

Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

Grado en Ingeniería Informática

SCRUM Wizard

Página web del proyecto:

<https://github.com/angelfdz/TFG.git>

Trabajo Fin de Grado

Presentado por: Fernández Martín, Ángel

Director/a: de la Fuente Valentín, Luis

Ciudad: Madrid

Fecha: 18 Julio 2015

Resumen

SCRUM es una metodología ágil donde los proyectos se ejecutan en iteraciones de bloques temporales (sprints) típicamente de un mes natural que, producen un resultado completo que podrá ser entregado al cliente cuando se decida. En SCRUM el centro es el cliente y su necesidad, y el equipo de trabajo concienciado de ello y autogestionado se encarga de asegurar los objetivos planteados. Lógicamente se requieren herramientas de software de apoyo que soporten la metodología y permitan a los integrantes del equipo centrarse en las labores de más valor añadido: la realización del proyecto.

La aplicación que se ha definido, diseñado y construido en este Trabajo Fin de Grado cumple este objetivo. Evitando la pérdida de información, poniendo al alcance de todos los miembros del equipo de forma sencilla y rápida la misma y simplificando la gestión de las reuniones.

SCRUM, Metodologías, Herramienta, Desarrollo, Ágil

Abstract

SCRUM is an agile methodology in which projects are executed through iterations of incremental time-boxes (sprints). These sprints usually take place during a natural month-period and whose result that can be provided to a client whenever he so wishes. In SCRUM, the centre is the client and his needs. The work team, taking this into account and through self-management, is tasked with successfully reaching the previously-laid down goals. Naturally, support software tools are needed. These have to be able to support the methodology and allow the team members to focus on the value-added tasks: The fulfilment of the project.

The application that has been incepted, designed and built in these Bachelor Paper meets these criteria. It avoids information loss through the simplification of the running of meetings by ensuring that the members of meetings have an easier, faster access to the information at hand.

SCRUM, Agile, Software, Development, Methodology

Contenido

Resumen.....	2
Abstract.....	2
Índice de Ilustraciones.....	5
Índice de Tablas.....	7
1. Introducción	8
2. Objetivo.....	10
3. Contexto y Estudio Preliminar	11
4. Metodología Ágil SCRUM	14
5. Identificación de requisitos	17
5.1. Funcionalidad de la aplicación	17
5.2. Diagrama de Casos de Uso	19
5.3. Metodología empleada	21
6. Descripción del Proyecto.....	24
6.1. Autenticación de usuario.....	24
6.2. El usuario “Administrador” en la aplicación	25
6.3. ¿Qué puede hacer el PO en la aplicación?	28
6.3.1. Trabajando con el PB.....	29
6.3.2. Trabajando con Proyectos.....	30
6.3.3. Cancelación de Sprint	32
6.3.4. Información de Seguimiento de Proyecto	32
6.3.5. Notificaciones.....	33
6.4. ¿Qué puede hacer el SM en la aplicación?	34
6.4.1. Gestionando convocatorias	34
6.4.2. Activar Sprint.....	36
6.4.3. Notificaciones.....	36
6.5. ¿Qué puede hacer el DT en la aplicación?	36
6.5.1. Gestionando Sprints.....	37

6.5.2. Gestionando el Sprint Backlog	40
6.5.3. Composición de Equipo de Proyecto	42
6.5.4. Notificaciones	42
7. Descripción Técnica	43
7.1. Aplicación – Visual Studio 2012 Express para aplicaciones de escritorio	43
7.1.1. Cómo se ha usado Visual Studio 2012 Express	45
7.2. Elementos propios	45
7.3. Componentes .NET Framework 4.5	51
7.4. Componente OleDb (Microsoft Jet OleDb 4.0)	56
7.5. Fichero de configuración “app.config”	59
7.6. SGDB – MS Access (mdb)	61
7.7. Tratamiento de Errores en “Scrum Wizard”	62
8. Evaluación	64
9. Conclusión y Trabajos Futuros	66
Anexo I: Procedimiento de Instalación	67
Anexo II: Detalle de las tablas de la aplicación	68
Referencias y enlaces	72

Índice de Ilustraciones

Ilustración 1: Metodologías Ágiles	8
Ilustración 2: El usuario... el centro	8
Ilustración 3: Logo de Scrum Wizard	10
Ilustración 4: Proceso de Scrum	16
Ilustración 5: Diagrama de CUs	19
Ilustración 6: Hitos del TFG	21
Ilustración 7: Autenticación	24
Ilustración 8: Fallo de Autenticación	24
Ilustración 9: Lista de Proyectos	25
Ilustración 10: Usuario sin Proyectos	25
Ilustración 11: Opciones de Menú del Administrador	25
Ilustración 12: Cambio de usuario y contraseña de la cuenta Administrador	26
Ilustración 13: Administración de Usuarios	26
Ilustración 14: Administración de Equipos	27
Ilustración 15: Incluyendo Usuarios en Equipo	28
Ilustración 16: Pantalla de presentación del rol PO	28
Ilustración 17: Estructura del PB	29
Ilustración 18: Tareas del PO relacionadas con Proyectos	30
Ilustración 19: Creación de un Nuevo Proyecto	30
Ilustración 20: Liberación de Incremento a Cliente	31
Ilustración 21: Cancelación de Sprint	32
Ilustración 22: Información de Seguimiento de Proyecto	33
Ilustración 23: Notificaciones	33
Ilustración 24: Pantalla de Presentación Rol SM	34
Ilustración 25: Opciones de Convocatorias	35
Ilustración 26: Envío de Convocatoria de Daily Scrum	35
Ilustración 27: Pantalla de Presentación del Rol DTM	37
Ilustración 28: Opciones de un Sprint	37
Ilustración 29: Selección de Sprint	38
Ilustración 30: Objetivo de Sprint	38
Ilustración 31: Fijar Duración Sprint	38
Ilustración 32: Aviso en cambio de la duración de Sprint	39
Ilustración 33: Información de Seguimiento de Sprint	40
Ilustración 34: Detalle del SB	41
Ilustración 35: Aviso de discrepancia entre duración en días y en horas	42

Ilustración 36: Arquitectura de Componentes	43
Ilustración 37: Carátula de arranque VS 2012	43
Ilustración 38: Elementos Propios Creados	46
Ilustración 39: Elemento DSConexion.xsd	46
Ilustración 40: Componentes Formulario PO	49
Ilustración 41: Formulario en tiempo de ejecución	51
Ilustración 42: Controles Disponibles	52
Ilustración 43: Controles Comunes	52
Ilustración 44: Componentes de Contenedores	53
Ilustración 45: Menús y barras de herramientas	53
Ilustración 46: Componentes de Datos	54
Ilustración 47: Componentes Sencillos	54
Ilustración 48: Componentes de Impresión	54
Ilustración 49: Componentes Cuadros de Diálogo	54
Ilustración 50: Componentes de Interoperabilidad	54
Ilustración 51: Componentes PowerPacks	54
Ilustración 52: Ejemplo de Control Botón "Salir"	55
Ilustración 53: Cadena de Conexión a SGBD	56
Ilustración 54: Elementos Clase OleDbCommand	58
Ilustración 55: Modelo de Datos de la solución	62
Ilustración 56: Cronograma TFG	64
Ilustración 57: Estructura Típica de una Fase	64

Índice de Tablas

Tabla 1: Scrum vs Scrumban	13
Tabla 2: Funcionalidades de la Aplicación.....	19
Tabla 3: Actores de la Aplicación	20
Tabla 4: Detalle Casos de Uso	21
Tabla 5: Requisitos no funcionales.....	21
Tabla 6: Eventos de Scrum	35
Tabla 7: Detalle de la Tabla Equipos.....	68
Tabla 8: Detalle de la Tabla Roles.....	68
Tabla 9: Detalle de la Tabla Usuarios.....	68
Tabla 10: Detalle de la Tabla Notificaciones.....	69
Tabla 11: Detalle de la Tabla Sprint Backlog	69
Tabla 12: Detalle de la Tabla Product Backlog	69
Tabla 13: Detalle de la Tabla Sprints.....	70
Tabla 14: Detalle de la Tabla Proyectos	70
Tabla 15: Detalle de la Tabla Releases	71

1. Introducción

Las metodologías ágiles son una realidad que se vienen empleando desde tiempo atrás y que cada día son de más utilización; fruto, probablemente, de la transformación digital que está teniendo lugar en el conjunto de la sociedad y que obliga a cambios muy rápidos y muy ágiles en las aplicaciones y soluciones informáticas que se utilizan.

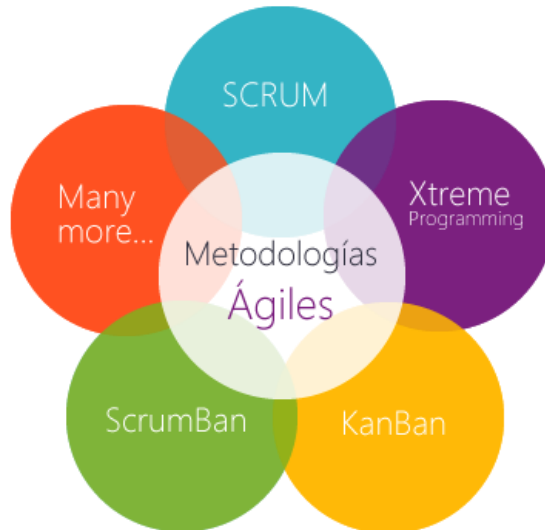


Ilustración 1: Metodologías Ágiles

Además, ahora, cualquier usuario influye en las aplicaciones que se construyen. Inciden a través de sus opiniones en las redes sociales o entre sus conocidos y amigos. El “boca a oreja” cada vez es más viral y surte más efecto. Las compañías se han visto obligadas a centrarse más y más en esta experiencia de usuario para mejorar sus soluciones en la línea demandada. Adaptarse o desaparecer.



Ilustración 2: El usuario... el centro

Con las metodologías tradicionales, en general más encorsetadas por el método y el procedimiento, se hace muy complicado alcanzar las cotas de agilidad demandadas y, los equipos de trabajo no están tan involucrados y concienciados de la importancia de su trabajo.

Las metodologías ágiles y, Scrum, como representante de las mismas, vienen a poner el foco en la agilidad, la iteración, la participación del usuario y la importancia del equipo de desarrollo como pieza fundamental en la construcción de la solución. Acercando la necesidad del usuario (conforme la siente y la expresa) al desarrollador que finalmente acaba implementándola. Reduciendo la enorme distancia que en metodologías tradicionales existe entre estas dos piezas clave de cualquier proyecto.

La aplicación eficiente y eficaz de Scrum se ve fuertemente reforzada mediante la utilización de herramientas que ayuden y guíen a los equipos a través de la metodología a lo largo de los proyectos que se abordan mediante ella.

Este Trabajo Fin de Grado ha producido una herramienta informática para uso en escritorio que podrá ser empleada por los equipos de Scrum en el desarrollo de los proyectos que acometan.

Para lograrlo se ha apoyado en la documentación que existe y que describe cómo abordar un proyecto usando Scrum, si bien, hay cuestiones prácticas que quedan abiertas y que han sido explicitadas por el autor de este Trabajo Fin de Grado y que se detallan a lo largo de la presente memoria.

Habrán aspectos que se han identificado, pero que quedarán para futuras evoluciones de la herramienta aquí presentada. Principalmente aspectos menos funcionales y más centrados en el contexto de la aplicación.

2. Objetivo

Este Trabajo Fin de Grado tiene como **objetivo** generar una aplicación de escritorio que sirva de ayuda y apoyo a cualquier miembro de un equipo de Scrum en la realización de las tareas que Scrum le asigna y encomienda.



Ilustración 3: Logo de Scrum Wizard

La aplicación construida:

- Guía al usuario que se conecta, en función de su rol en el proyecto, a lo largo de las etapas de la metodología ágil Scrum y de las actividades que, en cada momento de la metodología, tiene asociado
- Almacena la información generada durante el proceso. Notificando a los participantes de los eventos en los que tienen que intervenir
- Guarda los proyectos en los que se utiliza, de manera independiente de forma que puedan configurarse, tratarse y recuperarse de forma individualizada
- Además, y aunque Scrum no lo especifica, proporciona un rol denominado “admin” que permite la realización de tareas más burocráticas pero necesarias para su buen funcionamiento

Obviamente un objetivo de estas características se ve también constreñido por el marco temporal para su realización. Por este motivo, al final de la presente memoria se incluirán elementos de evolución que, de abordarse, pueden aportar y mejorar la aplicación construida.

3. Contexto y Estudio Preliminar

Ya por el año 2001 se consideraba que metodologías formales como CMM-SW (hoy CMMi), PMI, SPICE (ISO 15504),... eran excesivamente pesadas y rígidas por lo exigente de su normativa y por su gran dependencia de planificaciones muy detalladas, antes incluso de haber comenzado el desarrollo.

En Marzo de ese año un nutrido grupo de expertos se reunieron para tratar de buscar alternativas a esta situación. Fruto de esa reunión surgió el término “Métodos Ágiles” para referirse a las nuevas formas de trabajo que estaban surgiendo, como alternativa a las citadas, tratando de superar los problemas que planteaban.

El resultado se concretó en 4 valores y 12 principios que quedaron recogidos en el “Manifiesto Ágil” (Agile Manifesto) que se enumeran a continuación:

Valores

1. Se considera **más importantes** y se da, por tanto más valor, a los **individuos e interacciones** que a los procesos y a las herramientas
2. Se da **más importancia** al **software funcionando** que a disponer de una exhaustiva documentación
3. Se busca y **potencia trabajar colaborativamente con el cliente** en lugar de un acuerdo contractual
4. Es mucho **más importante ser capaz de adaptarse y reaccionar ante los cambios** que seguir al pie de la letra una planificación

Principios

1. La principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se adaptan al cambio como ventaja competitiva para el cliente
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos cortos
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto

5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara
7. El software que funciona es la principal medida del progreso
8. Los procesos ágiles promueven el desarrollo sostenido. Los sponsors, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica promueve la agilidad
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto-organizan
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia

Hoy en día, el paso del tiempo ha reforzado esta línea de pensamiento: el nuevo paradigma social y tecnológico, la generación millennial, para la que: Internet, el móvil, los medios sociales... son herramientas naturales para su día a día: compran, hacen transacciones, comparten productos y servicios: es su forma de vida. Y sobre todo, para las nuevas generaciones para las que aún ni podemos imaginar cómo pensarán y actuarán.

Todo ello abunda en la necesidad de dar respuesta rápida, fiable y ágil a las necesidades de la sociedad - que además son muy volátiles – lo que refuerza aún más si cabe estos métodos ágiles.

Muchas metodologías ágiles han surgido dando respuesta a los 4 valores y 12 principios del Manifiesto Ágil. Las más exitosas:

- **Scrum**: donde los proyectos se ejecutan en iteraciones de bloques temporales (sprints) típicamente de un mes natural que producen un resultado completo que podrá ser entregado al cliente cuando se decida
- **XP** (eXtreme Programming): se basa en 4 principios: simplicidad, comunicación, retroalimentación y valor. Se orienta a las pruebas y a la refactorización, de forma

que se diseñan e implementan las pruebas antes de la construcción de la funcionalidad. Típicamente organizado entre 10 y 15 ciclos o iteraciones

- **Kanban:** cuyo objetivo es gestionar de manera general cómo se van contemplando las tareas, aunque en los últimos años se ha empleado en la gestión de proyectos de desarrollo de Software. Las principales reglas son: a) Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo; b) Determinar el límite del WIP (Work in Progress – Trabajo en curso); c) Medir el tiempo en completar una tarea (Lead Time)
- **Scrumban:** es una metodología derivada de los métodos de desarrollo Scrum y Kanban. La tabla que sigue compara Scrumban con Scrum

Comparativa Scrum vs Scrumban		
Elemento	Scrum	Scrumban
Pizarra / Herramientas	Pizarra Backlogs Gráficos	Pizarra
Reuniones	Reunión diaria Planificación Revisión Retrospectiva	Reunión diaria
Iteraciones	Sí. Sprints	No. Flujo Continuo
Estimaciones	Sí	No
Equipo	Multidisciplinar	Puede ser especializado
Roles	Product Owner Scrum Master Equipo	Equipo y otros
WIP	En el Sprint	En la tarea
Cambios	Al siguiente Sprint	Al “TO DO”
Problemas	Se solucionan	Se evitan

Tabla 1: Scrum vs Scrumban

Todas tienen sus especificidades, sus puntos fuertes y sus puntos menos fuertes. Es posible la combinación de ellas para potenciar aún más si cabe el método finalmente elegido.

Este Trabajo Fin de Grado se ha inclinado por **Scrum**, debido principalmente a la versatilidad del mismo, su adecuación a la realidad actual en multitud de empresas y su escalabilidad. El siguiente apartado analiza en detalle esta metodología.

