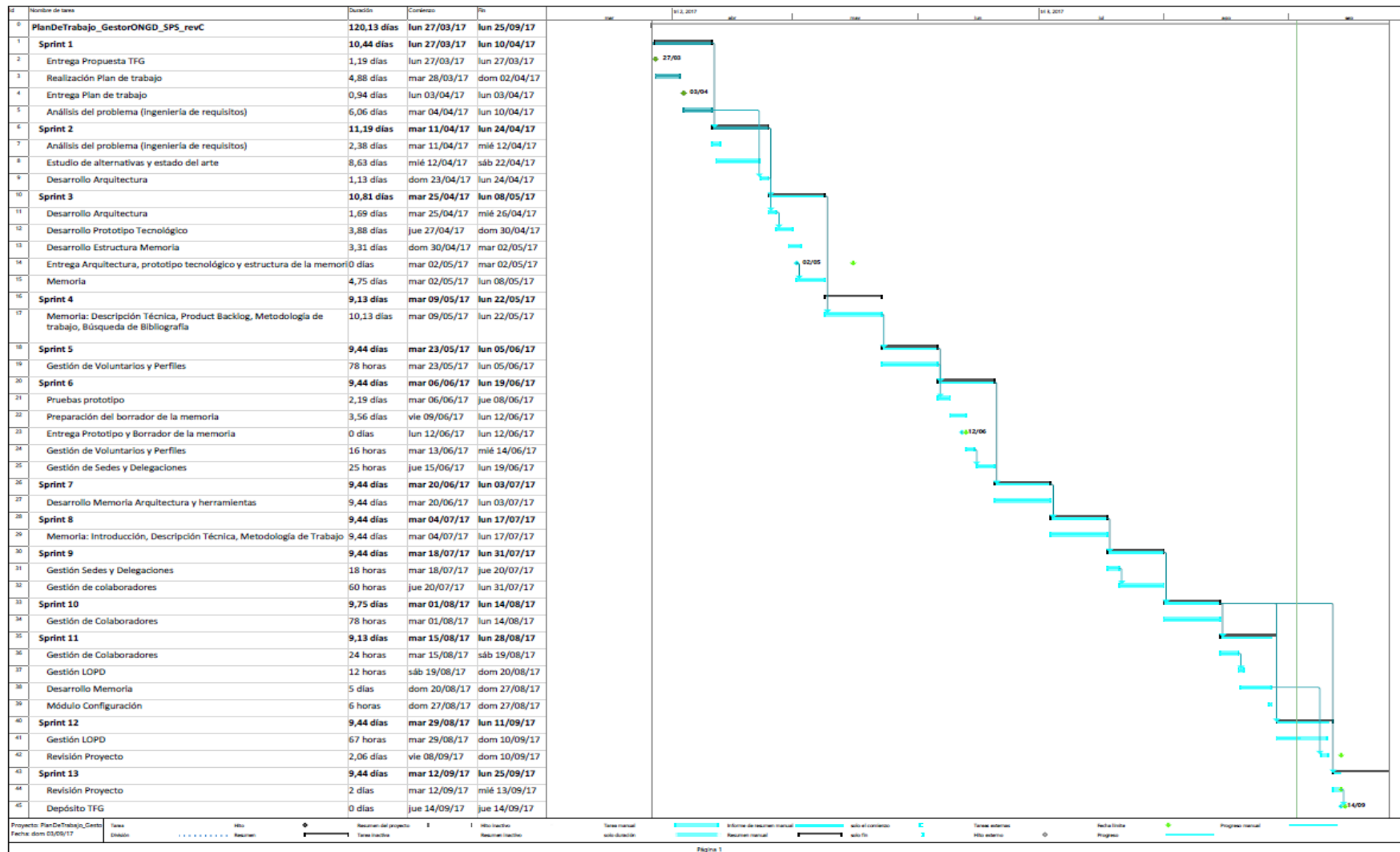


Ilustración 15. Diagrama de Gant del proyecto.

4.3. Descripción de la aplicación

Accediendo a la aplicación mediante su dirección web de la intranet, aparece la pantalla de inicio de la aplicación como se puede observar en la Ilustración 16, dónde además de tener el menú superior (siempre visible), aparece un menú compuesto por imágenes y botones (ambos elementos enlazan al listado del correspondiente módulo) de cada uno de los módulos desarrollados en la aplicación.



Ilustración 16. Página de inicio de la aplicación web.

El menú principal de la aplicación aparece a la derecha del logo y está compuesto por un enlace a cada uno de los “features” del “product backlog” o módulos que componen el sitio web. Como se puede observar en la Ilustración 17 , cuando se coloca el ratón sobre uno de estos elementos, se despliega un menú secundario formado por enlaces a otras funcionalidades del módulo, como en el caso del módulo de colaboradores:



Ilustración 17. Menú secundario al colocar el ratón sobre un elemento del menú principal.

Por este motivo, todos aquellos listados de datos tienen un buscador dinámico en tiempo real, se pueden filtrar las columnas susceptibles de ser filtradas como por código postal, comunidad autónoma o país (con búsqueda en tiempo real también) y ordenarlos. Así

mismo, aparecen un número de elementos por página para evitar que el usuario tenga que hacer "scroll" para seguir viendo más elementos, teniendo una paginación disponible para ver el resto de los datos. Su funcionamiento se puede observar en la Ilustración 18:

Listado de Voluntarios

Mostrando 10 elementos por página

Nombre *	Apellidos	Dirección	CP	Localidad	Provincia	País	Teléfono	E-mail	Sede	Acciones
Chrystal	Camis	2499 Sullivan Avenue	37130-000	Affenas	Rio de Janeiro	Brasil	55-(933)565-7976	ccams4@studipress.com	Delegación Brasil	Editar Detalle Eliminar
Clair	Giberd	7 Burrows Crossing	2005	Albacele	Castilla - La Mancha	España	34-(940)208-8553	cgberd1@360.cn	Sede Central	Editar Detalle Eliminar
Clark	Clarkson	697 Westridge Plaza	8075	Barcelona	Cataluna	España	34-(743)138-5797	cclarkson1@prig.org	Delegación Málaga	Editar Detalle Eliminar
Clem	Hyatt	58560 Clarendon Lane	31005	Pamplona/ruña	Navarra	España	34-(256)555-3599	chyatt18@ocn.ne.jp	Delegación Málaga	Editar Detalle Eliminar
Clementina	Deverose	8331 Fulton Point	22005	Huesca	Aragon	España	34-(100)426-1941	cdeveroseh@shop-pro.jp	Sede Central	Editar Detalle Eliminar
City	Warrack	5 Northwestern Avenue	28300-000	Itaperuna	Sao Paulo	Brasil	55-(706)249-7563	cwarrack1@nature.com	Delegación Brasil	Editar Detalle Eliminar
Ciyd	Shattock	7 Village Green Alley	31015	Pamplona/ruña	Navarra	España	34-(818)241-9327	cshattockb@indiatimes.com	Delegación Málaga	Editar Detalle Eliminar
Cos	Vedstra	47 Kennedy Crossing	78968-000	Presidente Médici	Rio de Janeiro	Brasil	55-(327)831-3354	cvedstra8@mozilla.org	Delegación Brasil	Editar Detalle Eliminar
Costa	Goodger	61 Lakeshore Parkway	61900-000	Maracanã	Sao Paulo	Brasil	55-(234)665-2974	cgoodger5@don.ne.jp	Delegación Brasil	Editar Detalle Eliminar
Crysta	Blanko	7033 Garrison Terrace	3611	San Juan	Buenos Aires	Venezuela	34-(140)222-0994	cbankoq@sun.com	Delegación Argentina	Editar Detalle Eliminar

Mostrando página 4 de 17

Ilustración 18. Implementación de la vista "Listado de Voluntarios"

En el resto de listados de la aplicación, su diseño y funcionamiento es similar, por lo que únicamente se mostrará a continuación el listado de colaboradores en la Ilustración 19 por tener unos botones adicionales, correspondientes a los requisitos de generar los certificados para la declaración de la renta y la creación de remesas bancarias:

Listado de Colaboradores

Mostrando 10 elementos por página

Nombre	Apellidos	País	Teléfono	E-mail	CIF/NIF	Cuenta Bancaria	Perfiles	Cantidad	Fecha Alta	Periodicidad	Acciones
Addia	Yeam	España	34-(580)824-1569	ayeam1a@facebook.com	75-7005884	FR676435166020YOGUAWKZP2G88	Colaborador	71.64	27/04/2012	Trimestral	Editar Detalle Eliminar
Aurelie	Shawcroft	España	34-(759)643-3747	ashawcroft1@goodreads.com	67-6780264	AE645148852106410590638	Colaborador	93.7	12/07/2016	Trimestral	Editar Detalle Eliminar
Barnabe	Eickhoff	España	34-(218)794-6713	beickhoff1@domainmarket.com	03-3350239	MR7653034560407319260127116	Colaborador	85.15	27/09/2012	Trimestral	Editar Detalle Eliminar
Bayard	Stoppard	España	34-(605)650-5924	bstopparde@pweb.com	54-0569493	IE66ESMU95841250270054	Colaborador	22.14	14/06/2011	Trimestral	Editar Detalle Eliminar
Brok	Mawditt	España	34-(781)717-5204	bmawditt@shareasale.com	72-3078470	FR843081524457CVJWJUGCDBB95	Colaborador	42.07	18/11/2013	Trimestral	Editar Detalle Eliminar

Mostrando página 1 de 4 (total de 70 registros)

Ilustración 19. Listado de colaboradores filtrado por periodo trimestral

Para la inserción de nuevos datos o edición de un registro ya existente, se presenta una vista con un formulario en dos columnas (en móvil sólo en una) para mostrar al usuario los datos sin que tenga que desplazarse por la pantalla. Se considera que estas vistas, al igual que la de detalle, son similares en cuanto a diseño, puesto que la diferencia reside en que en los formularios de edición y detalle se carga la información almacenada, mientras que en el de introducir datos (o similar en otros “features”), el formulario aparece en blanco, tal como se observa en la Ilustración 20:

Crear Nuevo Voluntario

Nombre

Dirección

Localidad

País

Teléfono 2

Fecha Nacimiento

Sede

Apellidos

CP

Provincia

Teléfono

E-mail

Fecha Alta

Perfiles

• Página Web SPS
• Correo Corporativo SPS
• Formación Voluntariado Internacional

W3C CSS HTML5 WCAG 2.0

© 2017 - Gestor ONGD Fundación Spínola Solidaria - Desarrollado por Joaquín A. Duro Arribas

Ilustración 20. Formulario de creación de un voluntario.

Como en el caso del listado de colaboradores, el formulario de creación y edición de un colaborador tiene un estilo y funcionamiento diferente separando el formulario en dos pasos, uno para introducir los datos de los colaboradores y, después (siempre que se hayan rellenado todos los campos requeridos se activa el botón siguiente), permitiendo introducir los datos referentes a la donación, puesto que una persona no puede ser colaborador sin tener una donación. Su comportamiento se puede ver en la Ilustración 21:

Crear Nuevo Colaborador

1 Colaborador 2 Donación

Nombre

Dirección

Localidad

País España

Teléfono 2

Fecha Nacimiento

Apellidos

CP

Provincia

Teléfono

E-mail

Perfiles Colaborador

[Siguiente](#)

• Página Web SPS
• Correo Corporativo SPS
• Formación Voluntariado Internacional

W3C CSS W3C HTML 4.01 W3C XML-AAA WCAG 2.0

© 2017 - Gestor ONGD Fundación Spínola Solidaria - Desarrollado por Joaquín A. Duro Arribas

Ilustración 21. Paso 1 del formulario de creación de un nuevo colaborador.

Una vez se han rellenado todos los campos, se accede al segundo paso del formulario, introduciendo los datos referentes a la donación. Se puede observar en la Ilustración 22 que, para seleccionar la fecha de alta, el sistema muestra un selector de fecha gracias a la librería “*jQuery UI DatePicker*”

Crear Nuevo Colaborador

1 Colaborador 2 Donación

CIF/NIF

Cuenta Bancaria

Cantidad

Periodicidad Trimestral

Fecha Alta

[Anterior](#) [Guardar](#) [Volver](#)

Lu	Ma	Mi	Ju	Vi	Sá	Do
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

• Página Web SPS
• Correo Corporativo SPS
• Formación Voluntariado Internacional

W3C CSS W3C HTML 4.01 W3C XML-AAA WCAG 2.0

© 2017 - Gestor ONGD Fundación Spínola Solidaria - Desarrollado por Joaquín A. Duro Arribas

Ilustración 22. Paso 2 del formulario de creación de un nuevo colaborador.

Otra de las vistas que la aplicación pone a disposición al usuario es la reflejada en la Ilustración 23, dónde se pueden ver los datos de un colaborador y, posteriormente, la donación que está realizando a la fundación. Cómo se puede observar, el sistema pregunta al usuario antes de eliminar el dato y le informa de que también será eliminada la donación:



Ilustración 23. Vista de eliminación de un colaborador y su donación asociada.

Al realizar cualquier operación de creación, edición o eliminación de datos, la aplicación muestra al usuario un mensaje informativo sobre el resultado de la operación como se puede ver en la Ilustración 24:



Ilustración 24. Vista del mensaje informativo al usuario después de realizar una operación.