4. Metodología Ágil SCRUM

Scrum¹, se viene empleando desde los años 90. Acorde a la Guía de Scrum (Sutherland & Schwaber, 2013), “es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente”.

Scrum se apoya en el empirismo, esto es, la experiencia es la pieza clave, hace que la toma de decisiones se base en lo que hemos aprendido y conocemos. Los tres ejes sobre los que pivota Scrum son: la transparencia, la inspección y la adaptación.

Scrum define²:

- **Roles**

Tres tipos de actores participan en Scrum:

- Product Owner (en adelante PO) o dueño del producto, es una persona no un comité, “es el responsable de maximizar el valor del producto y del trabajo del Development Team. Es la única persona responsable de gestionar el Product BackLog”. Es el único que puede modificar el alcance de un Sprint Backlog, así como cancelar un Sprint y es el encargado de generar los informes de situación y de proyección necesarios. Decide si un incremento es entregado o no al cliente
- Scrum Master (en adelante SM) vela porque todos los participantes en el proyecto asimilen y usen adecuadamente las teorías, prácticas y reglas de Scrum, asegurándose que los distintos actores acudan y se involucren en los distintos eventos. Realiza distintas tareas en función de quién sea su interlocutor.
- Development Team Member (en adelante DTM) o miembros del equipo de desarrollo, son los profesionales que trabajan en la generación de un incremento de producto “terminado”. Actualizan la situación del Sprint Backlog en función de los resultados del Daily Scrum
- Development Team (en adelante DT), compuesto por todos los DTM. Idealmente con más de 3 componentes y menos de 9

Todos ellos conforman el denominado Scrum Team (en adelante ST) o equipo de Scrum. Los STs, obligatoriamente están compuestos por 1 PO, 1 SM y, al menos, 1

¹ El término Scrum es un tipo de jugada en el deporte del rugby. Es lo que en español llamamos melé

² Las frases entrecomilladas son extractos de La Guía de Scrum (Sutherland & Schwaber, 2013)

DT. Por tanto el tamaño máximo óptimo irá desde los 5 hasta los 11 miembros. De esta forma se garantiza la agilidad y la competencia del DT.

- **Artefactos**

Scrum define cinco artefactos:

- Product Backlog (en adelante PB) o lista de producto, es una lista ordenada y priorizada de todas las necesidades del proyecto (denominadas **historias**). Es una lista que va cambiando a lo largo del proyecto. Empieza a alto nivel y poco a poco se va detallando. Cada elemento del PB, al menos, incluirá: identificador, descripción, prioridad, estimación de esfuerzo de realización – en días y en coste –, valor que aporta, estado y el sprint en el que se aborda. El dueño del PB es el PO, siendo el único que puede crear o refinar el PB
- Seguimiento del progreso hacia un objetivo, el PO en cualquier momento podrá calcular el trabajo restante para lograr el objetivo
- Sprint Backlog (en adelante SB) o lista de pendientes del sprint; existe un SB por cada sprint y al inicio del mismo (es un resultado que se obtiene del Sprint Planning Meeting) está formado por aquellos elementos del PB que serán realizados en el Sprint en curso. El SB se puede modificar por el DT
- Seguimiento del proceso del sprint, el DT en cualquier momento podrá calcular el trabajo restante para lograr el objetivo
- Incremento, “es la suma de todos los elementos del PB completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la definición de “Terminado” del ST. El incremento debe estar en condiciones de utilizarse sin importar si el PO decide liberarlo o no. Una Release o entrega a cliente, es un incremento que el PO decide entregar al cliente.

- **Eventos Formales**

El sprint es el elemento alrededor del cual se articula Scrum. Es un periodo de tiempo de, aproximadamente, un mes en el que se realiza un incremento de producto “terminado” (la definición de terminado debe compartirse entre todos los miembros del equipo, es decir, cuando se indica terminado debe significar para todos lo mismo). La duración de un sprint, una vez fijada, no se puede cambiar.

Existen cuatro eventos formales:

- **Sprint Planning Meeting** (en adelante SPM) o reunión de planificación del sprint, es una reunión, con una duración proporcional a la duración del sprint (para un sprint de 1 mes durará aproximadamente 8 horas), que se produce antes de arrancar cualquier sprint y donde se planifica qué se va a hacer en dicho sprint y cómo se va a lograr. En ella participa todo el ST y se fija el objetivo del sprint así como el SB
- **Daily Scrum** (en adelante DS) o Scrum diario, es una reunión diaria del DT de una duración máxima de 15 minutos, donde se abordan tres preguntas: a) “qué hice ayer que ayudó al DT a lograr el Objetivo del Sprint?”; b) “qué haré hoy para ayudar al DT a lograr el Objetivo del Sprint?” y c) “veo algún impedimento que evite que el DT o yo logremos el Objetivo del Sprint?”
- **Sprint Review** (en adelante SR) o revisión del sprint, se trata de una reunión que se produce a la finalización del sprint, en la que participa el ST, con una duración máxima de 4 horas para sprints de un mes. En esta reunión se inspecciona el incremento y se modifica el PB si se considera necesario
- **Sprint Retrospective** (en adelante SRP) o retrospectiva del Sprint, está reunión se produce tras el SR y antes del SPM. Tiene una duración máxima de 3 horas para sprints de 1 mes y participa el ST. El objetivo es el autoanálisis del equipo y extraer las mejoras aplicables a siguientes sprints.

El proceso de Scrum queda representado seguidamente. Cabe reseñar que habitualmente se implementa un Sprint denominado 0 donde se pone en marcha el proyecto y se establece el PB inicial.

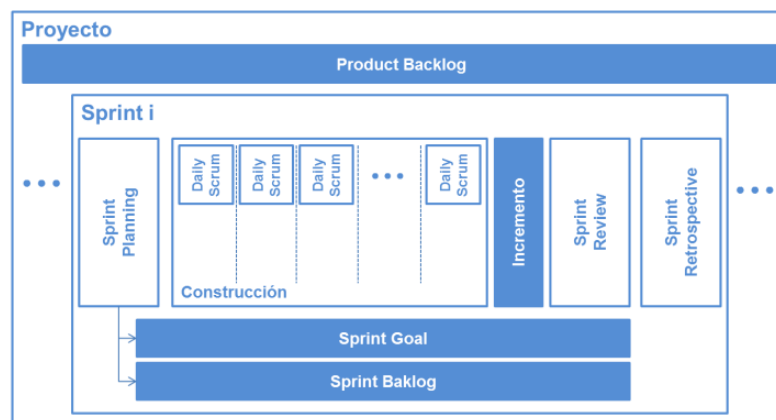


Ilustración 4: Proceso de Scrum

5. Identificación de requisitos

El establecimiento de los requerimientos que reúne la aplicación construida en este Trabajo Fin de Grado se ha cimentado mayoritariamente en el documento La Guía Scrum (Sutherland & Schwaber, 2013), y en menor medida en la consulta en internet de otros sitios web:

- Página web de Scrum.Org, (Scrum.org)
- Página web de Scrum Alliance, (Scrum Alliance)
- Página de Metodología Scrum, (Scrum Methodology)

Otra parte de ellos han surgido del enfoque iterativo metodológico aplicado, pues a medida que se iba construyendo la aplicación fueron apareciendo aspectos eminentemente prácticos que requerían un enfoque de solución que se han conformado como otros requisitos para la aplicación.

El nombre de la aplicación es “SCRUM Wizard”, puesto que se trata de una aplicación de escritorio que permite, a los distintos roles que la usan, realizar de forma sencilla y guiada todas las tareas y labores que determina Scrum para ese rol.

5.1. Funcionalidad de la aplicación

Desde la perspectiva de la aplicación existen cuatro roles bien diferenciados. Tres de ellos coinciden exactamente con los que establece Scrum, esto es: PO, SM y DTM. El cuarto rol es un perfil puramente Administrativo, denominado “admin” (como identificador de “Administrador”).

El usuario “admin” es el único que vendrá cargado en origen en la aplicación. A partir de este usuario es cómo se debería empezar a utilizar la aplicación para realizar todas las tareas más administrativas y burocráticas, por ejemplo, dar de alta usuarios o bien actualizar los datos de usuario. La clave inicial de este usuario coincide con su nombre de usuario.

Seguidamente se incluye la lista de las funcionalidades a las que da respuesta la aplicación. Se presenta en formato de tabla, donde para cada una de las funcionalidades identificadas se detalla:

- **Id.:** Se trata de un identificador único que se asigna a cada funcionalidad con el objetivo de simplificar la identificación y posterior referencia
- **Descripción:** Donde se recoge la descripción de alto nivel de la funcionalidad que incorpora la aplicación

- **Quién?:** En este campo se indica, para cada funcionalidad, qué rol o roles pueden realizar esta función. Los roles no mencionados no tendrán acceso a esta función dentro de la aplicación. A efectos de identificación se usarán los términos “PO”, “SM”, “DTM” y “admin” con los significados ya explicados previamente en este documento. Cuando todos los usuarios dispongan de esta funcionalidad indicaremos “todos”, y cuando sean los tres roles que indica Scrum usaremos “ST”

Cuando se use el término “mantenimiento” se estará refiriendo al alta, modificación y borrado.

En ocasiones Scrum permite delegar funciones en componentes del ST. La aplicación no contemplará esta situación. Se propone como una línea evolutiva de futuro.

Los requisitos funcionales identificados son:

Funcionalidades		
Id	Descripción	Quién?
1	Acceso a la aplicación mediante usuario y contraseña	Todos
2	Mantenimiento de usuarios	Admin
3	Un mismo usuario puede tener distinto rol en función del proyecto	ST
4	Modificación de datos del administrador de la herramienta	Admin
5	Creación del PB	PO
6	Refinamiento del PB	PO
7	Mantenimiento de DT, conforme a las reglas establecidas por Scrum	Admin
8	Mantenimiento de Proyectos (los proyectos no podrán ser eliminados). Se aplican los criterios que establece Scrum para la aceptación de los datos que se introduzcan, por ejemplo, duración máxima de los Sprints	PO
9	Cancelación de proyecto	PO
10	Terminación de proyecto	PO
11	Visualizar los integrantes de un equipo	ST
12	Cancelación de Sprint	PO
13	Generación de la información de Seguimiento de progreso hacia el objetivo	PO
14	Generar “Release” a entregar a cliente	PO
15	Convocar a actores involucrados al SPM	SM
16	Convocar a actores involucrados al SR	SM
17	Convocar a actores involucrados al SRP	SM
18	Convocar a actores al DS	SM
19	Establecer el objetivo del sprint	DT

Funcionalidades		
Id	Descripción	Quién?
20	Mantenimiento del SB (los SB no podrán eliminarse) <ul style="list-style-type: none">• Incluye la posibilidad de replanificación dentro del sprint• Cambio de alcance• También permite asignar historias a DTM	DT
21	Establecer la duración del Sprint	DT
22	Generación de la información de Seguimiento de progreso del Sprint	DT
23	Cierre de Sprint. Acto “formal” y explícito. No solo a través de la terminación de todas las labores del sprint	DT
24	Acceso a las notificaciones asignadas	ST
25	Activación de Sprint, permite saber cuál es el sprint activo	SM

Tabla 2: Funcionalidades de la Aplicación

5.2. Diagrama de Casos de Uso

Según (Schmuller, 2001) un caso de uso, en UML, es “una descripción de las acciones de un sistema desde el punto de vista del usuario del mismo”

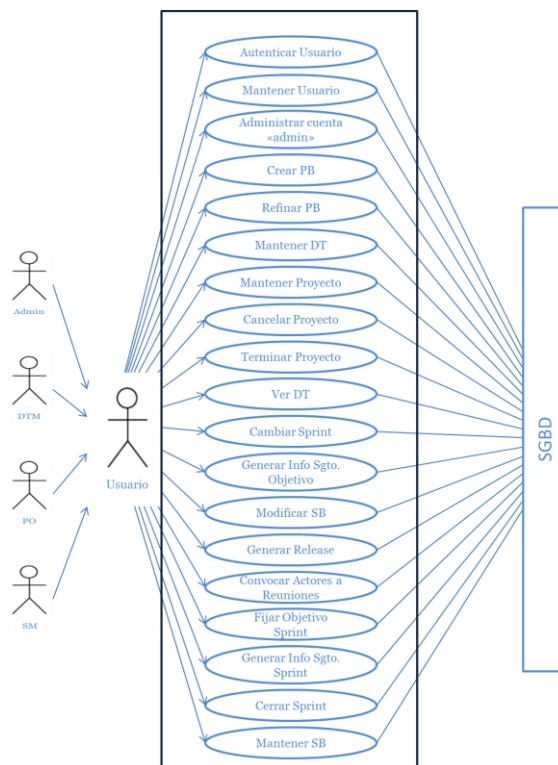


Ilustración 5: Diagrama de CUs

Donde los “monigotes” son los actores del sistema y la caja denominada “SGBD” es un actor externo al sistema. La caja representa el sistema y los óvalos cada uno de los casos de uso de la aplicación.

Actores	
Id	Descripción
Actores Internos	
1	PO. Product Owner. Dueño del Producto
2	SM. Scrum Master
3	DTM. Development Team Member. Miembro del equipo de desarrollo
4	Admin. Administrador del sistema
Actores Externos	
5	Sistema Gestor de Base de Datos

Tabla 3: Actores de la Aplicación

El conjunto de casos de uso que recogen todos los requerimientos funcionales del sistema son:

Casos de Uso	
Id	Descripción
1	Autenticar Usuario
2	Mantener Usuario
3	Administrar cuenta de administrador
4	Crear Product Backlog (PB)
5	Refinar PB
6	Mantener equipo de desarrollo (DT)
7	Mantener Proyecto
8	Cancelar Proyecto
9	Terminar Proyecto
10	Ver DT
11	Cambiar Sprint
12	Generar Información de Seguimiento de Objetivo
13	Modificar Sprint Backlog (SB)
14	Generar Release
15	Convocar a actores a reuniones
16	Fijar objetivo del sprint

Casos de Uso	
Id	Descripción
17	Generar Información de Seguimiento del Sprint
18	Cerrar Sprint
19	Activar Sprint
20	Mantener SB
21	Notificar Avisos

Tabla 4: Detalle Casos de Uso

Los requisitos no funcionales identificados se señalan seguidamente:

Requisitos no funcionales	
Id	Descripción
1	Disponer de MS Access (aconsejable aunque no necesario)
2	Disponer de conectividad de red
3	Disponer de máquinas con capacidad suficiente, tanto de operación como de presentación
4	Dotar de los permisos de lectura/escritura/modificación suficientes a los usuarios en función de la ubicación de los archivos

Tabla 5: Requisitos no funcionales

5.3. Metodología empleada

La elaboración de un Trabajo Fin de Grado unipersonal, como es el caso del presente, conlleva una adaptación de cualquier metodología que se pudiera emplear a dicha casuística, poco común, en proyectos de la vida real.

En todo caso el autor se ha ceñido al máximo a la metodología ágil Scrum, donde con la colaboración del director del Trabajo Fin de Grado, ha ejercido todos los roles, incluido el de usuario. Por lo que gran parte de la forma en la que la aplicación se comporta se debe al criterio del autor, validado por el Director del Trabajo Fin de Grado.

Hitos de Proyecto	
	Fechas Entrega
Definición alto nivel TFG	1-jun-15
Diseño de alto nivel de la solución	10-jun-15
Desarrollo de la solución	10-jul-15
Prueba de la solución	23-jul-15
Documentación TFG	
Plan de Trabajo	21-may-15
Arquitectura Prototipo y Estructura de la memoria	10-jun-15
Prototipo y Borrador de la memoria	29-jun-15
Redacción memoria final	24-jul-15
Depósito ordinario	24-jul-15

Ilustración 6: Hitos del TFG

En este sentido, y como ya se ha comentado, la fuente de toma de requisitos ha sido la documentación existente sobre Scrum así como la propia experiencia.

De esta forma se ha seguido una metodología eminentemente práctica e iterativa que ha permitido construir la aplicación, y su documentación asociada, en base a iteraciones y la resolución de los problemas y situaciones encontradas a la vez que se iban afrontando.

Esta memoria, junto con los entregables parciales producidos, contiene la documentación existente del Trabajo Fin de Grado.

Al tratarse de un Trabajo Fin de Grado unipersonal, los problemas comunes a cualquier proyecto de otras dimensiones, relacionados con gestión de configuración, control de versiones, procedimientos de documentación, entorno de desarrollo,... se simplifican enormemente.