Al pulsar sobre el botón “GENERAR CERTIFICADOS”, el sistema detecta la fecha actual y genera un documento en formato PDF por cada uno de los colaboradores con la cantidad total donada durante el año anterior. El documento tiene el aspecto que se puede ver en la Ilustración 25:



SPÍNOLA SOLIDARIA
Educar es formar el corazón

Dña. Rebeca Collado Pérez con DNI [REDACTED] como Directora de la FUNDACION SPINOLA SOLIDARIA, con CIF: [REDACTED] y sede en la C/ Cardenal Marcelo Spínola, nº 34, 28016 de MADRID,

CERTIFICA:

- 1º.- Que la Fundación SPINOLA SOLIDARIA se encuentra incluida entre las entidades reguladas en el Capítulo 1º Título III de la Ley 49/2002 de régimen fiscal de las entidades sin fines lucrativos y de los incentivos fiscales al mecenazgo.
- 2º.- Que ha recibido de DON/DOÑA Duro Arribas, Joaquín Alejandro con NIF [REDACTED] la cantidad de 182,14 €, en el año 2016, como donación. Dicha cantidad fue entregada con carácter irrevocable.
- 3º.- Que el importe donado se destina al cumplimiento de los fines específicos de la Fundación.

El presente certificado se emite a los efectos previstos en el art. 55 de la Ley 40/1998, de 9 de diciembre, del impuesto sobre la Renta de las Personas Físicas y a los de los artículos 17, 18, 19 y 24 de la Ley 49/2002, de 23 de diciembre, régimen fiscal de las entidades sin fines lucrativos y de los incentivos fiscales al mecenazgo.

Y para que así conste, expido la presente certificación en Madrid a 1 de marzo de 2017.

Dña. Rebeca Collado Pérez
Directora

FUNDACIÓN SPÍNOLA SOLIDARIA
C/Cardenal Marcelo Spínola nº34/28016/Madrid/ Tel. 91 827 66 81/670 913 507/email: info@spinolasolidaria.org

Ilustración 25. Ejemplo de certificado de donaciones realizadas por un colaborador en PDF

Pulsando en el otro botón titulado como “EMITIR REMESAS”, se genera un fichero XML con el formato mostrado en formato SEPA con los datos de los adeudos a pasar al banco para recibir el pago de las donaciones.

5. Descripción técnica

5.1. Arquitectura

Como se ha mencionado anteriormente, se trata de una aplicación web para facilitar el acceso a los usuarios del mismo, proporcionar una interfaz sencilla e intuitiva basado en el patrón arquitectónico de aplicación distribuida cliente-servidor. Este modelo se define como: “El modelo arquitectónico cliente-servidor es un modelo de sistema en el que dicho sistema se organiza como un conjunto de servicios y servidores asociados, más unos clientes que acceden y usan los servicios.” (Somerville, 2005, pág. 488)

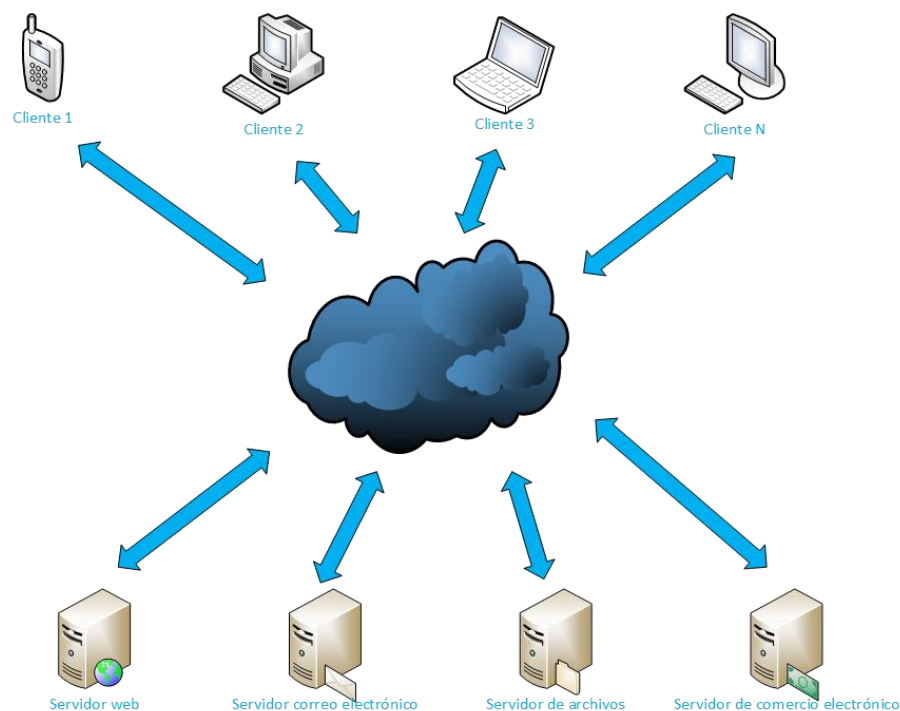


Ilustración 26. Diagrama de red de la arquitectura "cliente-servidor"

Dentro de este modelo, se utilizará una arquitectura "multi-tier" que se define como una arquitectura software en la que los diferentes componentes, organizadas en "tiers", aportan diferente funcionalidad. La arquitectura "multi-tier" más común es el sistema "three-tier" es la compuesta por un nivel de datos, un nivel de aplicación y un nivel de cliente. (Schuldt, 2009, págs. 1862-1865)

Esta arquitectura es la que se utiliza en este proyecto, definiendo los tres niveles a continuación y que se muestra gráficamente en la Ilustración 27:

- ✓ Capa de presentación: Contiene las interfaces de usuario. Muestran y capturan la información que el usuario ingresa en el sistema. Esta capa se comunica con la capa de lógica de negocio.

- ✓ Capa de lógica de negocio: Contiene la lógica del sistema. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de la base de datos almacenar o recuperar datos en/de ella.
- ✓ Capa de datos: Permite obtener información de la base de datos del sistema. También se podrá registrar y eliminar información de la base de datos.

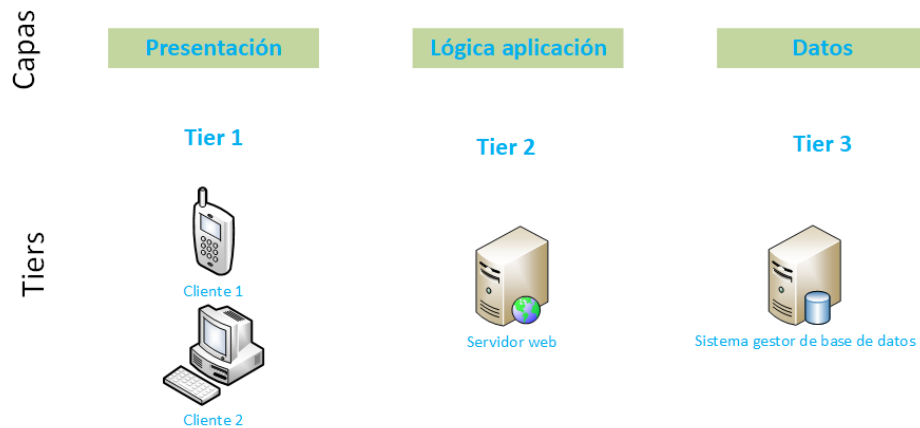


Ilustración 27. Arquitectura cliente-servidor 3-tier

En la Ilustración 28 se puede ver la arquitectura del sistema relacionando las diferentes capas con el software y hardware necesario, destacando que la capa de lógica de negocio y la capa de datos están en un servidor con RAID 1 para garantizar la seguridad de la información y evitar la pérdida de la misma:

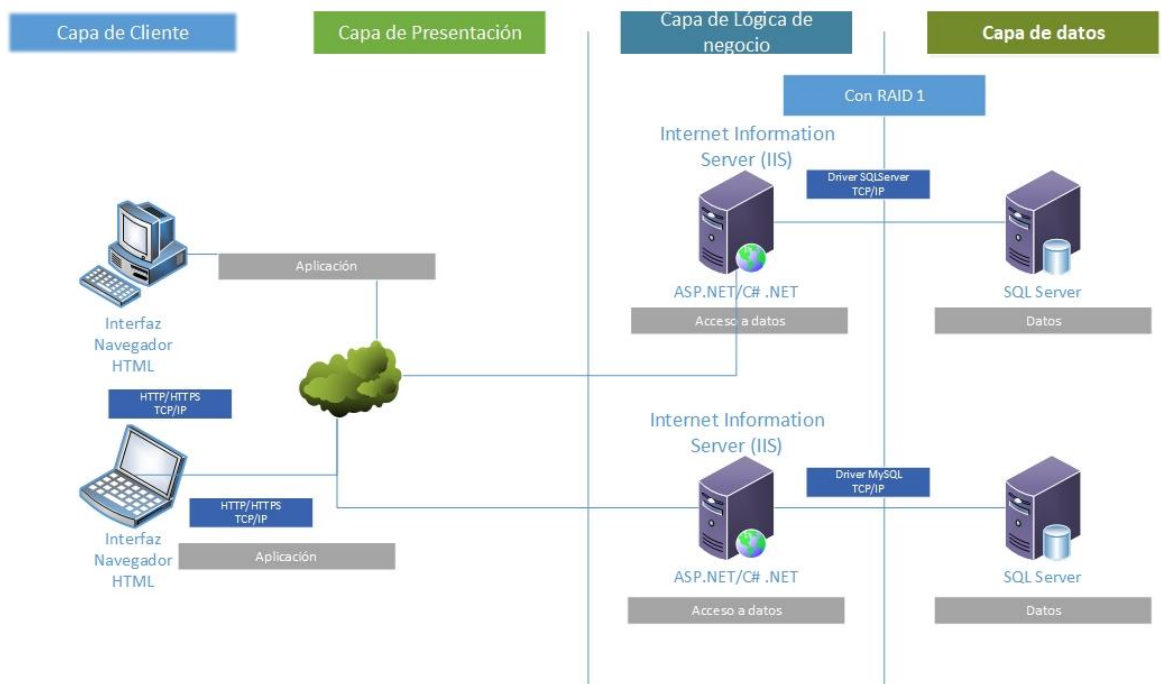


Ilustración 28. Arquitectura de red del sistema

Particularmente, se utilizarán diferentes tecnologías (detalladas en el punto Tecnologías utilizadas para la implementación del sistema reflejadas en la Tabla 8, en función de las capas dónde sea necesario:

Tabla 8. Relación de la arquitectura con las tecnologías utilizadas

Capa	Tecnología
Presentación	ASP.NET
	JQuery
	Bootstrap
Lógica de negocio	C# .NET
	PHP
Datos	Transact-SQL
	JSON
	XML

5.1.1. Capa de Presentación

Uno de los requisitos funcionales y objetivo de la aplicación era que ésta fuera sencilla e intuitiva. Por ello, es necesario que los elementos que componen la vista sean lo más atractivos, con la información justa y necesaria y que tenga un buen diseño que facilite el uso de esta aplicación. Como se puede observar en la Ilustración 29, el sistema muestra un formulario en dos columnas con un diseño limpio y evitando al usuario la acción de *scroll* vertical para ver todo el contenido.

Ilustración 29. Implementación de la vista "Editar nuevo Voluntario"

5.1.2. Lógica de Negocio o Modelo

Esta capa contiene las diferentes clases que componen el modelo de la aplicación, es decir, las clases que representan a las diferentes entidades que intervienen en la lógica de la aplicación. Sommerville define un diagrama de clases como “Los diagramas de clase pueden usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar las clases en un sistema y las asociaciones entre dichas clases.” (Sommerville, 2011, pág. 129)

Estos objetos son utilizados para transferir información entre el controlador y la vista o viceversa.

En la Ilustración 30 se pueden ver las dos clases del modelo desarrolladas específicamente para enviar información a la vista compuesta por dos o más clases de las que se ven en la Ilustración 31 :

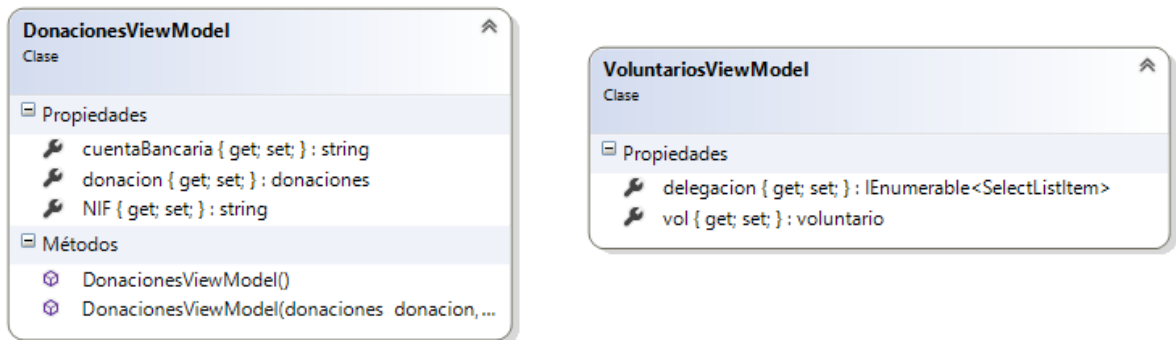


Ilustración 30. Diagrama de clases parcial compuesto por las clases que incluyen otras clases.