De forma sencilla, se estableció un control de versiones (y copias de seguridad) del software que se iba generando, basado en sistema de ficheros local consistente, en la creación de versiones finalizadas diariamente, siendo éstas nombradas como la versión “AAAAMMDD_TFGAF”, de manera que se podría recuperar cualquier versión previa que pudiera requerirse. Al solo haber un desarrollador, el propio autor, no podía existir conflicto en actualización en paralelo de una misma versión. El punto de fallo que supone la posible rotura del ordenador de desarrollo se ha suplido mediante la realización de copias de seguridad diaria a soporte externo, así como en la nube empleando Google Drive.

La documentación se ha ido generando a medida que avanzaba el proyecto, bien a través de los entregables parciales de la planificación, bien en el propio código de la aplicación – realizando comentarios auto-explicativos dentro del mismo – y con esta propia memoria.

El entorno de desarrollo ha sido Microsoft Visual Studio 2012 Express para aplicaciones de escritorio que, en su proceso de instalación configura y descarga de internet todos los recursos necesarios para el correcto funcionamiento de entorno de desarrollo.

El SGBD seleccionado finalmente fue Microsoft Access versión compatible 2000. (*Vea apartado Descripción Técnica para detalle*)

La elección de esta plataforma de desarrollo, técnica y funcionalmente completa, se debió principalmente a la menor curva de aprendizaje necesaria por parte del autor del Trabajo Fin de Grado de cara a maximizar el empleo de la ventana temporal disponible en la producción de la solución funcional. Al tratarse de una aplicación de escritorio solo queda explicar por qué la versión elegida de Visual Studio 2012 express. Pues bien, tras instalar la última

versión Microsoft Visual Studio 2013 Ultimate, empleando la licencia dreamspark de la UNIR, se observó que el consumo de máquina hacía inviable el proyecto pues, a modo de ejemplo, la carga inicial de entorno llevaba más de 2 minutos en la máquina de desarrollo y cualquier compilación se alargaba en demasía. La versión elegida, mucho más ligera y sin coste, planamente funcional y completa desde la perspectiva del Trabajo Fin de Grado, no presentaba estos problemas y de ahí su elección final.

La selección de Microsoft Access se ha debido a la sencillez de instalación, configuración, administración y mantenimiento frente a otros SGBD.

Esta misma ventaja se traslada al usuario final, lo que supone que no tenga que invertir nada de tiempo en las labores de gestión y administración del SGBD cuando la aplicación esté en producción. No tiene por qué tener instalado MS Access en su entorno, y no requiere de ningún tipo de personal de administración experto. Por otro lado los requisitos de carga del SGBD no son exigentes aunque la aplicación se emplease en condiciones reales de funcionamiento.

Un buen equipo de trabajo debe ser consciente de sus puntos fuertes y de aquellos menos fuertes de forma que se potencien los elementos más favorables para alcanzar los objetivos del proyecto en tiempo y forma, evitando desvíos de la planificación inicial.

6. Descripción del Proyecto

Este apartado de la memoria recoge la funcionalidad que se ha implementado en la aplicación que se ha desarrollado. No trata de ser un manual de usuario, y de hecho no lo es, sino es más bien una guía detallada de la potencia funcional que incorpora la aplicación.

Para ello se va a recorrer la aplicación, en flujo de trabajo siguiendo un Scrum canónico y en función del rol que se desempeña. De forma muy rápida, recordar que un equipo de Scrum está constituido por un PO, un SM, y al menos, un DTM; además la aplicación incorpora el rol de “Administrador” para la realización de tareas burocráticas.

6.1. Autenticación de usuario

Para determinar qué rol desempeña un usuario que se quiere conectar a la aplicación, éste debe identificarse y autenticarse en la aplicación. Esto se hace con sus credenciales basadas en usuario y contraseña.



Ilustración 7: Autenticación

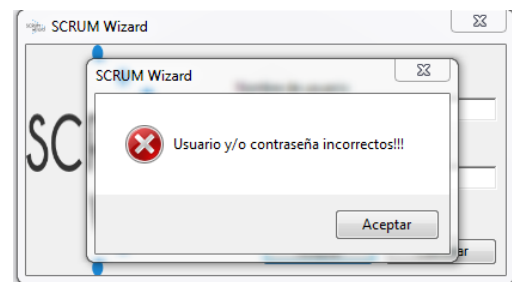


Ilustración 8: Fallo de Autenticación

La aplicación ha sido diseñada de forma que un mismo usuario pueda desempeñar distintos roles en distintos proyectos. Para ello, una vez autenticado el usuario, la aplicación mostrará la lista de proyectos (ver Ilustración 9: Lista de Proyectos) en los que ese usuario trabaja con un rol u otro para que, el usuario proceda a elegir el proyecto con el que quiere trabajar. Pulsando sobre el botón nombrado “Seleccionar” de la línea del proyecto que queremos para seleccionar el mismo.

Cuando un usuario no esté trabajando aún para ningún proyecto la aplicación lo detectará e informará al usuario permitiéndole crear un proyecto – para ello debe pinchar sobre el botón nombrado “Crear nuevo proyecto” (ver Ilustración 19: Creación de un Nuevo Proyecto) -. El rol que asigna la aplicación al usuario cuando decide crear un nuevo proyecto es el de PO. En otras palabras, el único rol autorizado por la aplicación a crear nuevos proyectos es el de PO. (Ver Ilustración 10: Usuario sin Proyectos)

Para evitar confusión, y entendiendo que en ocasiones puede resultar tedioso, el cambio de proyecto requiere salir completamente de la aplicación y volver a entrar.

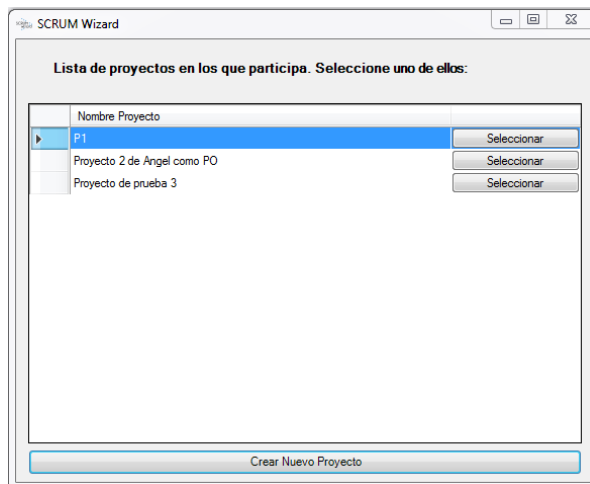


Ilustración 9: Lista de Proyectos

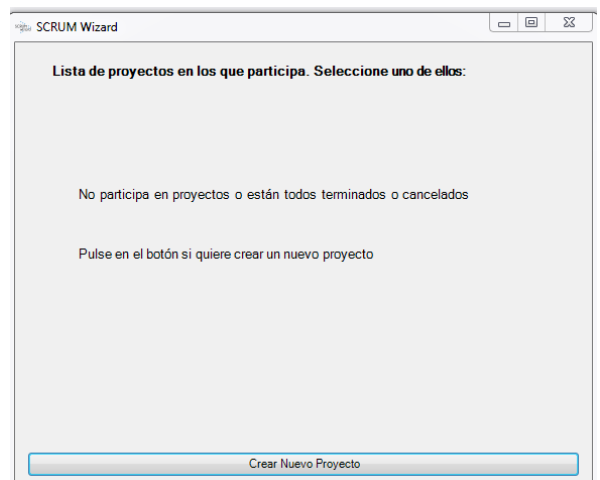


Ilustración 10: Usuario sin Proyectos

Para que un usuario pueda crear un proyecto, éste debe formar parte de un equipo de trabajo. Esto se debe a que en el momento de la creación del proyecto uno de los elementos obligatorios es explicitar el equipo de trabajo del mismo. Y si el PO no pertenece a ningún equipo entonces no sería posible asignar equipo de trabajo al proyecto.

La aplicación, cuando se produce esta situación lo detecta e informa al usuario y le indica que es el “Administrador” quien puede establecer los equipos.

6.2. El usuario “Administrador” en la aplicación

Este rol no está definido en Scrum, sin embargo ha sido incluido en la aplicación, por ser considerado fundamental para descargar de trabajo de poco valor añadido a cualquier otro de los integrantes del Equipo de Trabajo.

Las opciones de menú que presenta la pantalla de presentación del usuario administrador son:

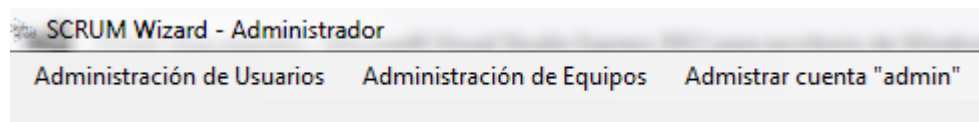
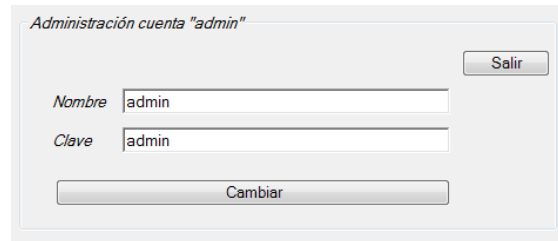


Ilustración 11: Opciones de Menú del Administrador

Además de las opciones mostradas, dispone de una más nombrada como “Salir” y que al ser pinchada nos saca completamente de la aplicación. Esta opción “Salir” está presente en toda las pantallas de presentación de todos los roles de la aplicación.

Este usuario es el único que originalmente vendrá cargado como usuario, siendo su nombre de usuario y contraseña: “admin”. Lógicamente el nombre de usuario y contraseña puede ser cambiada a voluntad del administrador. Para realizar estos cambios solo debe pinchar sobre la opción de menú nombrada como “**Administrar cuenta “admin”**”



Administración cuenta "admin"

Nombre

Clave

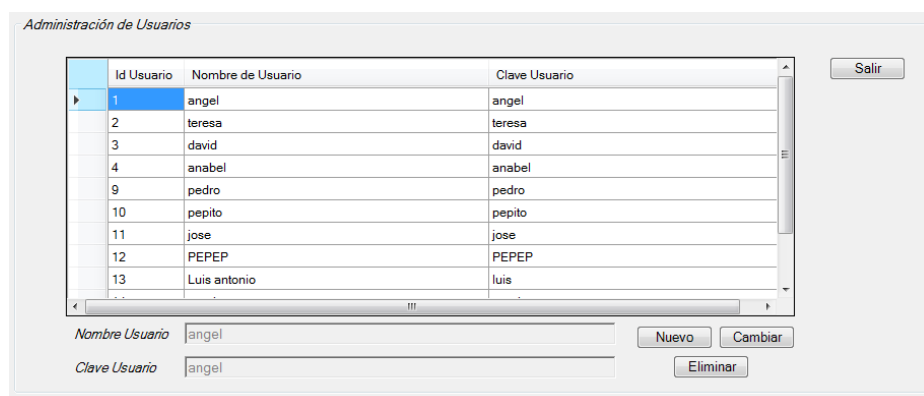
Ilustración 12: Cambio de usuario y contraseña de la cuenta Administrador

Se recomienda, en todo caso, no modificar el nombre de usuario y de hacerlo asegurarse de que no coincide con ningún otro del sistema. La aplicación es capaz de identificar esta situación y da prioridad a la cuenta “Administrador” y si se produjese esta situación el usuario “estándar” con el mismo nombre quedaría inhabilitado.

Las tareas que realiza el Administrador, no son de alto valor añadido y sin embargo sin ellas no podría utilizarse la aplicación.

Las otras dos funciones que puede realizar el usuario “admin” es el mantenimiento de usuarios de la aplicación y la administración de equipos.

Al **mantenimiento de usuarios** se llega pinchando en la opción de menú nombrada como “Administración de Usuarios” de la pantalla de presentación que habilita las siguientes posibilidades:



Administración de Usuarios

Id Usuario	Nombre de Usuario	Clave Usuario
1	angel	angel
2	teresa	teresa
3	david	david
4	anabel	anabel
9	pedro	pedro
10	pepito	pepito
11	jose	jose
12	PEPEP	PEPEP
13	Luis antonio	luis

Nombre Usuario

Clave Usuario

Ilustración 13: Administración de Usuarios

La aplicación presenta la lista de usuarios existentes en el sistema completamente caracterizados y permite: crear uno nuevo, cambiar alguna característica del usuario

seleccionado o bien eliminar el usuario seleccionado. La selección se produce pinchando sobre la línea que identifica al usuario con el que se quiere trabajar.

Para la **administración de equipos**, la aplicación, tras la selección de la opción de menú correspondiente, muestra una pantalla que varía en función de la opción elegida. Cambia para dotar de toda la funcionalidad necesaria para el mantenimiento de equipos de trabajo.

Administración de Equipos

	Id Equipo	Nombre de Equipo
▶	1	Equipo 1
	2	Equipo2
	3	Equipo 3
	4	Equipo 4
	5	Equipo 5
	6	Equipo 6
	11	Equipo 11
	12	Equipo 12
	13	Equipo trece

Nombre Equipo

Ilustración 14: Administración de Equipos

Muestra la lista de los equipos existentes y permite crear un nuevo equipo, cambiar un equipo o bien eliminar un equipo de trabajo. Además nos permite ver qué usuarios componen el equipo seleccionado.

El funcionamiento es similar, como ejemplo, si pinchamos sobre el botón nombrado “Nuevo” la aplicación permite la creación de un nuevo equipo de trabajo. Para ello lo primero que debemos declarar es el nombre que daremos a nuestro equipo de trabajo – si bien, la aplicación permite usar cualquier nombre, incluso repetidos, se sugiere emplear nombres no repetidos y que sean fácilmente identificables -, una vez declarado se procederá a incluir a los componentes del equipo.

La aplicación obliga a que cada equipo de trabajo tenga un PO, un SM y al menos un DTM, y además controla e informa de los tamaños de equipos recomendados por Scrum cuando es preciso.

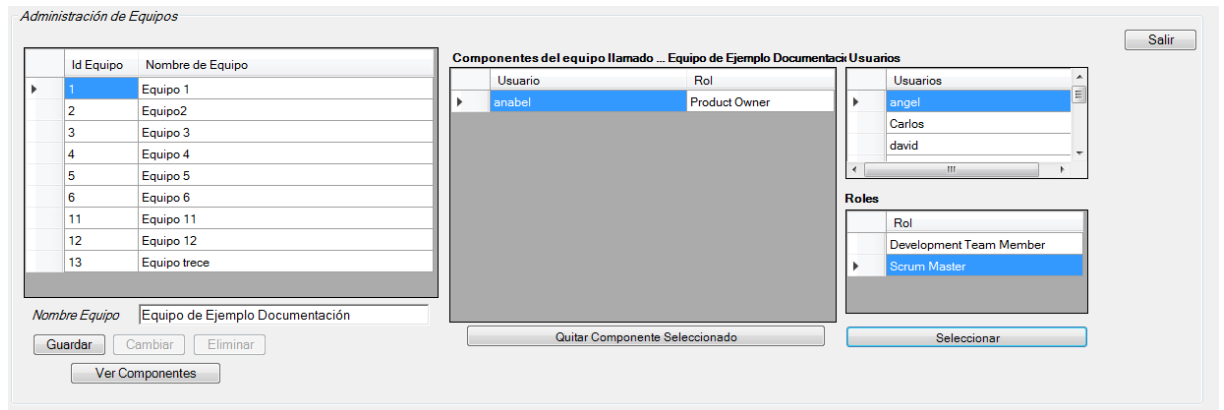


Ilustración 15: Incluyendo Usuarios en Equipo

Se muestran los usuarios del sistema, no incluidos en el equipo y, los roles posibles aún por incluir. Se selecciona el par usuario-rol y se pincha en el botón nombrado como “Seleccionar” y de esta forma pasan a formar parte del equipo. Hasta que no pinchemos sobre el botón nombrado como “Guardar” no se realizan los cambios.

6.3. ¿Qué puede hacer el PO en la aplicación?

Las tareas que el PO puede hacer en la aplicación son las que Scrum establece para este rol. Concretamente las indicadas por (Sutherland & Schwaber, 2013) en la Guía de Scrum.

Para ejercer su responsabilidad, la aplicación, al identificarse y seleccionar un proyecto para el que su rol es PO, le muestra la pantalla de presentación de PO:

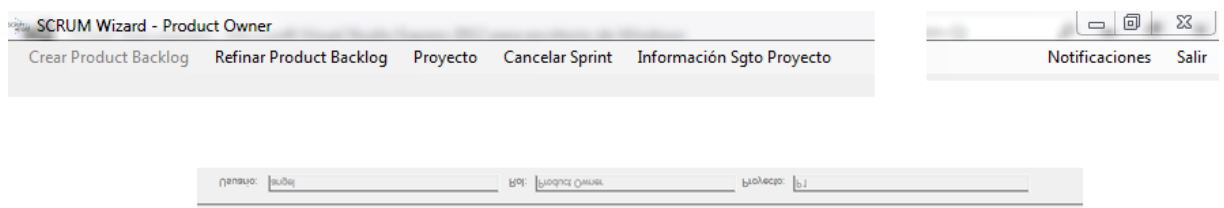


Ilustración 16: Pantalla de presentación del rol PO

En ella se ofrecen las opciones disponibles para un PO. Se hace mediante las opciones de menú mostradas en la Ilustración 16: Pantalla de presentación del rol PO: “Crear Product Backlog”, “Refinar Product Backlog”, “Proyecto”, “Cancelar Sprint”, “Información Sgto. Proyecto”, “Notificaciones” y “Salir”

Además muestra elementos de ayuda a la contextualización, es decir, quién soy (nombre de usuario), qué rol desempeño y con qué proyecto estoy trabajando.

6.3.1. Trabajando con el PB

El dueño del PB es el PO y a él le corresponde en exclusiva su creación y modificación, o usando la terminología Scrum su refinamiento.

Estas son las opciones que da la aplicación. La creación, solo está disponible la primera vez que se trata de generar un PB; tan pronto como se almacena ya solo podrá refinarse. En ambos casos el funcionamiento, de cara al usuario, de la aplicación es idéntico. Si bien desde un punto de vista técnico, en términos generales, la opción “Crear” realiza “INSERTS” en el SGDB y la opción “refinar” se apoya en “UPDATES” para las modificaciones, en “INSERTS” para las nuevas historias y en “DELETES” para la eliminación de historias.

	ID Historia	Descripción Historia	Estado	Tamaño (días)	Sprint	Prioridad	Coste Estimado	Valor Estimado	Comentarios
▶	1	H1	Liberada	2	1	4	5	6	7
	2	H2	Terminado	5	1	1	2	3	4
	3	H3	Terminado	9	2	5	4	6	7
	4	H4	Planificado	8	2	8	1	1	1
	5	H5	Cancelado	9	4	5	6	6	
	6	H6	Planificado	8	3	6	6	6	
*	7								

Ilustración 17: Estructura del PB

Cada una de las líneas del PB es una historia. Cada historia viene caracterizada por: Identificador único de historia, descripción de la historia, Estado en el que se encuentra la historia, tiempo que tardará (expresado en días) en implementarse la historia, el sprint en el que se abordará la historia, su prioridad, el coste y valor estimado en euros y cualquier comentario que queramos incluir.

El estado de la historia puede ser: Accionable, Cancelado, En Marcha, Planificado, Terminado (entendiéndose por terminado el concepto terminado de Scrum) y Liberada.

Los estados son auto explicativos. El estado liberada se produce automáticamente cuando el PO decide liberar una “release” de producto a cliente.

La aplicación controla que los valores que se introducen son correctos y permitidos. No solo en que un campo que espera un valor efectivamente tiene un valor válido, sino también que, por ejemplo, no se identifica un sprint por encima del número de sprints totales del proyecto, ... La validación se realiza tanto en el momento de la introducción de la información como en el momento de la grabación. Se ha preferido hacerlo de esta forma para evitar inconsistencias y la pérdida de la referencia sobre dónde se dan dichas inconsistencias.

En la Ilustración 17: Estructura del PB se observa que dos historias tienen la celda Id Historia de distinto color. El color rojo indica que la historia es parte de un incremento de

producto que ya ha sido liberado al cliente. El color celeste indica que la historia pertenece a un sprint que ha sido declarado terminado completamente.

Para guardar los cambios que se introduzcan se deberá pinchar sobre el botón nombrado como “Guardar” y siempre se podrá abandonar la tarea realizada con solo pinchar sobre el botón nombrado como “Salir”. En este último caso la aplicación avisa de la pérdida de los cambios que se hubiesen podido introducir.

6.3.2. Trabajando con Proyectos

Las tareas que el PO puede hacer con los proyectos se gestionan en la aplicación a través de esta opción del menú de la pantalla de presentación.

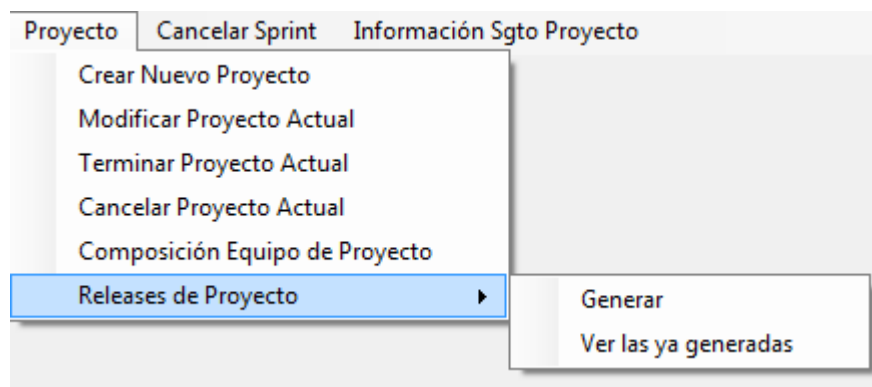


Ilustración 18: Tareas del PO relacionadas con Proyectos

Como se observa en la Ilustración, las opciones disponibles son: Crear un Nuevo Proyecto, Modificar Proyecto Actual, Terminar Proyecto Actual, Cancelar Proyecto Actual, Composición Equipo de Proyecto y Releases de Proyecto, a su vez con dos posibles opciones: Generar y Ver las ya generadas.

La opción de “Crear Nuevo Proyecto” permite la creación de un nuevo proyecto (es la misma funcionalidad que ofrece la aplicación desde la Lista de Proyectos ya comentada).

Ilustración 19: Creación de un Nuevo Proyecto

Los datos que se precisan son un nombre suficientemente significativo de proyecto, una fecha de inicio y otra de finalización, el número de sprints que lo conformarán (la aplicación hace una propuesta y controla que la duración de los sprints se ajusta a Scrum), el equipo

que realizará el proyecto que será seleccionado de entre aquellos equipos para el que el usuario es el PO del equipo y por último el número de horas al día que se trabajarán en el proyecto.

La opción “Modificar Proyecto Actual” funciona igual de cara al usuario que la creación, salvo por dos temas, no se puede modificar un proyecto con PB ya creados (en futuras versiones podría contemplarse como una mejora) y se presentan los datos ya registrados de ese proyecto para que estos pueden alterarse por parte del usuario.

La opción “Terminar Proyecto Actual” declara terminado el proyecto. La aplicación hace las comprobaciones pertinentes para que ello sea posible.

La opción “Cancelar Proyecto Actual” declara cancelado el proyecto. La aplicación hace las comprobaciones pertinentes para que ello sea posible.

La opción “Composición de Equipo de Proyecto” permite conocer la composición del equipo de trabajo del proyecto en curso. La Ilustración 41: Formulario en tiempo de ejecución muestra cómo se presenta esta información.

Por último la opción “Releases” permite al PO, por un lado generar una “Release” y por otro “ver las releases ya generadas”

Generación de la 'Release' para el proyecto

Id Release: Nombre Release:

Release PENDIENTE de liberar a cliente

Lista de historias candidatas a ser liberadas

Historia
H3

Historias de la Release

Historia
H2

Ilustración 20: Liberación de Incremento a Cliente

La aplicación, para generar las releases, requiere que se proporcione un nombre suficientemente identificativo a la release y que se incluyan las historias (ya terminadas) en la release, para lo cual se irán añadiendo las historias ya terminadas que interesen (la aplicación presenta las historias que están terminadas). Para proceder a la liberación basta con pinchar sobre el botón nombrado como “Liberar”.

La opción para ver las releases ya liberadas, muestra la lista de releases y las historias que la componen.

6.3.3. Cancelación de Sprint

Cuando el PO quiera cancelar un sprint, conforme a lo estipulado por Scrum al respecto, seleccionará esta opción de menú. Al hacerlo la aplicación le ofrecerá la lista de sprints del proyecto no terminados ni cancelados para que seleccione el que quiere cancelar. Una vez seleccionado la aplicación informará de las consecuencias de la acción a realizar, solicitando confirmación para su ejecución.

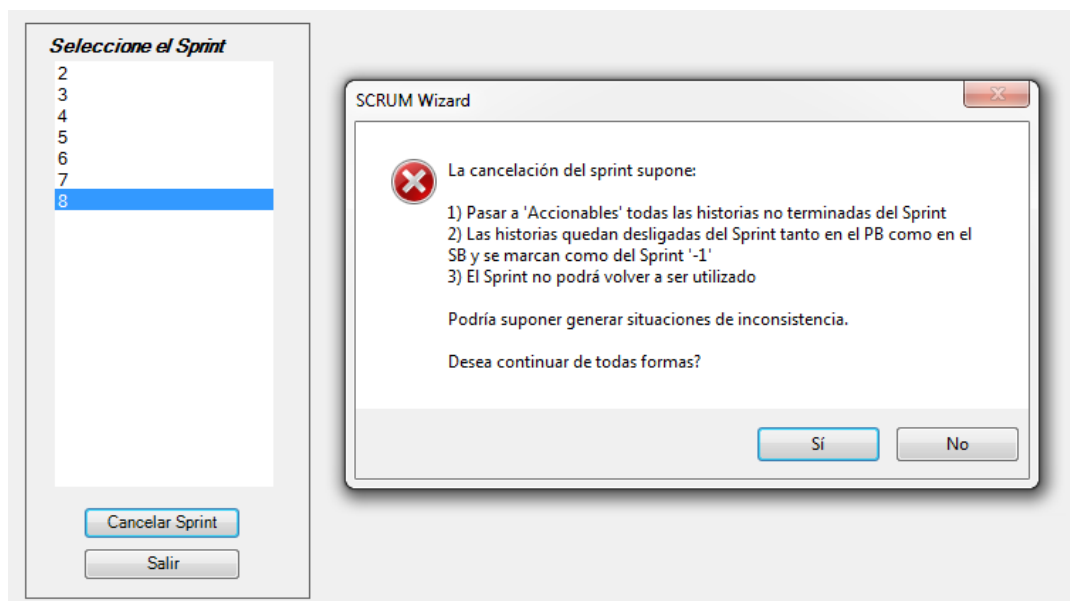


Ilustración 21: Cancelación de Sprint

6.3.4. Información de Seguimiento de Proyecto

Una de las labores que debe realizar el PO es conocer en todo momento la situación del proyecto; así como el comportamiento hasta el momento y el proyectado hasta la finalización del mismo.

Esta información puede ser requerida en cualquier momento. La aplicación genera esta información de forma sencilla y básica, aunque totalmente válida y útil. Este podría ser otro de los aspectos de mejora de la solución en futuras versiones.

La información que se proporciona es: la lista de sprints con su fecha de comienzo, duración en días y estado, así como valores estadísticos de comportamiento pasado y proyectado. Esta información proporciona: el número total de sprints, el número de ellos ya terminados a fecha del informe, el número de ellos pendientes de terminar, el número de días – en término medio - por sprint ya terminados y la proyección de la fecha de finalización del proyecto en base a estos valores calculados.

	Sprint	Fecha Comienzo	Duración	Estado
▶	1	29/06/2015	9 d	Cerrado
	2	14/07/2015	9 d	Activo
	3	29/07/2015	15 d	Planificado
	4	13/08/2015	15 d	Planificado
	5	28/08/2015	15 d	Planificado
	6	12/09/2015	15 d	Planificado
	7	27/09/2015	15 d	Planificado

Total Sprints: Días por Sprint:

Núm. Sprints Terminados: Fecha proyectada finalización:

Núm. Sprints Pendientes:

Ilustración 22: Información de Seguimiento de Proyecto

6.3.5. Notificaciones

Esta opción, que está disponible para todos los roles de Scrum y todos los usuarios (salvo el Administrador) permite ver las notificaciones que ese usuario tiene. Además de los avisos vigentes puede ver los que ya han pasado de fecha y no atendió.

El usuario puede marcar como leído un aviso simplemente pinchando en el check nombrado como “Leído” de la línea de interés.

	Leído	Fecha Reunión	Hora Reunión	Duración Reunión	Aviso
▶	<input checked="" type="checkbox"/>	14/07/2015	09:00	06:00	Convocatoria Reunión de Planificación de Sprint

Ilustración 23: Notificaciones

Los avisos son generados por la aplicación y enviados por el SM a los miembros del equipo que correspondan en función del tipo de aviso.

6.4. ¿Qué puede hacer el SM en la aplicación?

Las tareas que el SM puede hacer en la aplicación son las que Scrum establece para este rol. Concretamente las indicadas por (Sutherland & Schwaber, 2013) en la Guía de Scrum.

Para ejercer su responsabilidad, la aplicación, al identificarse y seleccionar un proyecto para el que su rol es SM, le muestra la pantalla de presentación de SM:

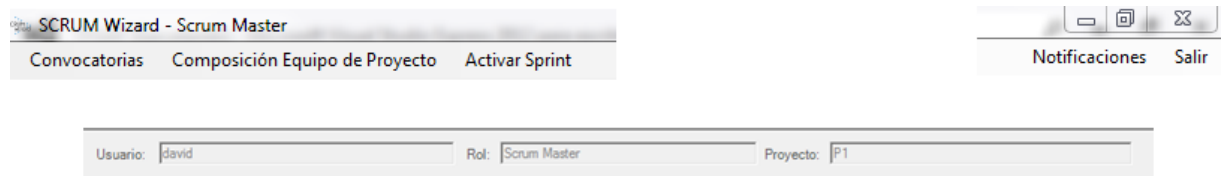


Ilustración 24: Pantalla de Presentación Rol SM

En ella se ofrecen las opciones disponibles para un SM. Se hace mediante las opciones de menú mostradas en la Ilustración 24: Pantalla de Presentación Rol SM: “Convocatorias”, “Composición Equipo de Proyecto”, “Activar Sprint”, “Notificaciones” y “Salir”

Además muestra elementos de ayuda a la contextualización, es decir, quién soy (nombre de usuario), qué rol desempeño y con qué proyecto estoy trabajando.

La opción “Composición de Equipo de Proyecto”, que como ya se ha comentado está disponible a todos los roles del equipo Scrum, permite conocer la composición del equipo de trabajo del proyecto en curso. La Ilustración 41: Formulario en tiempo de ejecución muestra cómo se presenta esta información.

6.4.1. Gestionando convocatorias

El SM es el rol responsable de asegurarse que se producen, en tiempo y forma, las reuniones de Scrum, y que, a las mismas, acuden las personas que deben asistir.

La guía de Scrum (Sutherland & Schwaber, 2013) determina que deben producirse los siguientes eventos con los asistentes que se muestran seguidamente.

Eventos de Scrum			
Nombre	Objetivo	Asistentes	Duración ³
SPM	Estimar el trabajo del sprint	ST	8 h
DS	<ul style="list-style-type: none">Responder a las 3 preguntasSincronizar el equipo de trabajo	DT	15 min
SR	<ul style="list-style-type: none">Demostrar al PO el incrementoBuscar cambios en el PB	ST	4 h

³ Valores para sprint de 4 semanas de duración. Para otras duraciones será proporcional

Eventos de Scrum			
Nombre	Objetivo	Asistentes	Duración ³
SRP	Identificar oportunidades de mejora en el DT	ST	3 h

Tabla 6: Eventos de Scrum

Para que el SM pueda desarrollar esta tarea que tiene asignada la aplicación al pinchar sobre la opción de menú nombrada “Convocatorias” le proporciona las opciones para convocar cada uno de los eventos indicados.

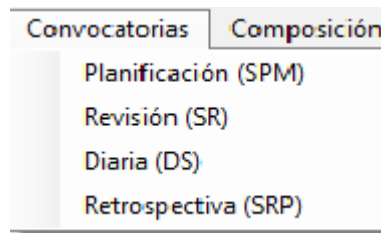


Ilustración 25: Opciones de Convocatorias

Al pinchar sobre cualquiera de las opciones mostradas la aplicación presenta una pantalla donde configurar cada una de las notificaciones que deben producirse. La aplicación propone los valores requeridos en función de la duración del sprint y las duraciones sugeridas.

Como ejemplo se muestra la pantalla que se presenta al pinchar sobre la opción nombrada “Diaria (DS)”.

A screenshot of a software interface titled 'Convocatoria de DS' (Daily Scrum Call). The interface is divided into two main sections. On the left, under the heading 'Listado de Sprint disponibles' (List of available sprints), there is a table with columns 'Sprint' and 'Fecha Comienzo' (Start Date). The table lists sprints 2 through 8, with sprint 4 selected. Below the table is a 'Seleccionar' (Select) button. On the right, under the heading 'Reunión de Planificación de Sprint' (Sprint Planning Meeting), there is a form for configuring the meeting. It includes fields for 'Proyecto' (Project), 'Sprint', 'Fecha Comienzo' (Start Date), '1a. Reunión' (1st Meeting), 'Hora Reunión' (Meeting Time), and 'Duración Reunión' (Meeting Duration). Below these fields is a table for 'Asistentes' (Attendees) with columns 'Asistente' (Attendee) and 'Rol' (Role). The table lists 'anabel' and 'teresa', both with the role 'Development Team M...'. To the right of the attendees table is a 'Ya Enviado?' (Already Sent?) section with radio buttons for 'Sí' (Yes) and 'No' (selected). At the bottom right are buttons for 'Enviar Convocatoria' (Send Call) and 'Abandonar' (Abandon). A 'Salir' (Exit) button is located at the top right of the main window.