Mientras que en la Ilustración 31 se pueden observar las clases que componen el modelo dónde cada clase hace referencia a su respectiva tabla de la base de datos. En el diagrama de clases, se puede observar la relación de herencia de las clases colaboradores y voluntarios con personas como clase base (abstracta).

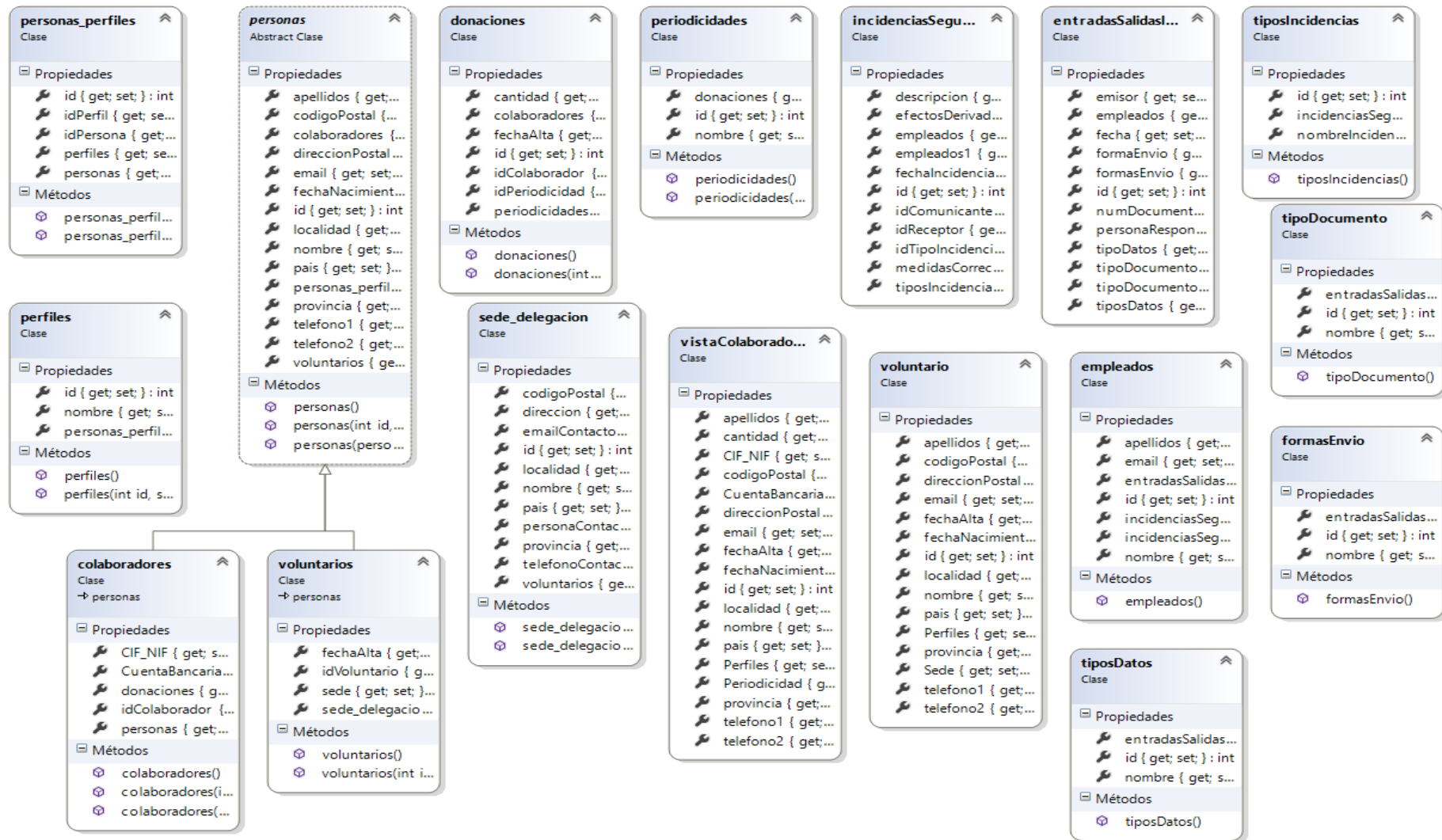


Ilustración 31. Diagrama de clases de la aplicación.

5.1.3. Capa de Datos

La capa de datos está compuesta por la base de datos. “Una base de datos relacionales es aquella que representa los datos y las relaciones entre los datos mediante una colección de tablas, cada una con un nombre único, donde una fila de una tabla representa una relación entre un conjunto de valores.” (Jiménez Capel, 2014, pág. 7)

Para conectar la aplicación con la base de datos, se ha utilizado el “*framework*” conocido como “*Entity Framework*” (véase Manipulación de datos en la base de datos mediante Transact-SQL y Entity Framework) en su modalidad “*Database First*”. Como se comenta en el sitio web “*Microsoft Developer Network*”, la técnica “*Database first*” permite al desarrollador realizar ingeniería inversa y generar el modelo desde una base de datos existente. (Ahmad, 2016)

En la Ilustración 32 se pueden ver las dos entidades generadas para agrupar la información de las vistas generadas en la base de datos, una para mostrar la información agrupada de los voluntarios y otra para los datos de los colaboradores y sus donaciones:

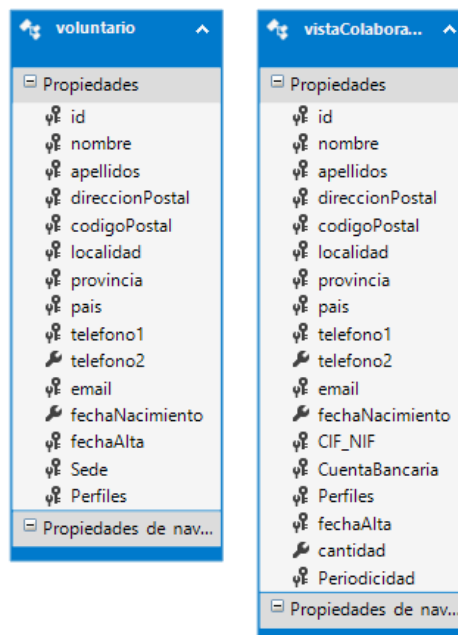


Ilustración 32. Vistas de la base de datos que componen el modelo.

Dentro de la Ilustración 33, se ven el resto de tablas que componen la base de datos y las relaciones entre ellas formando el modelo conocido como “entidad-relación”, que se define como “un modelo conceptual, basado en una percepción del mundo real mediante una colección de objetos, que como su nombre indica, se denominan entidades y relaciones, dónde la entidad es un objeto real sobre el cual quiere almacenarse información y la relación es la asociación entre las distintas entidades.” (Jiménez Capel, 2014, págs. 17-18)



Ilustración 33. Modelo Entidad-Relación de la base de datos.

Y a continuación, en la Ilustración 34, se puede ver reflejado su correspondencia en el modelo de datos, gracias al uso de “*entity-framework database first*”.

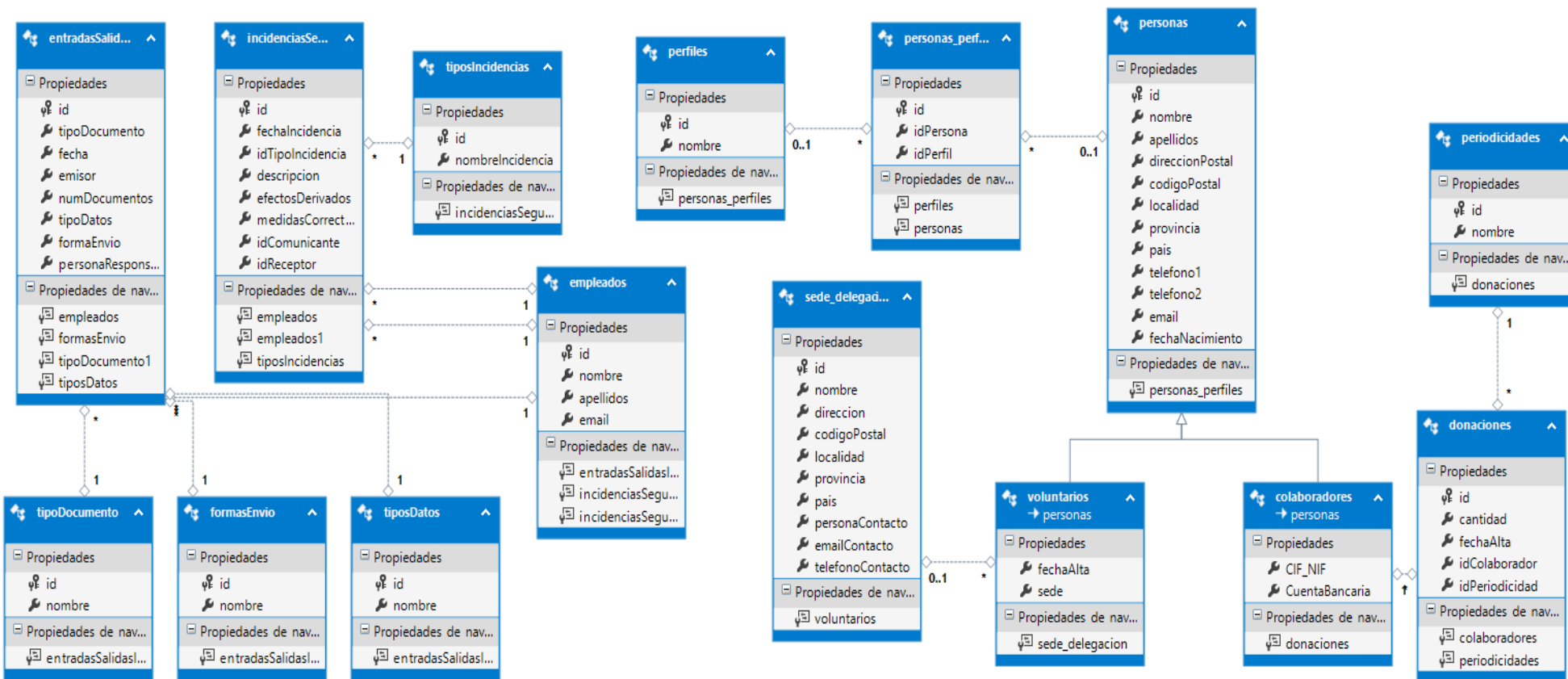


Ilustración 34. Modelo de datos utilizando "entity-framework"

5.1.4. Patrones de diseño

Adicionalmente, para el diseño de la aplicación se utilizarán diferentes patrones de diseño para modularizar la aplicación y darle flexibilidad. Se define un patrón de diseño como:

“El diseño basado en patrones crea una nueva aplicación al encontrar un conjunto de soluciones probadas para un conjunto claramente delineado de problemas. Cada problema y su solución se describen mediante un patrón de diseño que ha sido catalogado y examinado por otros ingenieros de software que han tenido el mismo problema y han implementado la solución.” (Pressman, 2010, pág. 347), esto significa que proporcionan una solución ya probada y documentada a un problema similar.

En el desarrollo del sistema, se hará uso de los siguientes patrones:

5.1.5. Patrón auxiliar Modelo-Vista-Controlador

“La arquitectura Modelo-Vista-Controlador (MVC) es uno de los diversos modelos sugeridos para las aplicaciones web que desacoplan la interfaz de usuario de la funcionalidad y el contenido.

El patrón MVC divide la aplicación en tres componentes (Pressman, 2010, págs. 386-387):

- ✓ El modelo, que contiene la funcionalidad básica y los datos. Por esto, está formado por los elementos de la capa de negocio y por los elementos de la capa de datos.
- ✓ La vista, que muestra y recoge información del usuario. Por esto, está formada por los elementos de la capa de presentación.
- ✓ El controlador, que media entre la vista y el modelo, gestionando las peticiones del usuario, eligiendo el comportamiento del modelo y llamando a la vista correspondiente.”

Para la implementación de este patrón, se utilizará la plantilla de proyecto disponible en el software Microsoft Visual Studio 2015 de ASP.NET MVC, como se puede observar en la Ilustración 35, donde el IDE (entorno de desarrollo) mencionado anteriormente, permite elegir entre varias plantillas disponibles en función de las necesidades. Esta plantilla proporciona una estructura de carpetas orientada a la arquitectura “modelo-vista-controlador”, así como una serie de “*namespaces*” (espacios de nombre que se utilizan para organizar múltiples clases (Wagner, Yishengjin1413, & Cai, 2017)). Esta plantilla también está orientada a la realización de pruebas, por lo que permite un desarrollo ágil del proyecto y sus respectivas pruebas.