Ilustración 26: Envío de Convocatoria de Daily Scrum

Al seleccionar el sprint de entre los presentados en el listado de sprints disponibles, se presenta la información que será enviada en la convocatoria de reunión. La información relativa al nombre del proyecto, el sprint y la fecha de comienzo no son modificables; la relativa a la primera reunión (pues en el caso del Daily Scrum hay una reunión diaria por cada día del sprint), la hora de la reunión y la duración de la reunión son modificables por el SM si bien la aplicación, de forma automática, presenta la información calculada a este respecto.

También se presenta a quién se enviará la convocatoria y si la misma ya se ha enviado o aún está por enviar. Cuando ya ha sido enviada, la aplicación mostrará esta información pero no permitirá cambios en la misma.

Al pinchar sobre el botón nombrado “Enviar Convocatoria” la(s) notificación(es) será(n) enviadas a todos los involucrados. Para ver éstas, el usuario deberá usar la opción de menú de su pantalla de presentación nombrada como “Notificaciones”)

6.4.2. Activar Sprint

A través de la activación de sprint, el SM informa cuál es el sprint en el que se encuentra el proyecto. De esta forma la aplicación obliga a explicitar el sprint activo.

Para ello muestra la lista de sprints que podrían ser activados. El SM seleccionará el que corresponda y al pinchar sobre el botón nombrado “Activar Sprint” éste queda activado.

6.4.3. Notificaciones

Esta opción, que está disponible para todos los roles de Scrum y todos los usuarios (salvo el Administrador) permite ver las notificaciones que ese usuario tiene. Además de los avisos vigentes puede ver los que ya han pasado de fecha y no atendió.

El usuario puede marcar como leído un aviso simplemente pinchando en el check nombrado como “Leído” de la línea de interés. (Ver Ilustración 23: Notificaciones)

Los avisos son generados por la aplicación y enviados por el SM a los miembros del equipo que correspondan en función del tipo de aviso.

6.5. ¿Qué puede hacer el DT en la aplicación?

Las tareas que pueden realizar los componentes (DTM) del DT pueden hacer en la aplicación son las que Scrum establece para este rol. Concretamente las indicadas por (Sutherland & Schwaber, 2013) en la Guía de Scrum.

Para ejercer su responsabilidad, la aplicación, al identificarse y seleccionar un proyecto para el que su rol es DTM, le muestra la pantalla de presentación de DT:

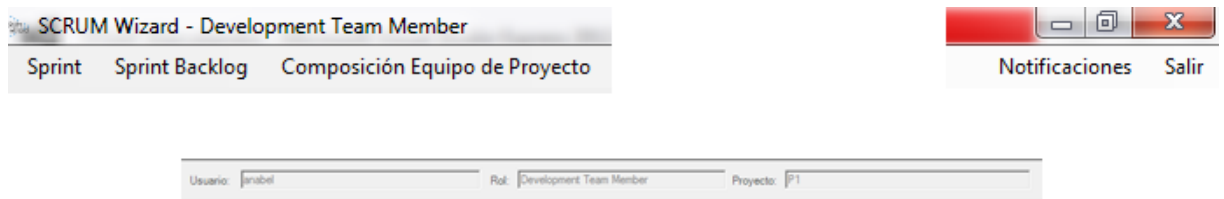


Ilustración 27: Pantalla de Presentación del Rol DTM

En ella se ofrecen las opciones disponibles para un DTM. Se hace mediante las opciones de menú mostradas en la Ilustración 27: Pantalla de Presentación del Rol DTM: “Sprint”, “Sprint Backlog”, “Composición Equipo de Proyecto”, “Notificaciones” y “Salir”

Además muestra elementos de ayuda a la contextualización, es decir, quién soy (nombre de usuario), qué rol desempeño y con qué proyecto estoy trabajando.

La opción “Composición de Equipo de Proyecto”, que como ya se ha comentado está disponible a todos los roles del equipo Scrum, permite conocer la composición del equipo de trabajo del proyecto en curso. La Ilustración 41: Formulario en tiempo de ejecución muestra cómo se presenta esta información.

6.5.1. Gestionando Sprints

La aplicación, a través, de la opción de menú nombrada “Sprint” de la pantalla de presentación del DTM, ofrece las siguientes opciones:

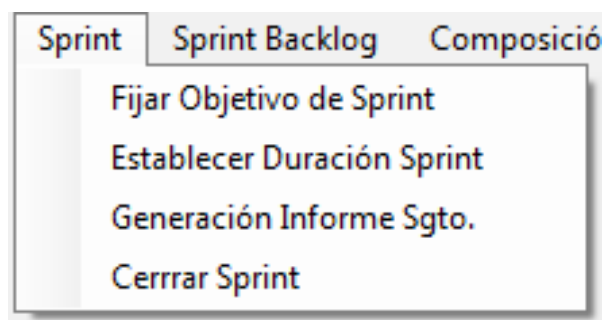
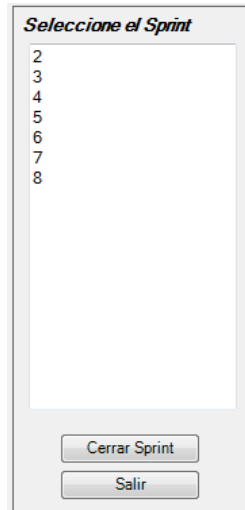


Ilustración 28: Opciones de un Sprint

Para cualquiera de las cuatro opciones mostradas, lo primero que requiere al usuario la aplicación, es que seleccione el sprint, de entre los disponibles presentados para el proyecto mediante la siguiente pantalla de selección de sprint:



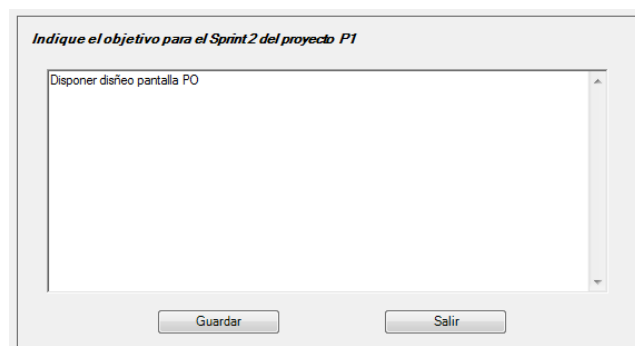
Seleccione el Sprint

2
3
4
5
6
7
8

Cerrar Sprint
Salir

Ilustración 29: Selección de Sprint

Cuando se está **“Fijando el objetivo del Sprint”** la aplicación presenta una pantalla donde tendrá que introducir dicho objetivo. Si el objetivo ya estuviese fijado permitirá su modificación. Al pinchar sobre el botón nombrado como “Guardar” se almacena el objetivo del sprint.



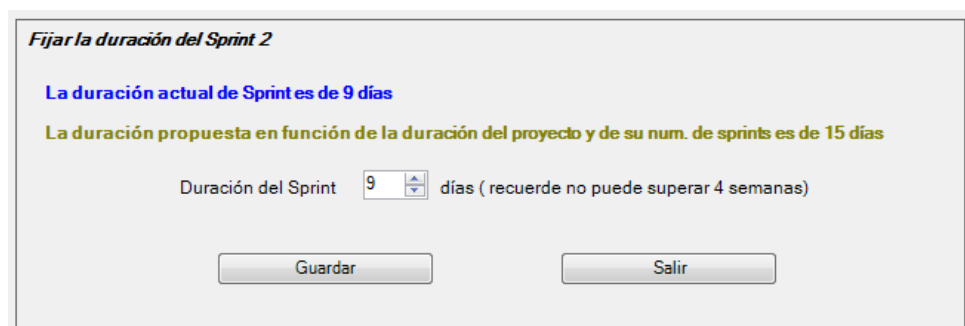
Indique el objetivo para el Sprint2 del proyecto P1

Disponer diseño pantalla PO

Guardar Salir

Ilustración 30: Objetivo de Sprint

Para **“Fijar la duración del Sprint”** la pantalla que presenta la aplicación se indica a continuación:



Fijar la duración del Sprint 2

La duración actual de Sprints es de 9 días

La duración propuesta en función de la duración del proyecto y de su num. de sprints es de 15 días

Duración del Sprint 9 días (recuerde no puede superar 4 semanas)

Guardar Salir

Ilustración 31: Fijar Duración Sprint

La pantalla presenta elementos informativos y elementos de configuración. La línea azul muestra la duración fijada actualmente para el sprint, y la línea de color verde la duración propuesta en base a la duración del proyecto y el número de sprints del mismo. Finalmente la aplicación habilita un espacio donde introducir la duración que queremos darle al sprint seleccionado.

Cuando se pincha sobre el botón nombrado “Guardar” de la pantalla la aplicación comprueba si el sprint tiene historias de duración en días superior a la duración que tratamos de fijar para el sprint. Si es así, la aplicación requerirá confirmación de la acción a realizar informando de esta situación. Se recomienda el uso responsable de esta opción pues se pueden generar graves inconsistencias.

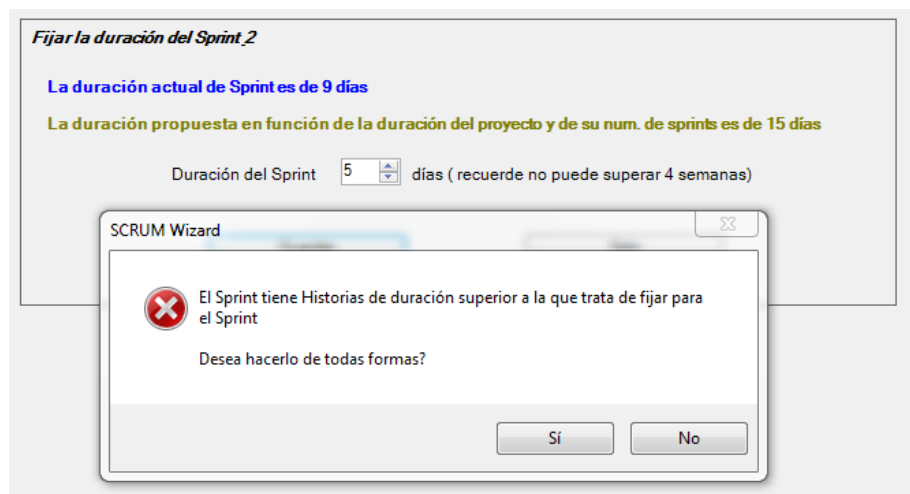


Ilustración 32: Aviso en cambio de la duración de Sprint

Otras de las labores que debe realizar el DT es conocer en todo momento la situación del sprint; así como el comportamiento hasta el momento y el proyectado hasta la finalización del mismo.

Esta información puede ser requerida en cualquier momento. La aplicación genera esta información de forma sencilla y básica, aunque totalmente válida y útil. Este podría ser otro de los aspectos de mejora de la solución en futuras versiones.

Para obtener esta información deberá seleccionar la opción nombrada “**Generación Informe Sgto.**”, una vez seleccionado el sprint.

La información que se proporciona es: la lista de historias con su fecha de comienzo, duración en días y en horas y su estado, así como valores estadísticos de comportamiento pasado y proyectado. Esta información proporciona: el número total historias, el número de

ellas ya terminadas a fecha del informe, el número de ellas pendientes de terminar, el número de días y horas – en término medio - por historia ya terminadas y la proyección de los días y horas proyectadas hasta la finalización del sprint en base a estos valores calculados.

	Historia	Fecha Comienzo	Duración Días	Duración Horas	Estado
▶	H3	14/07/2015	9 d	13 h	Terminado
	H4	14/07/2015	8 d	16 h	Planificado

Total Historias:	<input type="text" value="2"/>	Días por Historia:	<input type="text" value="9"/>	Horas por Historia:	<input type="text" value="13"/>
Núm. Historias Terminados:	<input type="text" value="1"/>	Días proyectados finalización:	Horas proyectadas finalización:		
Núm. Historias Pendientes:	<input type="text" value="1"/>	<input type="text" value="9"/>	<input type="text" value="13"/>		

Ilustración 33: Información de Seguimiento de Sprint

La última opción disponible es **“Cerrar Sprint”**, cuya finalidad es marcar el sprint seleccionado como cerrado, pasando a ser tratado de esta manera a partir de este momento. Esta es una opción silenciosa, esto es, no requiere confirmación por parte del usuario.

6.5.2. Gestionando el Sprint Backlog

El Sprint Backlog es el subconjunto de elementos del Product Backlog que serán implementados en un determinado sprint.

Por ello, la información que presenta la aplicación del SB se “hereda” del PB. Además pone a disposición del DT elementos que permiten caracterizar, en términos de plazo de ejecución por historia y qué DTM será el encargado de dicha tarea, la historia seleccionada.

Pero antes de pasar a detallar cada elemento del SB conviene señalar que la primera actuación que la aplicación requerirá es seleccionar el sprint para el que se va a crear o modificar su SB. La forma en que solicita esta información es idéntica a la explicada en el apartado anterior de esta memoria.

La Ilustración 29: Selección de Sprint muestra la forma en que se seleccionará el sprint.

Sprint Backlog del Sprint 2 del Proyecto P1															
	Id.His.	Historia	Responsable	Sprint	Estado	Duración (d)	D1 (h)	D2 (h)	D3 (h)	D4 (h)	D5 (h)	D6 (h)	D7 (h)	D8 (h)	D9 (h)
▶	4	H4	teresa	2	Accionable	8	5	5	2		1		1	2	
	3	H3	anabel	2	Terminado	9	1	2	4		6				

Ilustración 34: Detalle del SB

Cada línea del SB supone una historia de las pertenecientes al Sprint. La forma de incluir o eliminar nuevas historias en el SB es modificando el PB.

La caracterización de cada historia viene dada por: Identificador de Historia, Nombre de la Historia, *Responsable*, *Sprint*, *Estado de la Historia*, *Duración* de la historia en días, y las *horas* que se ejecutarán en cada uno de los días de trabajo.

Las características que pueden ser modificadas son las que están en cursiva en la lista anterior.

Responsable: recoge el miembro del equipo, asignado al proyecto, que se encargará de la implementación de la historia. La aplicación se encarga de mostrar solo los miembros del equipo asignado.

Sprint: indica en qué sprint se realizará la historia. Si alteramos este valor, la historia pasará automáticamente a ser ejecutada en el sprint que se indique. Este cambio tiene reflejo también en el PB. Ese cambio puede requerir autorización del PO, de ser así, deberá gestionarse por fuera de la aplicación.

Estado indica el estado de la historia. Pudiendo ser: Accionable, En marcha o Terminada

Duración (d) indica la duración en días necesarios para ejecutar la historia

Dx(h) la aplicación presenta tantas columnas nombradas como “Dx (h)” como la mayor de las duraciones de las historias (para el caso mostrado en la Ilustración 34, son 9 columnas) o en su defecto tantas columnas como días dure el sprint. Cada columna así nombrada permite incluir el número de horas que se trabajarán ese día para la historia.

Las comprobaciones de validez de la información introducida se producen en el momento en el que se tratan de guardar los cambios realizados, al haber pinchado sobre el botón nombrado “Guardar”. Las validaciones que se realizan son:

- El número de horas asignadas a un día no puede superar el número de horas asignadas al proyecto al que pertenece el sprint
- Los valores introducidos son válidos para la columna de que se trate o no están vacíos cuando sean obligatorios
- Se produce una pre-validación de coherencia entre la duración de la historia y el reparto de horas por día realizado. De forma que cuando la aplicación detecta una inconsistencia informa de la misma y espera confirmación para proceder a guardar los cambios

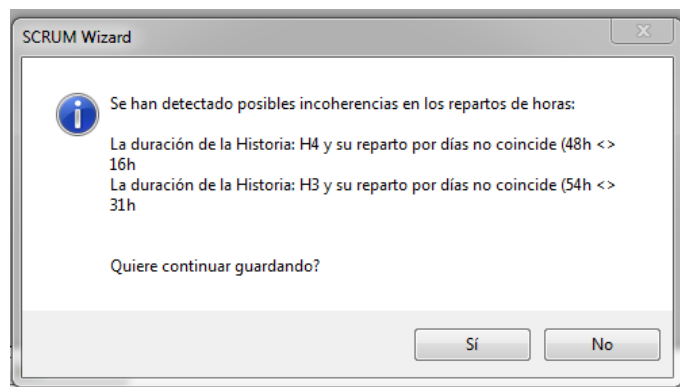


Ilustración 35: Aviso de discrepancia entre duración en días y en horas

Adicionalmente esta pantalla presenta la posibilidad de ver las historias “Accionables” del PB.

6.5.3. Composición de Equipo de Proyecto

La opción “Composición de Equipo de Proyecto” permite conocer la composición del equipo de trabajo del proyecto en curso. La Ilustración 41: Formulario en tiempo de ejecución muestra cómo se presenta esta información.

6.5.4. Notificaciones

Esta opción, que está disponible para todos los roles de Scrum y todos los usuarios (salvo el Administrador) permite ver las notificaciones que ese usuario tiene. Además de los avisos vigentes puede ver los que ya han pasado de fecha y no atendió.

El usuario puede marcar como leído un aviso simplemente pinchando en el check nombrado como “Leído” de la línea de interés. (Ver Ilustración 23: Notificaciones)

Los avisos son generados por la aplicación y enviados por el SM a los miembros del equipo que correspondan en función del tipo de aviso.

7. Descripción Técnica

De entre las posibles soluciones arquitectónicas y tecnológicas, múltiples y variadas, todas ellas válidas y cada una de ellas con sus puntos fuertes y con otros puntos menos fuertes; el autor, finalmente, ha optado por las que se expresan seguidamente por las razones expuestas en el apartado 5.3. Metodología empleada y que, a su juicio, han sido determinantes para alcanzar el objetivo del presente Trabajo Fin de Grado en plazo y forma.

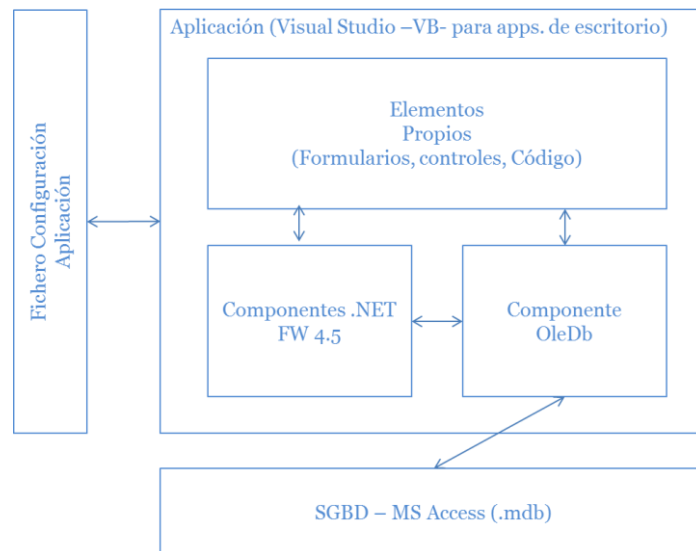


Ilustración 36: Arquitectura de Componentes

7.1. Aplicación – Visual Studio 2012 Express para aplicaciones de escritorio

El entorno global sobre el que se ha desarrollado este Trabajo Fin de Grado es el entorno de desarrollo integrado de Microsoft: Visual Studio 2012 Express para aplicaciones de escritorio de Windows (Microsoft).



Ilustración 37: Carátula de arranque VS 2012

Visual Studio 2012 Express para escritorio de Windows permite la creación de aplicaciones de escritorio en C#, Visual Basic, and C++ y admite Windows Presentation Foundation (WPF), Windows Forms y Win32.

El presente Trabajo Fin de Grado se ha realizado empleando Visual Basic. Este lenguaje presenta capacidades sobradas para la realización de los desarrollos de este Trabajo Fin de

Grado y la curva de aprendizaje del mismo ha sido sensiblemente inferior a cualquier otro lenguaje de entre los disponibles.

Los requerimientos del sistema que necesita Visual Studio 2012 Express son:

- Sistemas Operativos soportados
 - Windows 7 Service Pack 1; Windows 8; Windows Server 2008 R2 SP1; Windows Server 2012
 - Windows 7 SP1 (x86 y x64)
 - Windows 8 (x86 y x64)
 - Windows Server 2008 R2 SP1 (x64)
 - Windows Server 2012 (x64)
- Requisitos de hardware:
 - Procesador a 1,6 GHz o más rápido
 - 1 GB de RAM (1,5 GB si se ejecuta en una máquina virtual)
 - 5 GB de espacio disponible en el disco duro
 - Unidad de disco duro de 5400 rpm
 - Tarjeta de vídeo compatible con DirectX 9 con una resolución de pantalla de 1024 x 768 o superior

La máquina que se ha empleado para el desarrollo del Trabajo Fin de Grado ha sido:

TOSHIBA Satellite – Intel Core i7 – 3610 QM CPU @ 2,30 MHz RAM 6,00GB
Windows 7 Home Premium SP1 64 bits
Resolución de pantalla: 1366 x 768

Visual Studio 2012 Express es un entorno global de desarrollo de aplicaciones que proporciona multitud de herramientas (IDE, Ejecución paso a paso, Ejecución sin paradas, Ejecución por funciones, Trazabilidad, Control de Errores, Manejo de datos, ...).

El uso combinado de todas estas características permite maximizar el éxito de los proyectos que se abordan, pues pone a disposición del equipo de desarrollo multitud de elementos vitales para un buen desarrollo y posterior prueba de lo construido.

7.1.1. Cómo se ha usado Visual Studio 2012 Express

La forma en la que este entorno global de desarrollo funciona es muy simple y a la vez, esta simplicidad, es lo que lo dota de su extraordinaria potencia.

Visual Studio incorpora un conjunto de formularios y controles que pueden ser empleados en el desarrollo de la aplicación. Estos elementos, que serán explicados en detalle en los siguientes apartados de la presente memoria, incorporan una interfaz de usuario que puede ser alterada mediante la modificación de sus propiedades así como un conjunto de eventos que responden a ciertas acciones - por ejemplo, coger el foco, o perderlo, pinchar, se produce un error,... - realizadas por el usuario sobre el elemento, y que permiten incorporar código de usuario para la realización de la funcionalidad precisa.

Estos elementos se organizan en formularios o forms, que al igual que cualquier otro componente presenta propiedades que permiten su caracterización, para su ajuste a la necesidad del proyecto, así como eventos que admiten código de usuario que implementa la funcionalidad perseguida.

Sobre estas sencillas bases se puede implementar la herencia y el polimorfismo, al igual que construir desde cero o bien apoyado en algún elemento o controles totalmente nuevos.

En este Trabajo Fin de Grado no se han empleado módulos de terceros que potenciarían aún más las soluciones. Pero su uso es posible. Lógicamente cuanto más módulos se empleen, ya sean de MS o de terceros fabricantes, más exigente será la instalación a realizar.

7.2. Elementos propios

Los elementos que se han creado específicamente para la construcción de este Trabajo Fin de Grado se presentan en la Ilustración 38: Elementos Propios Creados. Cada uno de ellos con una finalidad que se expone a continuación:

My Project

Define y caracteriza los elementos comunes del proyecto. Se incluyen aspectos como: definición de la aplicación, forma de compilar, cómo depurar, referencias que utiliza la aplicación, recursos utilizados, servicios de autenticación, roles y perfiles empleados, configuración (por ejemplo la cadena de conexión a la base de datos), firma del código, extensiones My, Seguridad, publicación (generación de .exe) y análisis de código.

La adecuada configuración de estos elementos asegura el correcto funcionamiento de la aplicación. Un aspecto mal configurado podría suponer el no funcionamiento de la solución.

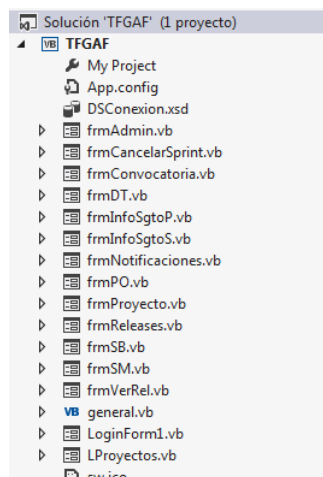


Ilustración 38: Elementos Propios Creados

App.Config

El fichero de configuración de la aplicación. Más adelante en esta memoria dedicaremos un apartado a este fichero, pieza clave, para el correcto funcionamiento de la aplicación.

DSConexion.xsd

Se trata de la conexión con la Base de Datos. Contiene los elementos de la base de datos y los atributos y métodos disponibles a través de este elemento. El detalle se muestra en Ilustración 39: Elemento DSConexion.xsd

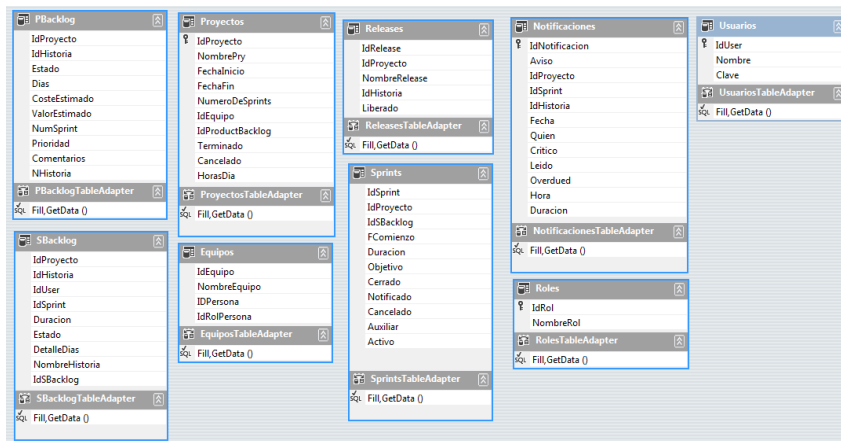


Ilustración 39: Elemento DSConexion.xsd

Este elemento se apoya en el modelo de datos creado en el SGBD para su construcción. En el momento de su creación el asistente nos guía en su generación. Para posteriores modificaciones, existe una herramienta que nos facilita la conexión con el SGBD y nos muestra aquellos cambios que se hubieran podido realizar en el SGBD. Esta herramienta nos permite realizar cualquier alteración que pudiese ser requerida.

De esta forma la conexión con el SGBD queda independizada a través de este control y nos aísla en el resto del código de tener que lidiar con los aspectos de configuración y adaptación.

frmAdmin.vb

Es el formulario que habilita todas las funcionalidades propias del rol “administrador”

frmCancelarSprint.vb

Formulario que permite la cancelación de un sprint.

frmConvocatoria.vb

Formulario que gestiona las convocatorias a reuniones.

frmDT.vb

Es el formulario que habilita todas las funcionalidades propias de aquellos usuarios cuyo rol en ese proyecto es DTM

frmInfoSgtoP.vb

Formulario que genera y presenta la información de seguimiento hacia un objetivo de un proyecto.

frmInfoSgtoS.vb

Formulario que genera y presenta la información de seguimiento y proyección de un sprint..

frmNotificaciones.vb

Formulario que habilita la lectura y gestión de los avisos existentes.

frmPO.vb

Es el formulario que habilita todas las funcionalidades propias de aquellos usuarios cuyo rol en ese proyecto es PO.

frmProyecto.vb

Formulario que permite el alta y mantenimiento de proyectos.

frmReleases.vb

Formulario que permite al PO hacer entrega de un incremento terminado a cliente.

frmSB.vb

Habilita la gestión del Sprint Backlog.

frmSM.vb

Es el formulario que habilita todas las funcionalidades propias de aquellos usuarios cuyo rol en ese proyecto es SM.

frmVerRel.vb

Permite la generación y visualización de las entregas realizadas a cliente.

General.vb

Es un módulo de ámbito general a toda la aplicación que contiene las variables, funciones y procedimientos de ámbito general.

LoginForm1.vb

Es el formulario inicial de la aplicación y sirve para la autenticación del usuario que pretende usar la aplicación. El acceso se controla mediante el uso de <nombre de usuario> y <contraseña>.

Lproyectos.vb

Formulario que presenta para el usuario con acceso autorizado la lista de proyectos en los que interviene. También permite crear un proyecto nuevo.

Sw.ico

El icono que se visualiza de la aplicación

Cada uno de los formularios descritos utiliza distintos elementos. Los elementos propios se han implementado sobre la base de elementos existentes en los componentes que se indican en la arquitectura y que se pormenorizan en siguientes apartados. El propio formulario es un elemento que también se apoya en componentes estándar.

Por ejemplo el formulario frmPO.vb en tiempo de diseño⁴ presenta los siguientes componentes:

⁴ En VS el tiempo de diseño hace referencia al momento en el que estamos diseñando la aplicación. Las acciones que se realizan se producen por acción manual sobre las características visuales y/o de comportamiento que presentan los elementos que empleamos

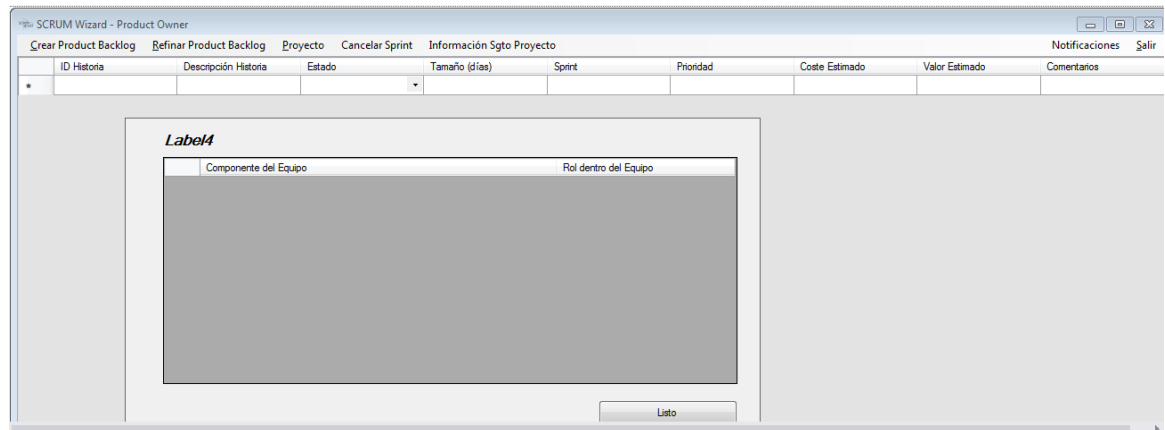


Ilustración 40: Componentes Formulario PO

En la ilustración se observa:

- el propio formulario, con su título e icono y los típicos botones de minimizar, maximizar y cerrar
- el uso del control de menú, con sus distintas opciones: Crear PB, Refinar PB, Proyecto, Cancelar Sprint, Información Sgto Proyecto, Notificaciones y Salir. Algunas de estas opciones de menú a su vez contienen otras opciones
- un control de tipo DataGridView que es el control que ayuda a representar y manejar las historias del PB. Este control en tiempo de diseño se caracteriza en aquellos aspectos generales y posteriormente, en tiempo de ejecución⁵, mediante el código insertado en los métodos apropiados, se puebla el DataGridView con la información relevante. Este control además permite la alteración de la información presentada
- Un elemento compuesto de 4 elementos simples (etiqueta, datagridview, botón Listo y panel general)

Este elemento compuesto se muestra cuando el usuario pincha sobre la opción del menú del formulario denominada “Composición del Equipo de Trabajo”, lo que hace que se muestren los componentes que integran un determinado equipo de trabajo y para que, efectivamente, esa funcionalidad se dé, el código necesario que hay que ubicar en el evento Click del elemento de menú citado es:

```
Private Sub ComposiciónEquipoDeProyectoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ComposiciónEquipoDeProyectoToolStripMenuItem.Click
    ' Se visualiza el equipo completo de ese proyecto
    Dim sql As String = ""
```

⁵ En VS el tiempo de ejecución hace referencia al momento en el que la aplicación se está ejecutando. Las acciones que se realizan se producen mediante la ejecución del código creado

```
Dim punto As New Point(50, 50)

sql = "SELECT Equipos.NombreEquipo, Usuarios.Nombre, Roles.NombreRol FROM
Equipos, Proyectos, Roles, Usuarios "
sql = sql & "WHERE (((Proyectos.IdProyecto)=" & IDPROYECTO & ") AND
((Equipos.IdEquipo)=[Proyectos].[IdEquipo]) AND
((Usuarios.IdUser)=[Equipos].[IDPersona]) "
sql = sql & "AND ((Roles.IdRol)=[Equipos].[IdRolPersona]))"

Dim cnn As New OleDbConnection(My.Settings.tfgConnectionString)

Try
    cnn.Open()
    Dim buscar As New OleDbCommand(sql, cnn)
    Dim pb As OleDbDataReader = buscar.ExecuteReader

    If pb.HasRows Then
        While pb.Read
            DGVEquipo.Rows.Add(pb.GetValue(1), pb.GetValue(2))
            lblEquipo.Text = "Lista de Integrantes de " & pb.GetValue(0)
        End While
        cnn.Close()
    Else
        MsgBox("No existe personal asignado al proyecto.", MsgBoxStyle.Information,
"SCRUM Wizard")
    End If
    DGVPBacklog.Visible = False
    butListo.Tag = CrearProductBacklogToolStripMenuItem.Enabled
    CrearProductBacklogToolStripMenuItem.Enabled = False
    RefinarProductBacklogToolStripMenuItem.Enabled = False
    CrearNuevoProyectoToolStripMenuItem.Enabled = False
    NotificacionesToolStripMenuItem.Enabled = False
    SalirToolStripMenuItem.Enabled = False
    CancelarSprintToolStripMenuItem.Enabled = False
    InformacionSgtoProyectoToolStripMenuItem.Enabled = False
    pnlEquipo.Location = punto
    pnlEquipo.Visible = True
Catch ex As Exception
    MsgBox("Ups!!! Problemas " & ex.Message, MsgBoxStyle.Critical, "SCRUM
Wizard")
End Try

End Sub
```

Que produce el siguiente resultado cuando el proyecto seleccionado es el equipo que realiza el proyecto denominado P3:

Componente del Equipo	Rol dentro del Equipo
angel	Product Owner
david	Scrum Master
anabel	Development Team Member
teresa	Development Team Member

Listo

Ilustración 41: Formulario en tiempo de ejecución

7.3. Componentes .NET Framework 4.5

Conforme se indica en (Microsoft .Net) “.NET Framework es una tecnología que admite la compilación y ejecución de la siguiente generación de aplicaciones y servicios Web XML. El diseño de .NET Framework está enfocado a cumplir los siguientes objetivos:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota
- Proporcionar un entorno de ejecución de código que minimiza los conflictos en el despliegue y versionado de software
- Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceros desconocidos o que no son de plena confianza
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web
- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código”

Los componentes de .Net Framework 4.5 están disponibles en Visual Studio de forma estándar y pueden ser utilizados conforme a la naturaleza de cada uno de ellos. Con solo desplegar la opción “Cuadro de Herramientas” Visual Studio 2012 permite seleccionar el componente de nuestro interés.