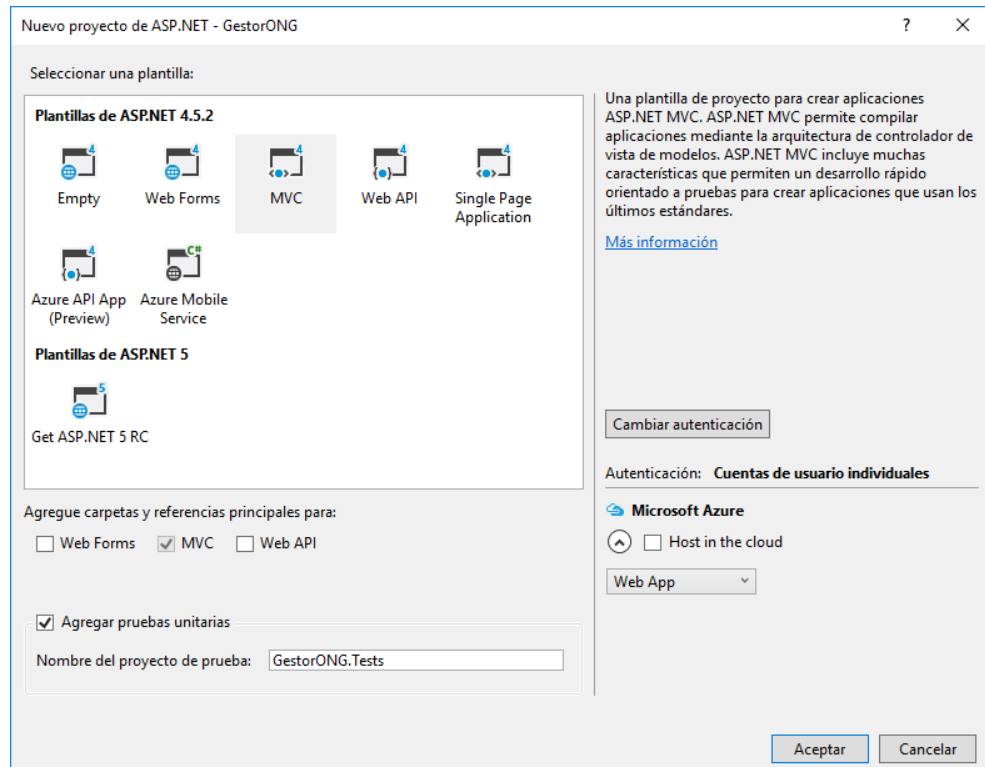


Ilustración 35. Proyecto ASP.NET con MVC

5.1.6. Patrón auxiliar “Singleton”

El patrón “*Singleton*” garantiza que sólo hay una instancia de una clase, proporcionando un único punto de acceso global a ella. (Millett, 2010, pág. 15)

Es utilizado en el diseño del controlador de la aplicación.

5.1.7. Patrón DTO (Data Transfer Object) u Objeto de Transferencia de Datos

Al trabajar con varias capas, un problema que puede surgir es cómo pasar los datos de una capa a otra. Aquí es donde aparecen los objetos de transferencia de datos.

Este tipo de objetos no tienen ningún tipo de lógica de negocio y son simples contenedores de datos estructurales. En estos objetos, se encapsularán los datos (generalmente conocidos como atributos de la clase) y que serán accesibles mediante métodos “*getters*” y “*setters*”, para poder transportar estos datos entre los diferentes niveles con propósitos de comunicación.

El patrón se utiliza en las clases del modelo para permitir el acceso a los atributos de la respectiva clase. Se puede ver un ejemplo de la implementación a continuación, donde se definen los diferentes atributos de la clase (id, nombre, etc..) y los métodos “*getters*” y “*setters*” mediante las palabras “*get*” y “*set*” que se encuentran entre los símbolos “*{}*”.

```

/// <summary>
/// ID Autoincremental de la persona
/// </summary>
public int id { get; set; }

/// <summary>
/// Nombre de la persona
/// </summary>
[DisplayName("Nombre")]
[Required]
[StringLength(100)]
public string nombre { get; set; }
    
```

En la parte izquierda de la Ilustración 36, mediante diagramas de secuencia UML, se puede observar la diferencia entre la comunicación de datos sin utilizar el patrón, mientras que, en la parte derecha, se puede ver cómo se reducen considerablemente las llamadas a estos métodos “getters” y “setters”: (Fowler, 2003)

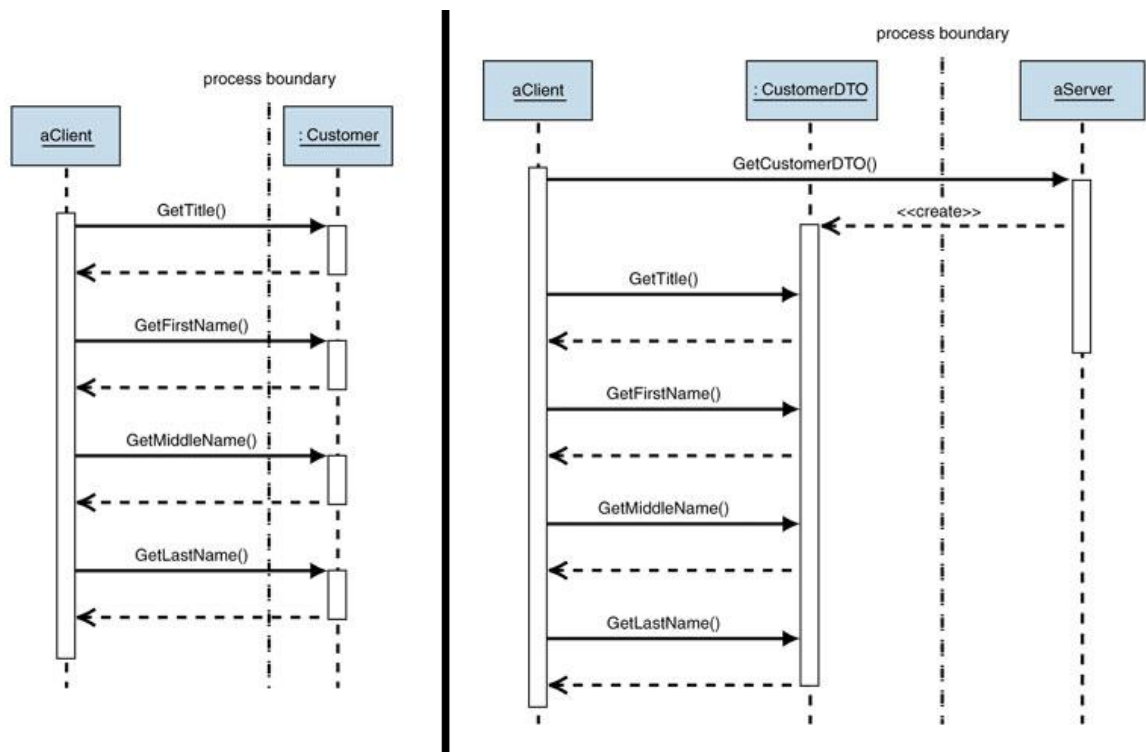


Ilustración 36. Diferencias comunicación patrón "DTO"

Fuente: Microsoft Corporation. (s.f.). Data Transfer Object. Obtenido de Microsoft Developer Network: <https://msdn.microsoft.com/en-us/library/ff649585.aspx>

5.1.8. Patrón DAO (Data Access Object) u Objeto de Acceso a Datos.

Un objeto de Acceso a Datos es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos como una base de datos, un archivo XML u otro tipo de fichero, y que permite aislar la aplicación de la tecnología de persistencia subyacente.

Se trata de un objeto con nivel de funcionalidad. Su responsabilidad será desempeñar operaciones “CRUD” (Create/Read/Update/Delete) con la base de datos.

Las clases que implementas este patrón están implementadas dentro del directorio DAL (Data Access Layer), ubicado en el repositorio del proyecto GIT. En particular, las operaciones CRUD están implementadas mediante la clase “**GestorONGDContext.cs**”, utilizada por “ADO.NET” y “Entity Framework”.

5.1.9. Patrón “Active Record”

“Se trata de un patrón muy popular que es especialmente efectivo cuando el modelo de la base de datos coincide con el modelo de la aplicación. Típicamente, un objeto de la lógica de negocio existe para cada una de las tablas de la base de datos, ese objeto representa una única fila de la tabla y contiene data y el comportamiento necesario para persistir y métodos para añadir nuevas instancias y buscar colecciones de objetos (generalmente métodos CRUD” (Millett, 2010, pág. 59)

Este patrón ha sido utilizado en el proyecto de manera implícita utilizando “*ADO.NET*” y “*Entity Framework Database First*” dónde cada tabla de la base de datos hace referencia a una clase del modelo (en el espacio de nombres “GestorONGDSPS.DataModel”).

5.1.10. Patrón “Layer Supertype”

“Define un objeto que actúa como clase base para todos los tipos de su capa, y está basado en torno a la herencia. Su objetivo es compartir la lógica de negocio común, evitando los duplicados y centralizando la lógica” (Millett, 2010, págs. 127-130)

Se ha utilizado el patrón definiendo la clase abstracta “persona” que contiene toda la lógica común a las personas, y las clases “voluntarios” y “colaboradores” que acceden a la lógica común de personas mediante una llamada a su clase base en los constructores y tienen definida también su propia lógica de negocio.

5.2. Prototipo gráfico

Como se ha mencionado en la sección de Objetivos específicos, la interfaz de la aplicación debe ser sencilla e intuitiva, con un diseño limpio y sobre todo, que cumpla con las principales recomendaciones de usabilidad y accesibilidad web promovidas por los diferentes organismos, particularmente el W3C.

5.2.1. Vista de la pantalla listado de colaboradores

En la Ilustración 37, se puede observar un prototipo gráfico del diseño de un listado de datos (en este caso de colaboradores), con su correspondiente paginación y los diferentes botones que permiten insertar, modificar o eliminar un colaborador.

SPINOLA Solidaria

Voluntarios Colaboradores Donaciones Sedes Correspondencia LOPD Configuración

Listado de Colaboradores

NUEVO COLABORADOR EDITAR COLABORADOR ELIMINAR COLABORADOR

FILTROS: CIUDAD PERIODICIDAD

Nombre	Apellidos	Dirección	Ciudad	Código Postal	Cantidad	Periodicidad	Nº Cuenta	DNI
Eduardo	Pérez García	C/ Dirección de prueba	Madrid	28016	60 €	Trimestral	ES90123456789012345	12345678A
Juan	García Jiménez	C/ Otra dirección	Málaga	29070	50 €	Mensual	ES80123456789012345	23456789B
Joaquín	Duro Arribas	C/ Dirección desconocida	Soria	42001	200 €	Anual	ES230123456789012345	34567890C

NUEVO COLABORADOR EDITAR COLABORADOR ELIMINAR COLABORADOR

« 1 2 3 4 5 »

Página Web SPS
 Correo Corporativo SPS
 Formación Voluntariado Internacional
 Campaña Repiensa, Reacciona, Redescubre
 Google Apps SPS

SPINOLA Solidaria

Ilustración 37. Vista de un listado de la aplicación

5.3. Tecnologías utilizadas

Para el desarrollo del proyecto se han utilizado diferentes tecnologías y lenguajes de programación que se describen a continuación, añadiendo una breve explicación sobre dónde ha sido utilizada cada una de ellas.

5.3.1. Desarrollo del proyecto con ASP.NET

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET. (Guérin & Piqueres Juan, 2016, pág. 15)

El desarrollo del proyecto se ha realizado sobre este modelo de desarrollo a través de la herramienta Microsoft Visual Studio 2015 a través de la plantilla de proyecto que incluye la estructura necesaria para implementar un proyecto con el modelo MVC (Modelo-Vista-Controlador).

5.3.2. Desarrollo del modelo y el controlador en C# .NET

“Para entender lo que es C# es imprescindible decir antes lo que es Microsoft .NET Framework o abreviadamente .NET. Se trata de un entorno de desarrollo multilenguaje diseñado por Microsoft para simplificar la construcción, distribución y ejecución de aplicaciones para Internet. Tiene fundamentalmente tres componentes: una máquina virtual

(CLR: Common Language Runtime) que procesa código escrito en un lenguaje intermedio (MSIL: Microsoft Intermediate Language), una biblioteca de clases (biblioteca .NET) y ASP.NET que proporciona los servicios necesarios para crear aplicaciones Web.” (Ceballos Sierra, 2007, pág. 2)

Una vez explicado que es .NET, se puede definir C# como “un lenguaje de programación que es descendiente directo de dos de los más exitosos lenguajes de programación: C y C++ y, también está emparentado cercanamente con Java” (Schildt, 2010, pág. 3)

Este lenguaje de programación está muy presente en el desarrollo de este proyecto, utilizándose principalmente para realizar la programación de las clases que componen el modelo y el controlador de la aplicación.

5.3.3. Diseño gráfico y responsive con Bootstrap

Se trata del framework más popular basado en HTML, CSS y Javascript (JS) para desarrollar proyectos web con diseño adaptativo (“*responsive-design*”).

Este framework ha sido de utilidad en el proyecto para aportar estilo y diseño a las diferentes vistas que componen la aplicación. Mediante la inclusión de ciertas clases en los elementos HTML que contiene la vista, permite implementar un diseño gráfico que sea compatible en todos los tamaños de pantalla (ordenadores, “*tablets*” y “*smartphones*”. Bootstrap ¹⁰permite la utilización y modificación del código bajo una licencia MIT.

5.3.4. Optimización del estilo del portal web con CSS

Se trata de un lenguaje de estilos que se ha utilizado para mejorar el aspecto gráfico de la interfaz gráfica. Todos los archivos de estilos se encuentran ubicados en la carpeta “Content” del proyecto.