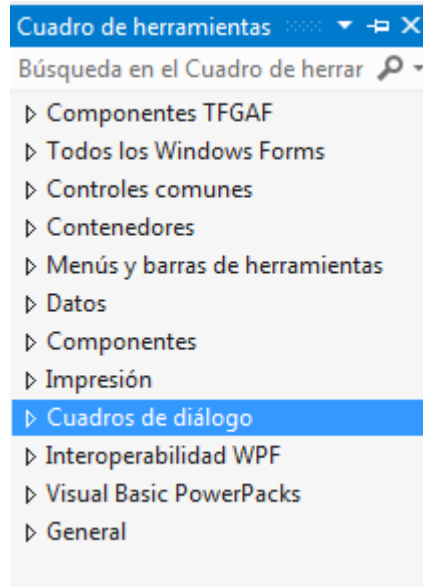
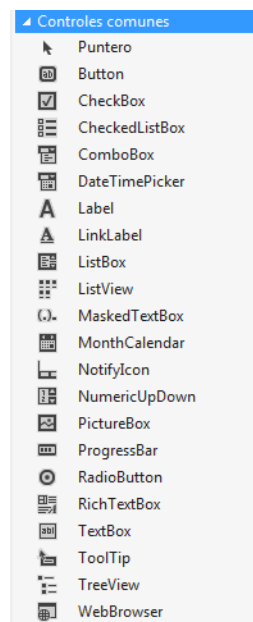


Ilustración 42: Controles Disponibles

La entrada “Componentes TFGAF” incluye el conjunto de elementos propios explicados en el apartado previo.

La entrada “Todos los Windows Forms” pone a disposición, al ser desplegado, todos los controles disponibles.

La entrada “Controles Comunes” incluye los siguientes controles:



**Ilustración 43:
Controles Comunes**

Los controles que se han empleado en este Trabajo Fin de Grado han sido:

- Button
- CheckBox
- ComboBox
- Label
- ListBox
- MarkedTextBox
- Month Calendar
- NumericUpDown
- RadioButton
- TextBox

La entrada “Contenedores” presenta elementos que permiten contener en ellos otros componentes, de forma que acciones de tipo mostrar, ocultar, activar, etc. se simplifican enormemente pues basta con ocultar el contenedor y todos los elementos contenidos se ocultan. Si no se empleasen y, siguiendo con el ejemplo, habría que ocultar componente a componente. Luego estos elementos ayudan a simplificar el código.

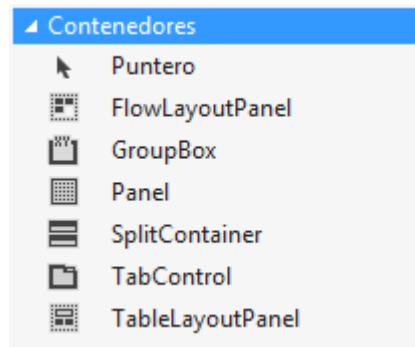


Ilustración 44: Componentes de Contenedores

Los elementos de este tipo utilizados en el Trabajo Fin de Grado han sido: GroupBox y Panel

La entrada “Menús y barras de herramientas” presenta elementos que se usan como opciones de menú. Este Trabajo Fin de Grado ha empleado solo menús tradicionales de ventanas, no contextuales, por lo que solo ha utilizado el elemento llamado “MenuStrip”

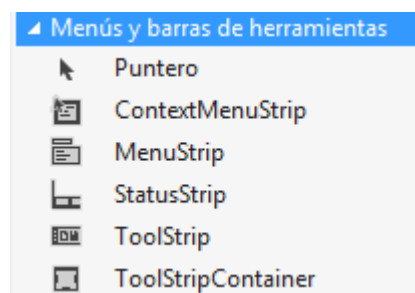


Ilustración 45: Menús y barras de herramientas

La entrada “Datos” permite emplear los componentes de acceso y manejo de datos. Este Trabajo Fin de Grado ha empleado intensivamente el control “DataGridView” pero sin estar “ligado automáticamente” a datos externos de base de datos. Esto se ha hecho de esta forma para permitir una hipotética migración de la aplicación a otro SGBD sin necesidad de

modificar el código. Solo mediante la parametrización del fichero “app.config” que se describe en el apartado 7.5. Fichero de configuración “app.config”.

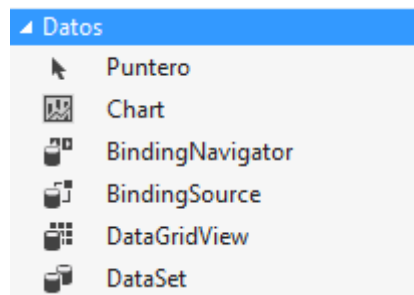


Ilustración 46: Componentes de Datos

Otras entradas:

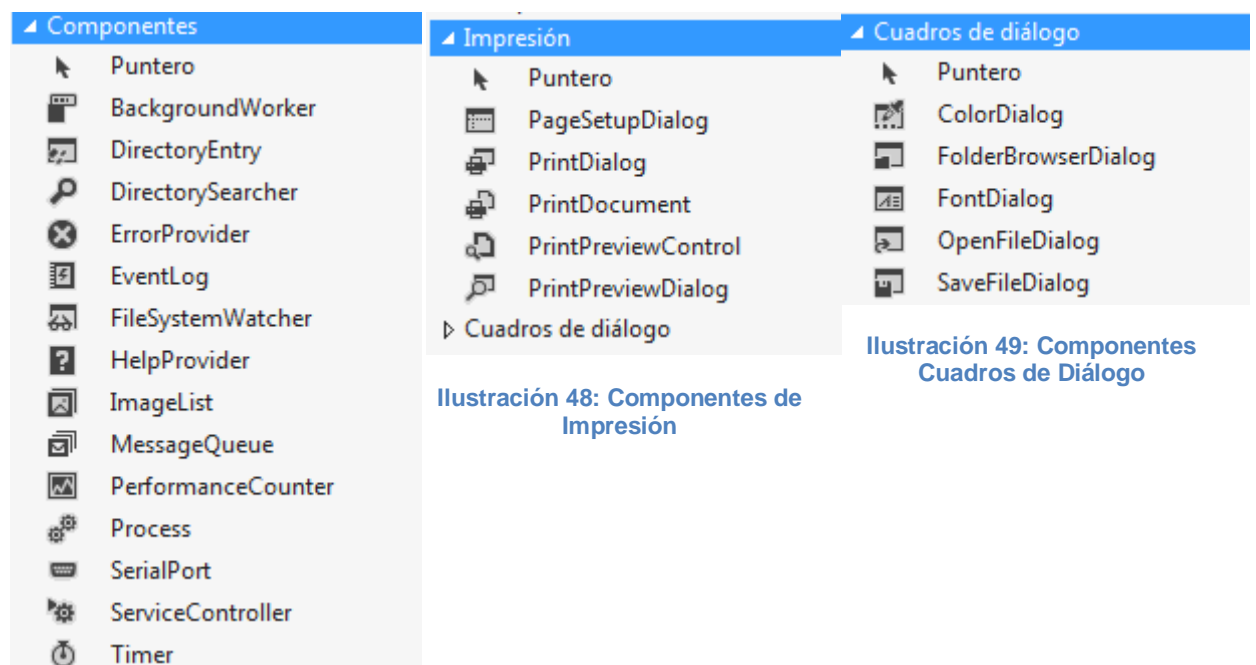


Ilustración 47: Componentes Sencillos

Ilustración 48: Componentes de Impresión

Ilustración 49: Componentes Cuadros de Diálogo

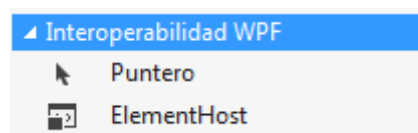


Ilustración 50: Componentes de Interoperabilidad

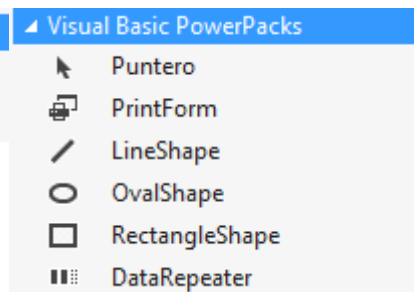


Ilustración 51: Componentes PowerPacks

El Trabajo Fin de Grado no ha utilizado ningún componente de los anteriores.

A modo de ejemplo se incluye un componente de tipo “button” cuya funcionalidad consiste en abandonar una pantalla cuando dicho botón es pulsado:

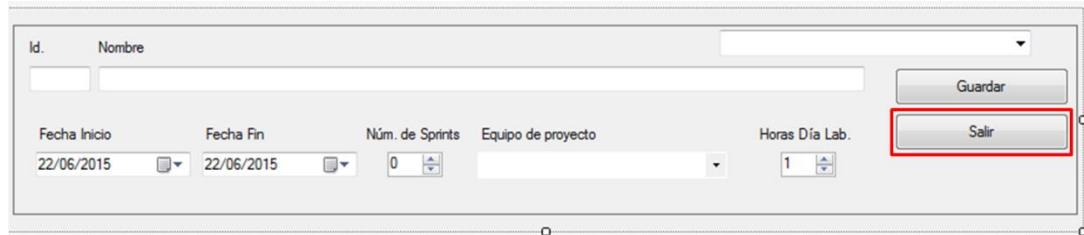


Ilustración 52: Ejemplo de Control Botón "Salir"

En tiempo de diseño se ha dotado del aspecto que se observa incluido el texto “Salir” que se visualiza. Además se le da el nombre “butSal” que permite identificar al control de forma única.

La funcionalidad que satisface es abandonar la creación de un nuevo proyecto al ser pulsado, para ello se inserta el siguiente código en el evento “Click” del control “button” denominado “butSal”

```
Private Sub butSal_Click(sender As Object, e As EventArgs) Handles butSal.Click
    Dim resp As Integer

    resp = MsgBox("Si no ha guardado sus cambios, hágalo antes de salir. De otra forma  
perderá los posibles cambios realizados." & vbNewLine & vbNewLine & "Desea Salir de  
todas formas?", MsgBoxStyle.YesNo + MsgBoxStyle.Question, "SCRUM Wizard")
    If resp = vbYes Then
        cbEquipos.Items.Clear()
        cbIdEquipo.Items.Clear()
        txtlProyecto.Text = ""
        txtNProyecto.Text = ""
        dtFInicio.Value = Now()
        dtFFin.Value = Now()
        nuNSprints.Value = 0
        Me.Close()
    Else
        'seguir aqui
    End If
End Sub
```

Los elementos propios descritos en el apartado anterior se apoyan en el uso de estos componentes.

7.4. Componente OleDb (Microsoft Jet OLEDB 4.0)

OLEDB (Object Linking and Embedding Database) es un API de bajo nivel de Microsoft que permite el acceso a distintas fuentes de datos. Incorpora capacidades de SQL (Structured Query Language) así como de ODBC (Open Database Connectivity), que es una interfaz de Microsoft para acceso a datos.

OLE DB dispone de métodos que facilitan la lectura y escritura de datos. Los principales objetos que constituyen el OLE DB son: la fuente del objeto, la sesión del objeto, comandos del objeto y un objeto de tipo conjunto de filas.

La secuencia que debe seguirse para el uso de OLE DB dentro de una aplicación típica sería:

1. Inicialización de OLE
2. Conexión con la fuente de datos
3. Lanzar el comando
4. Procesar el resultado
5. Liberar el objeto OLE

El motor de base de datos **Microsoft Jet** es el motor de base de datos para los archivos “.mdb” creados con Microsoft Access. Todas las versiones de Jet se incluyen en las ediciones de Microsoft Windows, hasta en Windows 7, y las aplicaciones de 32 bits las utilizan para conectarse a las bases de datos Access de los archivos “.mdb”.

Para la utilización de OleDb en nuestro Trabajo Fin de Grado han sido necesarias una serie de tareas que aseguran el correcto funcionamiento del mismo.

Primero de todo debe indicarse que se va a usar y para ello debemos parametrizar nuestra cadena de conexión con la información de proveedor adecuada:

Nombre	Tipo	Ámbito	Valor
tfgConnectionString	(Cadena de conexión)	Aplicación	Provider=Microsoft.Jet.OLEDB.4.0;Data Source="C:\Users\amfm\Documents\Visual Studio 2012\Projects\TFGAF\ftg.mdb"

Ilustración 53: Cadena de Conexión a SGBD

Los elementos de la cadena son el **proveedor**, que lógicamente es **Microsoft.Jet.OLEDB.4.0**, y la ruta de acceso a la base de datos.

Esta cadena, queda registrada en el fichero “app.config”, que en el siguiente apartado de esta memoria queda descrito en profundidad.

Este componente en Visual Studio se implementa como una clase de los componentes de .NET Framework 4.5 que se ha analizado en profundidad en el apartado anterior de esta memoria.

Esta clase representa una instrucción SQL o un procedimiento almacenado que se ejecutará en un origen de datos. Para el caso concreto de la aplicación construida, representa exclusivamente instrucciones SQL que se ejecutan contra la base de datos Access.

Para poder ser usada es imprescindible la inclusión de su definición en cada uno de los módulos donde se vaya a utilizar. Para ello se emplea la sentencia: "Imports System.Data.OleDb", que se ubica en un ámbito de aplicación todo el formulario en el área de declaraciones, de forma que con una única inclusión esté visible para cualquier elemento del formulario o módulo.