5.3.5. Mejora de funcionalidades con Javascript y JQuery

Se trata de un lenguaje y una librería (jQuery¹¹) que permite manipular datos, responder a eventos, animar elementos HTML y realizar peticiones Ajax de una manera más sencilla. En otras palabras, se podría definir la librería JQuery como una librería basada en JavaScript que permite automatizar tareas comunes y simplifica otras para construir sitios web interesantes e interactivos. (Chaffer & Swedberg, 2011, pág. 9)

¹⁰ <http://getbootstrap.com/> Framework para desarrollar proyectos “responsive-design”

¹¹ JQuery es una biblioteca de Javascript que agiliza y ayuda al desarrollo de funcionalidades. <https://jquery.com/>

Además de ser un requisito para la utilización del framework “*bootstrap*”, se han incluido dos librerías basadas en JQuery para añadir funcionalidad al portal web. Mediante la librería “JQuery datatables¹²”, ha sido posible implementar la paginación, el filtrado de resultados o la ordenación de las diferentes vistas “Index” del proyecto (listado de voluntarios, listado de colaboradores, ect.), mientras que la librería JQuery UI Datepicker¹³, ha hecho posible que en los campos de tipo fecha, salga un selector de fecha para que la introducción de datos sea más intuitiva para el usuario. La librería JQuery datatables está disponible bajo una licencia MIT y JQuery UI Datepicker a través de una licencia Creative Commons.

Por último, se ha utilizado JQuery para realizar una petición Ajax a un servicio web y obtener una lista de países para incluir en los desplegables de creación y edición de un voluntario o de un colaborador.

También se ha utilizado AJAX (mediante JQuery) como tecnología de programación para realizar peticiones asíncronas desde el cliente (navegador) al servidor. (Millett, 2010, pág. 375)

5.3.6. Manipulación de datos en la base de datos mediante Transact-SQL y Entity Framework

Transact-SQL es el lenguaje que se utiliza para administrar instancias del SQL Server Database Engine (Motor de base de datos de SQL Server), para crear y administrar objetos de base de datos, y para insertar, recuperar, modificar y eliminar datos. (Microsoft Corporation, 2005). Se utilizará para acceder a la base de datos de la aplicación, desplegada sobre SQL Server.

Aunque el desarrollo en .NET no incluye sentencias Transact-SQL explícitamente, las consultas, inserciones, actualizaciones o eliminaciones de datos se han realizado mediante “*Entity Framework*” y “*ADO.NET*”.

“*Entity Framework* es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos. Entity Framework permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de cliente, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. “ (Microsoft Corporation, 2016). En

¹² Es un plugin para JQuery que proporciona controles interactivos avanzados para cualquier tabla HTML (paginación, ordenación, filtrado, etc.) <https://datatables.net/>

¹³ Permite que, al pulsar sobre un campo de texto, se muestre un calendario interactivo para seleccionar una fecha. <https://jqueryui.com/datepicker/>

otras palabras, se podría decir que “*Entity Framework*” es el ORM desarrollado por Microsoft.

Sin embargo, para la creación del modelo relacional en la base de datos SQL Server y/o creación de vistas, se ha utilizado este lenguaje. Todos los scripts realizados mediante este lenguaje se pueden encontrar en la carpeta del repositorio Github llamada **SQLScripts**.

5.3.7. Creación del fichero del banco con las remesas bancarias en formato XML

Estándar que significa "eXtensible Markup Language" o lenguaje de marcado extensible, que fue diseñado para almacenar y transportar datos. (W3Schools, s.f.).

El formato XML ha sido necesario en el proyecto puesto que es el único formato compatible con el software de la entidad del banco para almacenar la información de las remesas bancarias. Además, se tiene la particularidad de que el formato XML utilizado debe ser compatible con el formato XML SEPA definido por el estándar ISO 20022 dónde se establecen los requisitos técnicos y empresariales para las transferencias y los adeudos domiciliados en euros mediante la utilización de mensajes para las comunicaciones de órdenes de pago que no se transmitan de forma individual (CECA, 2015).

5.3.8. Generación de las remesas bancarias con PHP

PHP es un lenguaje de *scripts* que se ejecuta en un servidor web y permite el acceso a archivos, el acceso a bases de datos o la visualización de contenido al usuario mediante XHTML. (Harris, 2009). Se utilizará para crear el servicio web que creará las remesas bancarias en formato SEPA, utilizando una librería llamada “SepaWriter¹⁴” disponible en Github para este propósito y que necesita PHP como lenguaje de programación.

5.3.9. Intercambio de datos entre las diferentes capas en formato JSON

Es un formato ligero de intercambio de datos compuesto por dos estructuras: una colección de pares de nombre/valor; y una lista ordenada de valores. (JSON.ORG, s.f.).

Se ha utilizado para obtener la lista de países desde una RESTfull API llamada “REST COUNTRIES¹⁵”

¹⁴ Librería SepaWriter para generar adeudos bancarios en formato SEPA <https://github.com/perrich/sepawriter>

¹⁵ REST API “REST COUNTRIES” para obtener un listado de países <https://restcountries.eu/#filter-response>

5.4. Herramientas utilizadas

A continuación, se describen las diferentes herramientas utilizadas para el desarrollo del proyecto:

5.4.1. ArgoUML

Es una herramienta “open source” que incluye soporte para diagramas UML 1.4 y basado en la plataforma Java. (Tigris.org, 2009). Se ha utilizado para realizar los diagramas UML de la fase de análisis (modelo de negocio y casos de uso).

5.4.2. Microsoft Visual Studio 2015 Enterprise

Se trata de un entorno de desarrollo integrado (IDE) a través del cual se ha creado el proyecto ASP.NET MVC y se ha desarrollado todo el código de la aplicación. A través de la extensión de control de versiones “Team Server” se han realizado los “*commits*” y “*push*” del código en el repositorio.

5.4.3. Microsoft Visual Studio Team Services¹⁶

Es una herramienta en la nube que permite compartir código, realizar el seguimiento del trabajo (tareas, bug, product backlog ítems, features, ect.) mediante metodologías ágiles (Scrum, Kanban, Agile) y compilaciones de integración continua.

Ha sido de especial utilidad para la planificación de tareas mediante metodologías ágiles (concretamente Scrum, detallado en el elemento Metodología de trabajo; **Error! No se encuentra el origen de la referencia.**).

5.4.4. Mockaroo

Para la generación de datos de ejemplo aleatorios y su inserción en la base de datos, se ha utilizado la herramienta online “*Mockaroo*¹⁷” en su versión gratuita (limitada a un número máximo de 1.000 registros por tabla)

5.4.5. Bootstrap Studio

Es una herramienta para crear prototipos de portales web mediante el framework “bootstrap”. Mediante esta herramienta, además de crearse los prototipos gráficos detallados en el punto Prototipo gráfico de una manera ágil y sencilla, ha permitido la generación del código HTML basado en este framework que posteriormente ha servido para la realización de las vistas del portal web.

¹⁶ Seguimiento de proyectos de equipo de software <https://www.visualstudio.com/es/team-services/>

¹⁷ Mockaroo es una herramienta de generación de datos aleatorios, útil para generar datos de prueba de aplicaciones software <https://www.mockaroo.com/>

5.4.6. Microsoft Project 2016 Profesional

Software de administración y planificación de proyectos con el que se ha realizado el plan de trabajo del presente trabajo de fin de grado.

5.4.7. SQL Server 2016

Plataforma de base de datos de Microsoft que aporta información en tiempo real de los datos transaccionales y analíticos. (Microsoft Corporation, 2016). Aloja la base de datos principal del proyecto que ha sido gestionada mediante la herramienta proporcionada por Microsoft “SQL Server Management Studio (SSMS)”.

5.4.8. Selenium IDE

Selenium IDE¹⁸ es un add-on para el navegador Firefox desarrollado originalmente por Shinya Kasatani como una vía para usar el núcleo original de *Selenium* sin tener que copiar éste es un servidor. Fue desarrollado para permitir a los desarrolladores y *testers* grabar sus acciones siguiendo el flujo de acciones que se necesitan probar. (Burns, 2012, pág. 8)

Se ha utilizado para realizar las pruebas unitarias y de regresión en el proyecto.

5.4.9. Apache JMeterTM

Apache JMeterTM¹⁹ es una aplicación “*open-source*” basada al 100% en Java diseñada para cargar test funcionales del comportamiento y medir el rendimiento. (The Apache Software Foundation, 2017)

Se ha utilizado para cargar distintos números de peticiones simultáneas a la aplicación y obtener informes del rendimiento de la misma.

5.4.10. TAW

TAW²⁰ una herramienta web desarrollada por el CTIC para comprobar la accesibilidad de las páginas web, permitiendo seleccionar el nivel de accesibilidad y las tecnologías a validar.

¹⁸ Selenium IDE, Plugin para realizar pruebas automáticas <http://www.seleniumhq.org/projects/ide/>

¹⁹ Apache JMeterTM Herramienta para ejecutar test de rendimiento <http://jmeter.apache.org/>

²⁰ TAW Herramienta de validación de la accesibilidad web <http://www.tawdis.net/>

5.5. Implementación

Como ya se ha mencionado anteriormente en el punto Desarrollo del proyecto con ASP.NET, para la implementación del software se ha utilizado ASP.NET MVC y Microsoft Visual Studio 2015. A continuación, se describe el contenido de las diferentes carpetas y archivos principales que componen el proyecto (disponible en Github) mostradas en la Ilustración 38, omitiendo las vacías:

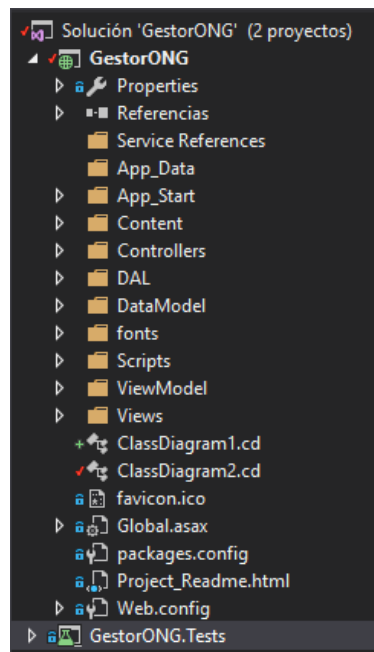


Ilustración 38. Carpetas que componen el código del proyecto.

- ✓ **App_Start:** Contiene tres archivos, dónde “BundleConfig.cs” se utiliza para registrar todas las librerías externas utilizadas en el proyecto y mencionadas en el punto Tecnologías utilizadas; “FilterConfig.cs” utilizado para definir configuraciones de filtrados de peticiones HTTP y “RouteConfig.cs” que el archivo utilizado por ASP.NET para personalizar el enrutamiento de la aplicación y definir rutas personalizadas.
- ✓ **Content:** En esta carpeta, que contiene a su vez subcarpetas y archivos, se definen los archivos CSS que utiliza la aplicación, destacando el archivo “Site.css” que contiene los estilos propios de la aplicación (los que no son de ninguna librería). Esta carpeta también incluye un directorio llamado “Images” dónde están almacenadas todas las imágenes que utiliza el sitio web.
- ✓ **Controllers:** Contiene las clases que gestionan la relación entre las vistas y el modelo, conocidos como Controladores. Incluye un controlador por cada uno de los módulos de la aplicación.
- ✓ **DAL:** Dentro de este directorio, se encuentran dos ficheros C#: “GestorONGDContext.cs” dónde se definen los métodos y las propiedades de “Entity

Framework” para realizar las operaciones CRUD en la base de datos; y el archivo “GestorONGDInilizer.cs” que contiene instrucciones de inserción de datos iniciales en la base de datos y que es llamado desde el archivo Web.config la primera vez que se ejecuta la aplicación.

- ✓ **DataModel:** En esta carpeta se encuentra el archivo con extensión “.edmx” que incluye las clases del modelo de datos generado por “*Entity Framework*” como se muestra en la Ilustración 34.
- ✓ **Fonts:** En este directorio se encuentran las tipologías personalizadas.
- ✓ **Scripts:** Dentro de esta carpeta, se incluyen todos aquellos archivos javascript (.js) utilizados para el desarrollo del proyecto.
- ✓ **ViewModel:** Contiene dos clases del modelo utilizadas para pasar información de distintas clases del modelo y tipos de datos (del espacio de nombres DataModel) a la vista.
- ✓ **Views:** Contiene una subcarpeta con las vistas con las que interactúa cada uno de los controladores y el archivo “_Layout.cshtml” dónde se introduce el código común a todas las páginas como es el caso del *header* (encabezado) con el menú superior y el *footer* (pie de página) del sitio web.
- ✓ **Global.asax:** Este archivo es el punto de entrada de la aplicación cuando se ejecuta.
- ✓ **Web.config:** En este fichero con estructura xml se definen parámetros como la versión de .NET a utilizar, el modo de autenticación de usuarios de la aplicación, los compiladores utilizados o la cadena de conexión con la base de datos.

Como ya se ha comentado anteriormente, se ha utilizado “GIT” como herramienta de control de versiones. A continuación, se describe qué es un sistema de control de versiones y “GIT”:

“Un sistema de control de versiones es un sistema capaz de grabar todos los cambios realizados sobre un archivo o conjunto de archivos en un periodo de tiempo de tal manera que permita volver atrás a una versión específica del código.” (Somasundaram, 2013, pág. 8)

“GIT es un paquete de software relativamente nuevo que fue diseñado desde cero para evitar fallos que existían en muchos otros sistemas de control de versiones. Linus Torvalds, el hombre que desarrolló el *kernel* de Linux, es el orgulloso iniciador de este proyecto. La propia arquitectura de GIT se adapta a una mayor velocidad, rendimiento, flexibilidad y usabilidad” (Somasundaram, 2013, pág. 15)

Entre sus múltiples comandos se encuentra “*git commit -m [mensaje]*”, que sirve para confirmar los cambios de los archivos junto con un mensaje dónde se describen los cambios realizados, y “*git push*”, que es el utilizado para introducir dichos cambios en el repositorio.

6. Evaluación y pruebas

En este capítulo se explican brevemente los tipos de pruebas utilizados para la evaluación y validación del proyecto, las pruebas que se han realizado y los resultados de las mismas.

6.1. Plan de pruebas

Para la realización de la validación del sistema se han realizado tres tipos de pruebas: las pruebas unitarias, las pruebas de regresión y las pruebas de rendimiento. A continuación, se explica en qué han consistido cada una de ellas.

6.1.1. Pruebas unitarias

Las pruebas de unidad o unitarias son el proceso de probar componentes del programa, como métodos o clases de objetos. Hay que escribir dos tipos de casos de prueba. El primero debe reflejar una operación normal de un programa y mostrar que el componente funciona. El otro tipo de caso de prueba tiene que basarse en probar la experiencia de donde surgen problemas comunes. Debe usar entradas anormales para comprobar que se procesan de manera adecuada sin colapsar el componente. (Somerville, 2005, págs. 211-213).

Se han llevado dos tipos de pruebas, manuales y automáticas. En ambos casos, estas pruebas dieron como resultado la identificación de 13 “bugs” o errores como se mostraba en la Ilustración 14 que se han solucionado.

Para la realización de las pruebas automáticas se ha utilizado el plugin llamado “Selenium²¹” mencionado en el apartado 5.4.8 Selenium IDE.

En la Ilustración 39 se muestra un ejemplo de la prueba realizada para crear un nuevo voluntario en el sistema. Al ser una prueba con datos normales, el test devuelve que la operación ha sido realizada satisfactoriamente, mostrando un pequeño resumen en la parte inferior de la pantalla de las operaciones ejecutadas:

²¹ Selenium, herramienta de automatización de pruebas <http://www.seleniumhq.org/>

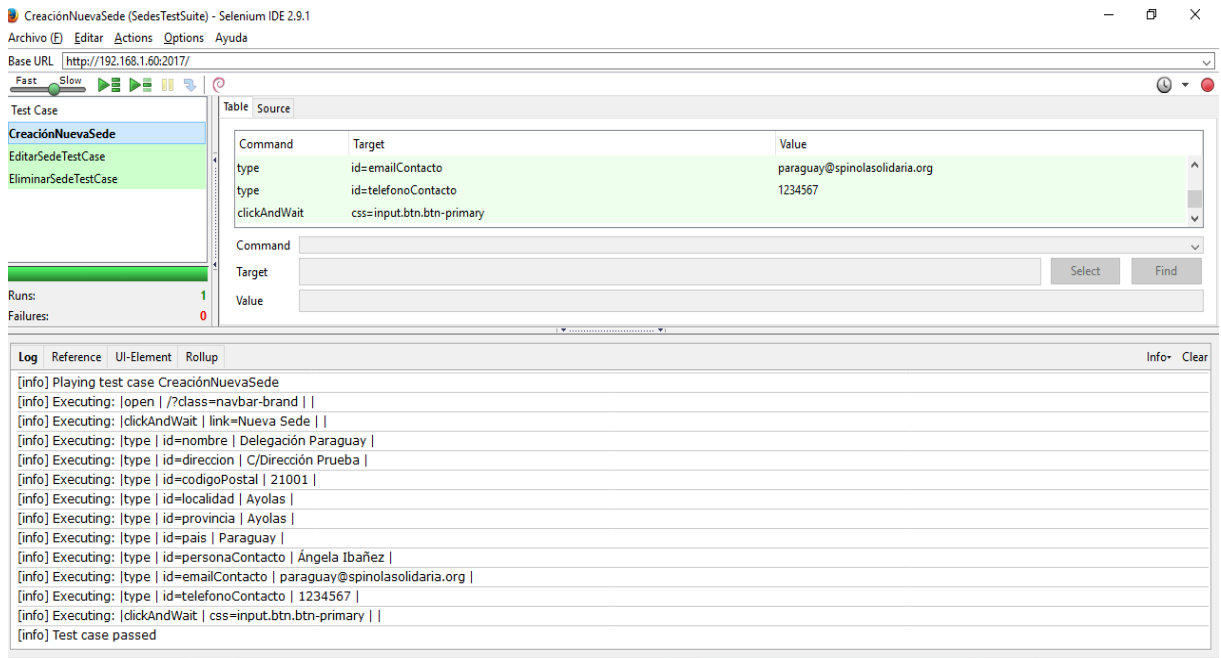


Ilustración 39. Ejecución del caso de prueba para crear una nueva sede con Selenium IDE.

Ahora, si se realiza una prueba manual de inserción de una nueva sede con datos que se saben que son incorrectos, se puede observar que la aplicación muestra una serie de errores de validación (sobre todo de longitud) y no permite continuar con la creación de la sede hasta que se han introducido todos los datos. Este comportamiento se puede ver en la Ilustración 40:



Ilustración 40. Cuando se introducen datos incorrectos, la aplicación muestra errores de validación.

De manera similar a la mostrada anteriormente, se han realizado pruebas unitarias del resto de módulos de la aplicación para asegurar su correcto funcionamiento, obteniendo que el sistema muestra un comportamiento estable.

6.1.2. Pruebas de regresión

Somerville define las pruebas de regresión como Un conjunto de pruebas se desarrolla incrementalmente conforme se desarrolla un programa. Siempre es posible correr pruebas de regresión para demostrar que los cambios al programa no introdujeron nuevos bugs. (Sommerville, 2011, pág. 223)

Para la realización de estas pruebas de regresión, se ha utilizado de nuevo el plugin “*Selenium IDE*”, lanzando un paquete de pruebas completo llamado “*test-suite*”.

Como ejemplo de este tipo de pruebas, cabe destacar la prueba del menú principal de la parte superior, dónde se ha lanzado la prueba de regresión cada vez que se ha introducido un controlador en la aplicación y, por tanto, se ha añadido su correspondiente menú con los enlaces a las diferentes acciones del controlador.

En la Ilustración 41 , se puede ver como al introducir todos los controladores, los test fallaban porque no se habían introducido los enlaces correctamente, concretamente, el enlace para crear una nueva donación no funcionaba (marcado en la parte superior de la imagen en rojo), lo que dio origen a un *bug* en el sprint correspondiente para cambiar el destino del enlace del elemento de menú y volver a pasar la prueba:

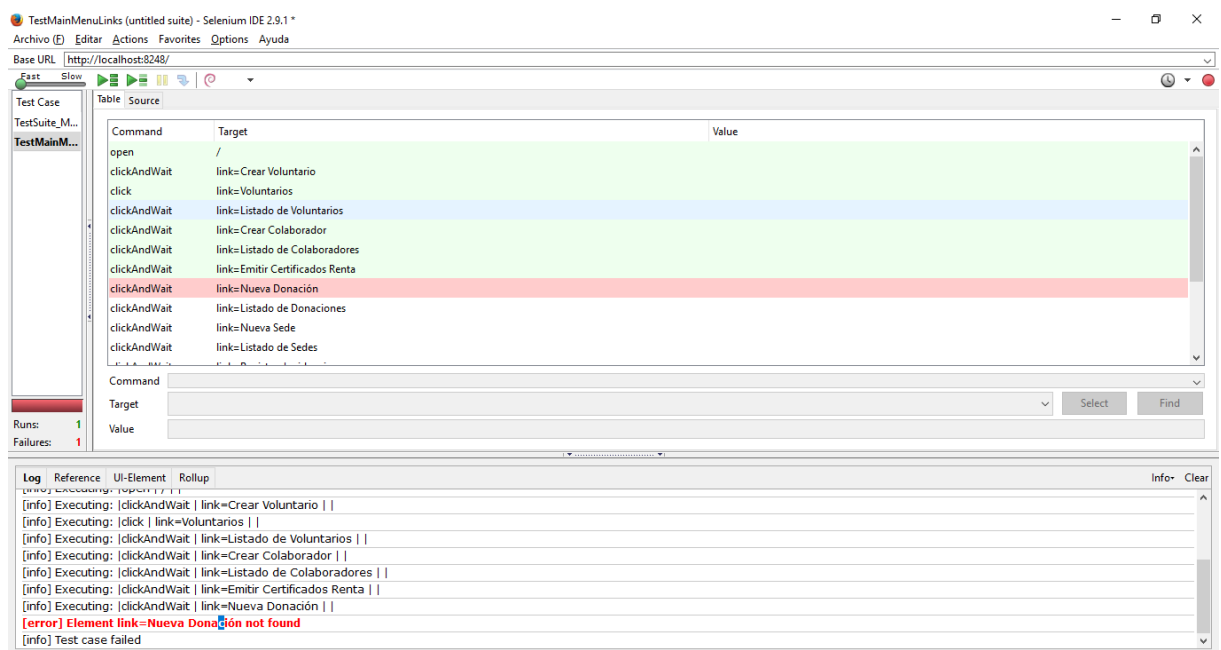


Ilustración 41. Prueba de regresión del menú principal con resultado fallido.

En cambio, en la Ilustración 42 , se muestra el resultado de la prueba automática de regresión del módulo de gestión de sedes y delegaciones, ejecutando un total de 3 casos de prueba con un resultado de éxito en todos ellos:

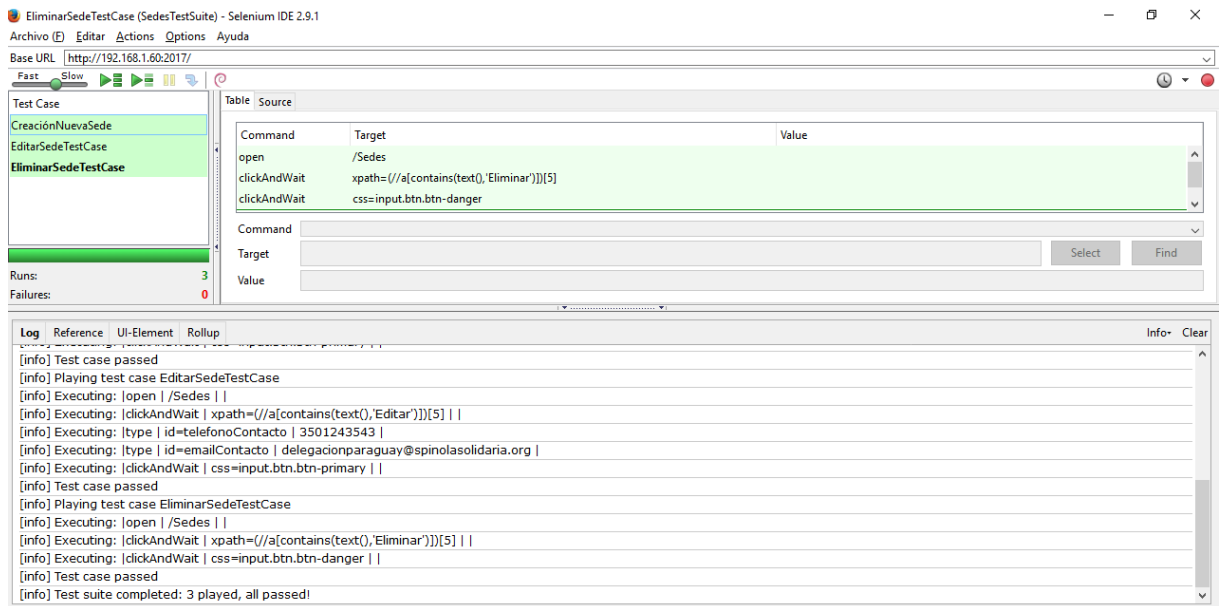


Ilustración 42. Resultado correcto al ejecutar las pruebas de regresión del módulo de sedes.

6.1.3. Pruebas basadas en requerimientos

“Un principio general de buena práctica en la ingeniería de requerimientos es que éstos deben ser comprobables; esto es, los requerimientos tienen que escribirse de forma que pueda diseñarse una prueba para dicho requerimiento. Luego, un examinador comprueba que el requerimiento se cumpla. En consecuencia, las pruebas basadas en requerimientos son un enfoque sistemático al diseño de casos de prueba, donde se considera cada requerimiento y se deriva un conjunto de pruebas para éste. Las pruebas basadas en requerimientos son pruebas de validación más que de defecto: se intenta demostrar que el sistema implementó adecuadamente sus requerimientos.” (Sommerville, 2011, pág. 224)

Si se considera el último requisito no funcional mencionado en la sección 3.5 Requisitos no funcionales, cualquier operación realizada en la aplicación, no debe exceder de los 10 segundos.

Para validar este requisito se ha utilizado la aplicación Apache JMeter™ mencionada en el apartado 5.4.9 Apache JMeter™, introduciendo 5 hilos (usuarios) simultáneos (actualmente como máximo habrá 4 usuarios reales) con un intervalo de entrada de los mismos de 5 segundos (cada usuario entra a la aplicación 5 segundos más tarde que el anterior) y al cargar el listado de voluntarios que es el que más registros tiene en base de datos (.).

Ejecutando la prueba durante un minuto aproximadamente, como se puede visualizar en la Ilustración 43, el rendimiento está en torno a los 120 milisegundos, por lo que se da por validado el requisito no funcional:

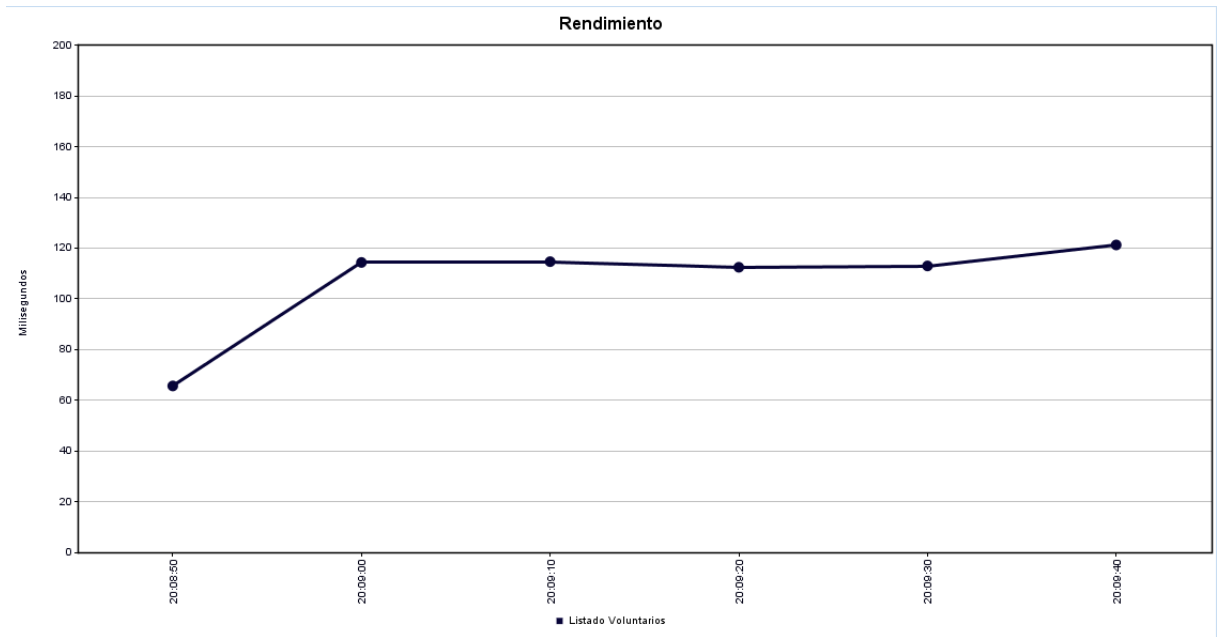


Ilustración 43. Rendimiento de la aplicación con 5 usuarios simultáneos.

En la Ilustración 44, se puede ver que se ha ejecutado un test con un número de muestras de 2474, obteniendo una media de 113ms, una mediana de 111ms y una desviación de 18ms, con un resultado global de 2543,349 peticiones por minuto:

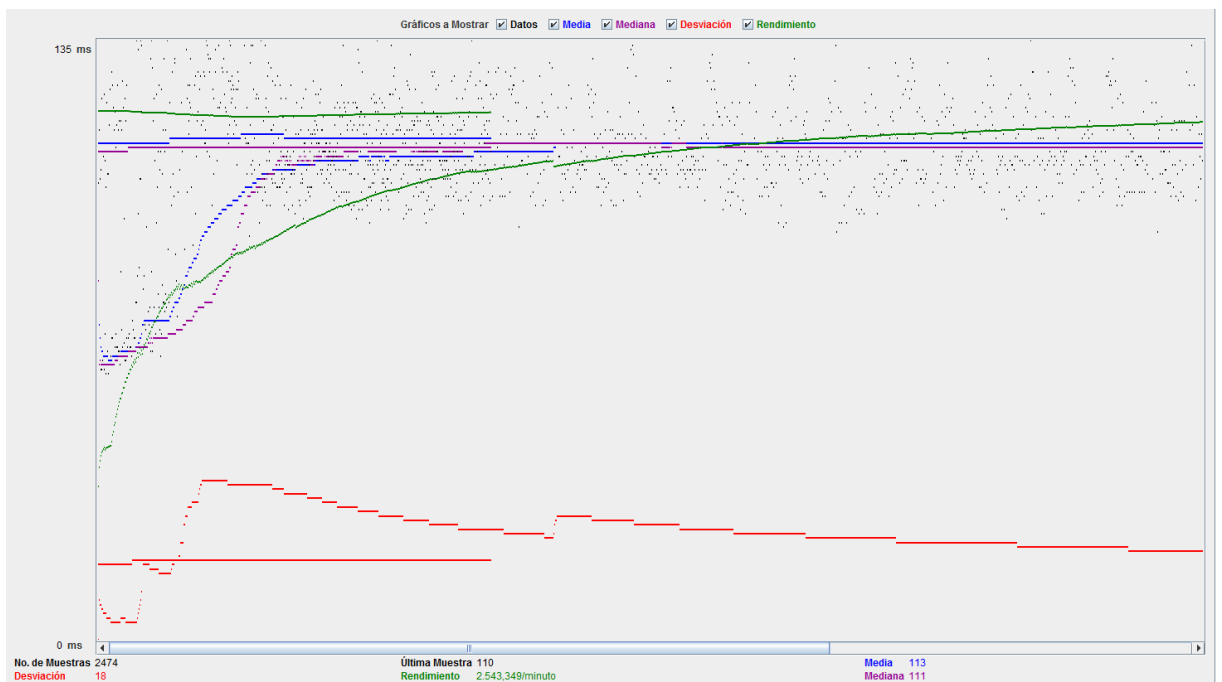


Ilustración 44. Media, Mediana, Desviación y Rendimiento de la aplicación con 5 hilos.

Por último, si se comparan los resultados añadiendo 5 hilos, 10 hilos y 20 hilos simultáneos, se observa que entre 5 y 10 usuarios se penaliza un poco el rendimiento, mientras que, a

partir de 10 usuarios, aunque el rendimiento disminuye como es lógico, la variación es menos significativa. Estos resultados se pueden ver en la Tabla 9:

Tabla 9. Tabla comparativa de los test de rendimiento con 5, 10 y 20 usuarios.

Etiqueta	# Muestras	Media(ms)	Mínimo(ms)	Máximo(ms)	Rendimiento(s)
Listado Voluntarios 5 usuarios	2974	184	14	681	3,59753
Listado Voluntarios 20 usuarios	974	548	66	689	17,48497
Listado Voluntarios 10 usuarios	977	546	15	677	17,54165

6.2. Usabilidad

La usabilidad puede medirse según varios criterios como son la velocidad de carga (ya analizada anteriormente), el tiempo que permanece el usuario en la página, la validación de los ficheros HTML y CSS o analizando la visibilidad del contenido teniendo en cuenta aspectos relevantes como el tamaño de letra o el contraste entre colores.

En la Ilustración 45 se puede ver cómo la aplicación desarrollada tiene el código HTML y el código CSS validado según las recomendaciones del W3C, mostrando sendos logos en la parte inferior de la aplicación como se aprecia en la Ilustración 13:

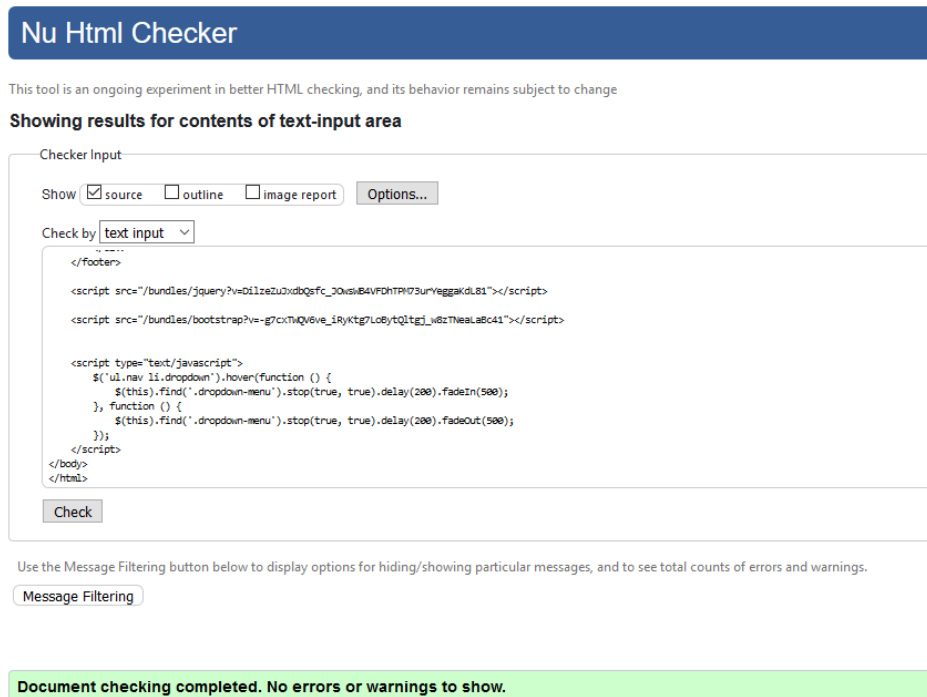


Ilustración 45. Validación correcta del código HTML.

6.3. Accesibilidad

En un documento publicado por el W3C, se define la accesibilidad web como un acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

Con esta idea nace la Iniciativa de Accesibilidad Web (WAI), cuyo objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad mejorando las herramientas para la evaluación y reparación de accesibilidad Web, llevando a cabo una labor educativa y de concienciación en relación a la importancia del diseño accesible de páginas Web.

Para ello, se define cada punto de verificación en función de unas pautas con tres niveles de prioridad diferentes:

- ✓ Prioridad 1: son aquellos puntos que un desarrollador Web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios no podrían acceder a la información del sitio Web.
- ✓ Prioridad 2: son aquellos puntos que un desarrollador Web debería cumplir ya que, si no fuese así, sería muy difícil acceder a la información para ciertos grupos de usuarios.
- ✓ Prioridad 3: son aquellos puntos que un desarrollador Web debería cumplir ya que, de otra forma, algunos usuarios experimentarían ciertas dificultades para acceder a la información. (W3C Consortium, 2013)

Y de acuerdo a estas prioridades, tres niveles de conformidad que indican en qué medida los puntos de verificación se satisfacen: “A” (prioridad 1), “Doble A” (prioridad 1 y 2) y “Triple A” (prioridad 1, 2 y 3).

Como se hacía referencia en el primer objetivo del punto Objetivos específicos, se debía verificar que la aplicación contaba con un nivel “Doble A” de accesibilidad web como mínimo.

Como demuestra la Ilustración 13 (mostrando el nivel de accesibilidad de la página), la aplicación tiene un nivel “Triple A”, satisfaciendo los puntos de verificación de prioridad 1, 2 y 3. La forma de conseguirlo ha sido introduciendo un texto alternativo para cada imagen del software, siguiendo el nivel de encabezados (h1,h2,h3,h4,h5 y h6) de HTML o introduciendo el tipo de script (“text-javascript”) al comienzo de cada uno de los scripts de la página, entre otras muchas pautas descritas en la iniciativa WAI mencionada anteriormente.

Para realizar dicha validación se ha utilizado la herramienta online “TAW”, descrita en la sección 5.4.10 TAW. Los resultados de la validación automática se pueden ver en la Ilustración 46:

t.a.w.
CTIC Centro Tecnológico

Resumen de resultados

Información del análisis

Recurso: [REDACTED]
Fecha: 04/09/2017 20:54
Pautas: WCAG 2.0
Nivel del análisis: AAA
Tecnologías: HTML, CSS

X 0 Problemas
detectados de forma automática

No hay problemas de carácter automático
Deben revisarse de forma manual las advertencias y los puntos no verificados para poder garantizar un nivel de Accesibilidad adecuado

Ilustración 46. Resumen de resultados de la validación de la accesibilidad web.

7. Conclusiones y trabajo futuro

En este último capítulo se hará un análisis del proyecto evaluando de qué forma se han cumplido los objetivos planteados en la sección 1.2 Objetivos y se plantearán los requisitos futuros que tendrá la aplicación y que se realizarán en la próxima versión (*release*) de la aplicación.

7.1. Conclusiones respecto a los objetivos iniciales

A continuación, se explicará la manera de consecución de los diferentes objetivos específicos planteados al principio del proyecto, que demuestran que se ha cumplido el objetivo principal con la realización del desarrollo.

Con el primer objetivo se pretendía analizar soluciones genéricas o específicas ya desarrolladas para organizaciones sin ánimo de lucro. Mediante el análisis realizado en la sección 2.2, se han estudiado soluciones similares analizando sus puntos fuertes y débiles que han servido como base para plantear los requisitos de esta aplicación en conjunto con las necesidades particulares de una organización como la Fundación Spínola Solidaria.

En el segundo objetivo, se quería conseguir una aplicación web usable y como mínimo un nivel de conformidad “Doble A” de accesibilidad web. Además, éste se ha cumplido obteniendo un nivel “Triple A” como se demuestra en el apartado 6.3 Accesibilidad.

Posteriormente, se pretendía ampliar los conocimientos de programación del desarrollador del proyecto en tecnologías Microsoft como ASP.NET, C# y SQL Server, buscando libros y documentación online acerca de estas tecnologías y preguntando en páginas de consulta expertas como “*StackOverflow*²²”, consiguiendo desarrollar una aplicación funcional y sencilla y permitiendo al creador del proyecto obtener un nuevo puesto de trabajo como consultor y desarrollador .NET para desarrollar proyectos con estas tecnologías.

El tercer objetivo a conseguir era el de utilizar herramientas de control de versiones para la gestión del código fuente. Para ello, se ha inicializado un proyecto de “*Github*”, se ha configurado en Microsoft Visual Studio y se han hecho “*commits*” y “*push*” en dicho repositorio cuando se terminaba una nueva funcionalidad de la aplicación. Puede verse qué es GIT en la sección 5.5 Implementación.

²² StackOverflow, pregunta relacionada con un problema con el framework “Entity Framework” https://stackoverflow.com/questions/45904879/insert-new-record-in-parents-table-instead-inserting-only-on-children/45905332?noredirect=1#comment78777331_45905332

Respecto al objetivo 4 de utilizar metodologías de gestión de proyectos estudiadas durante el grado y más concretamente metodologías ágiles, se ha conseguido mediante la utilización de SCRUM dividiendo el proyecto en 13 *sprints* y realizando un *product backlog* completo de los requisitos de la aplicación, tal como se explica en el apartado 1.4 Metodología de trabajo.

Siguiendo con el objetivo número 5 de analizar “*frameworks*”, “API”s o librerías que faciliten el desarrollo, se ha desarrollado el proyecto utilizando “*Bootstrap*” como “*framework*” de diseño y “*Jquery*” para simplificar el desarrollo de funcionalidades dinámicas e interactivas de la aplicación, se ha conectado a una REST API para obtener la lista de países en los desplegados de los formularios y múltiples librerías de terceros como “*Bootstrap select*” para dar estilos a los desplegados (*select*) de los formularios o “*JQuery UI Datepicker*” para mostrar un selector de fecha al usuario.

Otro de los objetivos que se pretendían conseguir es que fuera un software personalizable a otras organizaciones con las mismas necesidades. Al desarrollar una funcionalidad específica para modificar el logo del sistema mediante un formulario y configurar los desplegados de los menús en el módulo de configuración, esta necesidad queda satisfecha.

Al haber conseguido todos estos objetivos, se ha desarrollado una aplicación sencilla, intuitiva y útil para los empleados de la Fundación Spínola Solidaria, que ya están utilizando con un *feedback* muy positivo que les está ahorrando tiempo y simplificando las tareas a la hora de gestionar la información de los voluntarios, colaboradores, donaciones, sedes y permitiendo realizar un registro de la protección de los datos de acuerdo a la LOPD.

7.2. Trabajo futuro

Sin embargo, pese a la respuesta tan positiva que ha tenido el software en los usuarios finales, se proponen algunas mejoras que se enumeran a continuación para las siguientes versiones:

- ✓ Una de las funcionalidades que se salen del alcance de este proyecto pero que es necesaria es la gestión de correspondencia y comunicaciones para agilizar los procesos de comunicación de la Fundación con los colaboradores y voluntarios de la organización.
- ✓ Con este software se pretendía agilizar la gestión de la información de los voluntarios y colaboradores ya existentes. Sin embargo, otro de los propósitos de la organización es la de mejorar los procesos de captación de voluntarios y colaboradores y aumentar su número. Para ello, se quiere realizar una integración del software de gestión con la nueva página web de la organización mediante servicios web (webservices), debido a que la nueva página tendrá unas secciones

para que el visitante se una a la organización como colaborador (realizando una donación) y/o voluntario. Esta nueva página web seguramente estará realizada con el gestor de contenidos *Wordpress*, por lo que se contemplará esta necesidad una vez finalice el desarrollo del nuevo sitio web que aún no ha comenzado.

- ✓ Por último, otra de las necesidades detectada, aunque no de manera prioritaria, es la gestión de proyectos (como sí incluyen otras aplicaciones analizadas en la sección Soluciones existentes). De esta manera, gestionando la información de los mismos, se podría llevar un seguimiento más detallado del proyecto y generar un reporte similar a la memoria anual de proyectos que se realiza actualmente.

Bibliografía

- Ahmad, A. (23 de Octubre de 2016). *Entity Framework Database First*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/en-us/library/jj206878\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/jj206878(v=vs.113).aspx)
- Asociación SinergiaTIC. (s.f.). *¿Qué es?* Obtenido de SinergiaCRM: <http://sinergiacrm.org/es/>
- Bellostas Pérez-Grueso, A., Marcuello Servós, C., Moneva Abadía, J., & Marcuello Servós, C. (2007). *Capital social y organizaciones no lucrativas en España*. Bilbao: Fundación BBVA.
- Boletín Oficial del Estado (BOE). (8 de Julio de 1998). *Ley 23/1998, de 7 de julio, de Cooperación Internacional para el Desarrollo*. Recuperado el 20 de Mayo de 2017, de Boletín Oficial del Estado: <http://www.boe.es/boe/dias/1998/07/08/pdfs/A22755-22765.pdf>
- Burns, D. (2012). *Selenium 2 Testing Tools*. Birmingham: Packt Publishing Ltd.
- Ceballos Sierra, F. (2007). *Microsoft C#: lenguaje y aplicaciones*. Paracuellos del Jarama, Madrid: RA-MA Editorial.
- CECA. (2015). *Órdenes en formato ISO 20022* . Recuperado el 2017 de Junio de 12, de CECA: <http://www.ceca.es/wp-content/themes/ceca/assets/Cuaderno%20XML%20SCT%20Noviembre%202015.pdf>
- Chaffer, J., & Swedberg, K. (2011). *Learning JQuery*. Birmingham: Packt Publishing Ltd.
- CIVICRM LLC. (2005). *About*. Obtenido de CIVICRM: <https://civicrm.org/about>
- Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Boston, Massachusetts, Estados Unidos de América: Addison-Wesley Professional. Recuperado el 2 de Mayo de 2017, de MARTINFOWLER.COM: <https://martinfowler.com/eaCatalog/dataTransferObject.html>
- González Marcos, A., Alba Elías, F., & Ordieres Meré, J. (2014). *Ingeniería de proyectos*. Madrid: DEXTRAL EDITORIAL.
- Guérin, B.-A., & Piqueres Juan, F. J. (2016). *ASP.NET en C# con Visual Studio 2015. Diseño y desarrollo de aplicaciones Web*. Barcelona: Ediciones ENI.

- Harris, A. (2009). *Programación con PHP 6 y MySQL*. Madrid: EDICIONES ANAYA MULTIMEDIA.
- Jefatura del Estado. (14 de Diciembre de 1999). *Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos*. Obtenido de Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos : <http://www.boe.es/buscar/pdf/1999/BOE-A-1999-23750-consolidado.pdf>
- Jiménez Capel, M. (2014). *Bases de datos relacionales y modelado de datos*. Antequera, Málaga: IC Editorial.
- JSON.ORG. (s.f.). *Introducción a JSON*. Recuperado el 2 de Mayo de 2017, de JSON: <http://www.json.org/json-es.html>
- Kehoe, D. (2016). *Learn Ruby on Rails: Book One*.
- Microsoft Corporation. (2005). *Referencia de lenguajes de SQL Server*. Recuperado el 2 de Mayo de 2017, de Microsoft TechNet: [https://technet.microsoft.com/es-es/library/ms166026\(v=sql.90\).aspx](https://technet.microsoft.com/es-es/library/ms166026(v=sql.90).aspx)
- Microsoft Corporation. (23 de Octubre de 2016). *Información general de Entity Framework*. Recuperado el 2017 de Junio de 12, de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx)
- Microsoft Corporation. (2016). *Novedades de SQL Server 2016*. Obtenido de Microsoft Developer Network: <https://msdn.microsoft.com/es-es/library/bb500435.aspx>
- Millett, S. (2010). *ASP.NET Design Patterns*. Indianapolis, IN: Wiley Publishing Inc.
- Murray, P., & P. Shaughnessy, B. (2011). *Using CiviCRM*. BIRMINGHAM - MUMBAI: Packt Publishing Ltd.
- ONG, I. g. (2017). *Informe global sobre tecnología en línea de ONG 2017*. (Y. P. Registry, Ed.) Obtenido de Informe global sobre tecnología en línea de ONG 2017: <http://www.techreport.ngo/>
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (Seventh Edition ed.). New York, NY, United States of America: Mc Graw Hill Higher Education.
- Quigley, R. (28 de Abril de 2012). *The day Steve Jobs phoned the Pope as a prank: Watch the 'most revealing' interview with the Apple boss that lay unseen in a garage for 15 years*. Recuperado el 25 de Abril de 2017, de Daily Mail UK:

<http://www.dailymail.co.uk/news/article-2136492/The-revealing-A-lost-interview-Steve-Jobs-offers-rare-insight-childhood-memories-thought-science-liberal-art.html>

Schildt, H. (2010). *C# 3.0*. México, D.F.: McGrawHill Editorial.

Schuldt, H. (2009). *Multi-Tier Architecture[Arquitectura multi-nivel]*. Springer US.

semilla de SOFTWARE LIBRE. (2017). *GONG*. Obtenido de GONG: <http://gong.es/>

Somasundaram, R. (2013). *Git: Version Control for Everyone*. Birmingham: Packt Publishing Ltd.

Somerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson Educación S.A.

Sommerville, I. (2011). *Ingeniería de Software*. México: Pearson Educación.

The Apache Software Foundation. (2017). *Apache JMeter™*. Obtenido de Apache JMeter™: <http://jmeter.apache.org/>

Tigris.org. (2009). *Welcome to ArgoUML*. Recuperado el 2017 de Junio de 12, de Tigris.org Open Source Software Engineering Tools: <http://argouml.tigris.org/>

Viscardi, S. (2013). *The Professional ScrumMaster's Handbook*. Birmingham: Packt Publishing Ltd.

W3C Consortium. (2013). *Guía Breve de Accesibilidad Web*. Obtenido de W3C España: <http://www.w3c.es/Divulgacion/GuiasBreves/Accesibilidad>

W3C Consortium. (2017). *Web Accessibility Initiative*. Recuperado el 17 de Abril de 2017, de Web Accessibility Initiative (WAI): <https://www.w3.org/WAI/>

W3Schools. (s.f.). *XML Tutorial*. Recuperado el 2 de Mayo de 2017, de W3schools.com: <https://www.w3schools.com/xml/>

Wagner, B., Yishengjin1413, & Cai, S. (24 de Marzo de 2017). *Espacios de nombres (Guía de programación de C#)*. Obtenido de Microsoft .NET: <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/namespaces/index>

Zavala, L. (1994). *Las ONGD en España: origen y evolución*. Madrid.

Listado de acrónimos

A	
AD	
Active Directory	31
AECID	
Agencia Española de Cooperación Internacional para el Desarrollo	25
API	
Application programming interface	14
Application Programming Interface ..	2, 3, 25, 26, 30, 31, 34, 69, 83
B	
BOE	
Boletín Oficial del Estado.....	22
C	
CECA	
Confederación Española de Cajas de Ahorros..	13
CMS	
Content Management System.....	24
CRM	
Customer Relationship Management....	9, 24, 25, 26, 27, 28, 29, 30
CRUD	
Create, Read, Update, Delete.....	65
CTIC	
Centro Tecnológico.....	71
D	
DTO	
Data Transfer Object u Objeto de Transferencia de Datos.	63
E	
edmx	
Entity Data Model.....	73
EUPL	
European Union Public License	25
I	
IIS	
Internet Information Services	2, 3, 28, 29, 31, 39
IRPF	
Impuesto sobre la Renta de las Personas Físicas.	22
J	
Joomla	
Gestor de contenidos web basado en tecnologías PHP y MySQL y el modelo MVC para creación de sitios web.	22
L	
LAN	
Local Area Network o Red de Área Local.	39
LOPD	
Ley Orgánica de Protección de Datos de Carácter Personal..	6, 9, 11, 13, 14, 15, 22, 38, 40, 44, 45, 83
M	
MVC	
Modelo-Vista-Controlador	2, 3, 10, 45, 72
Patrón arquitectónico Modelo Vista Controlador.....	62, 63, 66, 70
O	
ONG	
Organización no gubernamental	9, 18, 22, 24, 29, 36, 39
ONGd	
Organización sin ánimo de lucro para el desarrollo.....	21
ONGD	
Organización sin ánimo de lucro para el desarrollo	2, 12, 19, 21, 22, 25, 33
ORM	

Object Relational Mapping	69	U	
P		UML	
PBI		Unified Modeling Language	42, 64, 70
Product Backlog Item	17, 32, 46	UNIR	
S		Universidad internacional de La Rioja	1
SEPA		W	
SEPA mejora la eficiencia de los pagos en Europa mediante el desarrollo de instrumentos, estándares y reglas comunes.	28	W3C	
Single Euro Payments Area 12, 13, 22, 27, 31, 43, 53, 69		Consorcio W3C	45, 79, 80
T		World Wide Web Consortium	65
TFG		X	
Trabajo de fin de grado	21	XML	
		eXtensible Markup Language	36, 56, 69
		Extensible Markup Language	7, 12, 15, 22, 53