La clase⁶ queda caracterizada como se detalla:

Clase OleDbCommand (Microsoft Dev)	
Constructores	Métodos
OleDbCommand	Cancel
OleDbCommand(Cadena)	Clone
OleDbCommand(Cadena, OleDbConnection)	CreateObjRef
OleDbCommand(Cadena, OleDbConnection, OleDbTransaction)	CreateParameter
	Dispose
Propiedades	Equals(Object)
CommandText	ExecuteNonQuery
CommandTimeout	ExecuteNonQueryAsync
CommandType	ExecuteNonQueryAsync(CancellationToken)
Connection	ExecuteReader
Container	ExecuteReader(CommandBehavior)
DesignTimeVisible	ExecuteReaderAsync
Parameters	ExecuteReaderAsync(CancellationToken)
Site	ExecuteReaderAsync(CommandBehavior)
Transaction	ExecuteReaderAsync(CommandBehavior, CancellationToken)
UpdatedRowSource	ExecuteScalar

⁶ Marcados en amarillo pálido los elementos empleados en la aplicación

	Métodos
Eventos	ExecuteScalarAsync
Disposed	ExecuteScalarAsync(CancellationTok
	GetHashCode
	GetLifetimeService
	GetType
	InitializeLifetimeService
	Prepare
	ToString

Ilustración 54: Elementos Clase OleDbCommand

A modo de ejemplo del uso que se ha realizado de esta clase se adjunta el código de la función “TerminarHistoriasSprint”, cuya misión es pasar a estado “terminado” todas las historias de un sprint cuando éste es cerrado.

```
Public Function TerminarHistoriasSprint(isp As Integer) As Boolean
    Dim cnn As New OleDbConnection(My.Settings.tfgConnectionString)
    Dim cdu As New OleDbConnection(My.Settings.tfgConnectionString)

    Dim trans As OleDbTransaction
    Dim comando As New OleDbCommand()
    Dim Historias() As Integer

    Dim sql As String = ""
    Dim sql1 As String = ""
    Dim sql2 As String = ""

    Dim i, j As Integer

    Dim v As Boolean = True
    'para recorrer
    sql2 = "SELECT PBacklog.IdHistoria FROM PBacklog, SBacklog WHERE "
    sql2 = sql2 & "((PBacklog.IdHistoria = SBacklog.IdHistoria) AND "
    (PBacklog.NumSprint = SBacklog.IdSprint) AND "
    sql2 = sql2 & "(PBackLog.IdProyecto = Sbacklog.IdProyecto) AND "
    (SBacklog.IdProyecto = " & IDPROYECTO & ") AND (Sbacklog.IdSprint = " & isp & ")")

    cdu.Open()
    Dim buscar As New OleDbCommand(sql2, cdu)
    Dim existe As OleDbDataReader = buscar.ExecuteReader

    i = 0
    While existe.Read()
        Historias(i) = CInt(existe.GetValue(0))
        i += 1
    End While
    cdu.Close()
```

```
comando.Connection = cnn
cnn.Open()
trans = cnn.BeginTransaction()
comando.Connection = cnn
comando.Transaction = trans

'actualizacion sbacklog
sql = "UPDATE SBacklog SET Estado = 'Terminado' WHERE ((IdProyecto = " &
IDPROYECTO & ") AND (IdSprint = " & isp & "))"

Try
'ejecuto sql
comando.CommandText = sql
comando.ExecuteNonQuery()

'actualizacion pbacklog
For j = 0 To i - 1
    sql1 = "UPDATE PBacklog SET Estado = 'Terminado' WHERE
((PBacklog.IdProyecto = " & IDPROYECTO & ") AND (PBacklog.NumSprint = " & isp & ")
AND "
    sql1 = sql1 & "(PBacklog.IdHistoria = " & Historias(j) & "))"
'ejecuto sql
comando.CommandText = sql1
comando.ExecuteNonQuery()
Next j

trans.Commit()

Catch ex As Exception
    MsgBox("Ups!!! Problemas" & ex.Message, MsgBoxStyle.Critical, "SCRUM
Wizard")
    trans.Rollback()
    v = False
End Try

Return v
End Function
```

Indicar que se ha utilizado a lo largo de todo el desarrollo SQL estándar, de forma que si en algún momento se decidiese migrar a otro SGBD este aspecto no supusiese ningún problema en cuanto a compatibilidad por uso de funciones extendidas de SQL.

Se puede visitar la dirección (Soporte Microsoft) para la obtención de la última versión disponible del motor para el sistema operativo del que se disponga.

7.5. Fichero de configuración “app.config”

La aplicación se apoya en el fichero de configuración “app.config”, especialmente para lo relacionado con la ubicación de la base de datos.

La situación que resuelve es independizar el funcionamiento de la aplicación de la ubicación donde finalmente resida el SGDB. De esta forma cada instalación podrá adecuarse a la realidad de cada situación.

Será responsabilidad del administrador de los sistemas o quien corresponda dotar de los permisos de acceso suficientes a los usuarios de la aplicación para que puedan acceder sin problema a la ruta donde resida el fichero de configuración.

Se trata de un fichero xml donde se pueden establecer configuraciones que pueda necesitar la aplicación. Para el caso concreto de este Trabajo Fin de Grado solo se ha empleado para recoger la cadena de conexión al SGDB. Dentro de esta cadena de conexión figura la ruta donde residirá la Base de Datos.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <clear />
    <add name="TFGAF.My.MySettings.tfgConnectionString"
connectionString="Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=&quot;C:\Users\amfm\Documents\Visual Studio
2012\Projects\TFGAF\ftg.mdb&quot;;"
    providerName="System.Data.OleDb" />
  </connectionStrings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
</configuration>
```

Dentro del fichero se observa la sección <connectionString> y dentro de ella el elemento “Data Source”. Es en este último donde se ubica la ruta donde reside la base de datos Para el entorno de desarrollo la ruta es: “C:\Users\amfm\Documents\Visual Studio 2012\Projects\TFGAF\ftg.mdb”.

El resto de secciones y entradas no deben modificarse para el corercto funcionamiento de la aplicación.

Este archivo “app.config” pasa a denominarse “SCW.exe.config” – para el caso concreto de este Trabajo Fin de Grado - cuando se genera el ejecutable de la aplicación. Si bien es exactamente el mismo que “app.config”. Es decir, en el entorno de producción solo existe el archivo “SCW.exe.config”.

Por lo tanto, para que la aplicación funcione correctamente en la ubicación de producción deberán seguirse los siguientes pasos:

1. Ubicar el fichero “SCW.exe.config” en la misma ubicación que esté el fichero ejecutable de la aplicación: “SCW.exe”
2. Con un editor de texto (notepad por ejemplo o cualquier otro similar) abrir el fichero “SCW.exe.config” y sustituir la ruta “C:\Users\amfm\Documents\Visual Studio 2012\Projects\TFGAF\ftg.mdb” por la ruta donde se ubique finalmente la base de datos en el entorno de producción.
3. Guardar el archivo modificado

Como ejemplo si la ruta final es [\\ejemplo\ejemplo\](#) el fichero debería contener:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <clear />
    <add name="TFGAF.My.MySettings.ftgConnectionString"
connectionString="Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=&quot;\\ejemplo\ejemplo\ftg.mdb&quot;;"
    providerName="System.Data.OleDb" />
  </connectionStrings>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
</configuration>
```

De esta forma podremos disponer de un repositorio de información centralizado accesible por cualquier usuario.

Otra importante ventaja de este archivo es que si en un futuro se decidiese utilizar otra base de datos distinta a Access solo habría que cambiar el elemento “Provider” y “providerName” de la sección <connectionString> por el proveedor y nombre del proveedor que se fuese a utilizar.

7.6. SGDB – MS Access (mdb)

En el apartado 5.3. Metodología empleada de esta memoria se ha detallado la motivación para la elección de Microsoft Access como SGDB.

En este proyecto, MS Access, en tiempo de desarrollo, solo se ha empleado para la creación de la base de datos en formato “.mdb”, pues el driver de oledb,

Microsoft.**Jet.OLEDB.4.0**, está especializado en el manejo de bases de datos en esta versión y esto hace que MS Access no sea necesario.

Una vez creada la base de datos no ha sido necesaria la utilización de MS Access. En el entorno de producción no es necesario disponer de MS Access.

El bloqueo de los registros es manejado directamente por el driver, para lo cual genera un fichero con el mismo nombre de la base de datos y extensión “.ldb”.

El modelo de datos que se emplea se describe seguidamente:

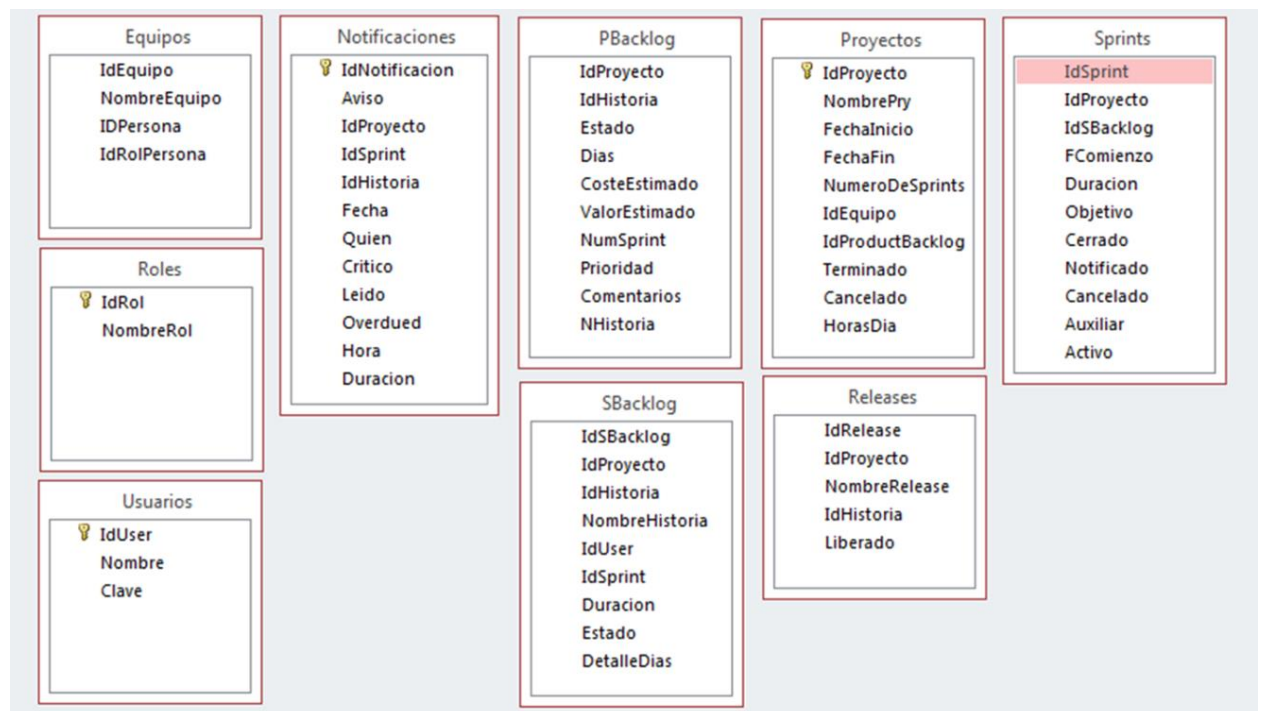


Ilustración 55: Modelo de Datos de la solución

El Anexo II: Detalle de las tablas de la aplicación incluye el detalle de cada una de las tablas empleadas en la aplicación.

7.7. Tratamiento de Errores en “Scrum Wizard”

La aplicación se ha diseñado para capturar todos los errores que se pudiesen producir e informar de los mismos proporcionando el máximo de información disponible posible.

Salvo error inesperado la aplicación no debería producir fallos irreversibles. En todo caso si se produjesen se aconseja cerrar la aplicación y volver a ejecutar.

La forma en la que se interceptan los errores sigue el mismo patrón a lo largo de toda la aplicación:

Try

' AQUÍ SE COLOCARIA EL CÓDIGO

Catch ex As Exception

 MsgBox("Ups!!! Problemas " & ex.Message, MsgBoxStyle.Critical, "SCRUM Wizard")

End Try

8. Evaluación

La naturaleza de un Trabajo Fin de Grado de tipo individual apoyado por el Director ha marcado decisivamente la forma de validar el trabajo realizado, desde un punto de vista conceptual, de diseño, de construcción y prueba.

Para realizar esta validación en el contexto descrito, el autor presentó un plan de trabajo al Director del Trabajo Fin de Grado para su revisión y validación. Este plan de trabajo reflejaba las distintas fases del Trabajo Fin de Grado, fases de tipo formal (aquellas exigidas por la asignatura) y otras de tipo trabajo (aquellas necesarias para el correcto avance del proyecto).

El cronograma consensuado y aprobado del plan de trabajo se expresa seguidamente:

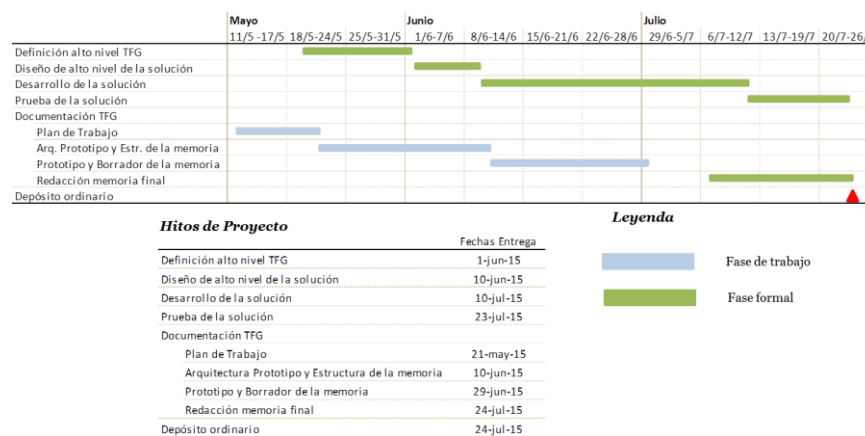


Ilustración 56: Cronograma TFG

Cada Fase ha tenido la siguiente estructura:

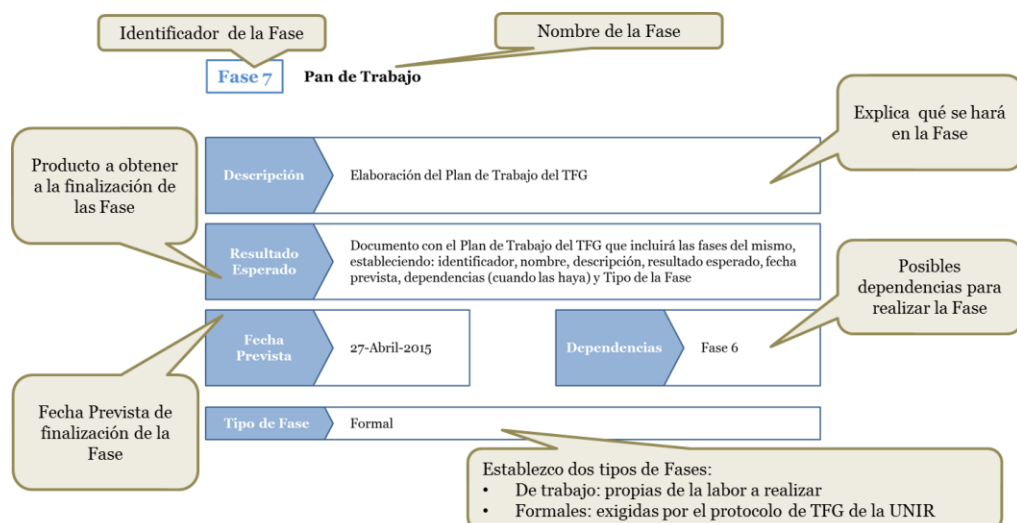


Ilustración 57: Estructura Típica de una Fase

De esta forma todas las fases del Trabajo Fin de Grado, conforme a su calendario y compromisos establecidos, han sido presentadas (sus entregables) en tiempo y forma al Director del Trabajo Fin de Grado para su validación. Todas las sugerencias del Director han sido consideradas e incluidas en el Trabajo Fin de Grado. Y cuando finalmente han sido completamente aceptadas por éste se han dado por validadas y cerradas.

Las fases más complejas, en cuanto a cómo demostrar, la funcionalidad y avance, se han satisfecho mediante la generación de documentación escrita y audiovisual (vídeo del prototipo).

Lógicamente antes de presentar ningún resultado el autor ha realizado concienzudas pruebas, especialmente en la fase de construcción, no solo en el entorno de desarrollo sino también en otros entornos disponibles para el autor. Además se solicitó a terceras personas la prueba de la aplicación construida, para evitar lo rutinario en la prueba.

Cuando estas pruebas satisficieron plenamente al autor fueron presentados al Director los resultados para su validación final.

También se han realizado pruebas simulando distintos entornos productivos posibles: SGBD en misma máquina, en máquina distinta, en ubicación de red, Incluso pruebas de concurrencia mediante coordinación de distintos probadores de la aplicación.

Las pruebas funcionales no solo han seguido el camino de prueba más habitual sino también aquellos caminos de prueba menos probables o más infrecuentes. Para lograr este objetivo, complicado si solo se realiza por el autor, solicité ayuda a personas de mi entorno laboral que realizaran sus propias pruebas independientemente del resto de probadores.

9. Conclusión y Trabajos Futuros

La conclusión personal obtenida es que esta metodología que pone el foco en el usuario y en sus necesidades por delante de todo, tiene mayores probabilidades de éxito en el mundo real que las metodologías más tradicionales que cuando entregan resultados el usuario “casi” se ha olvidado de lo que había pedido.

Si bien es cierto igualmente, al menos basado en mi experiencia, que no todos los usuarios quieren involucrarse, pues en muchas ocasiones, esto significa comprometerse y no siempre esto es posible o deseable.

Algunas líneas de potencial evolución de la aplicación que se han identificado son:

- Delegación de funciones entre el ST
- Permitir cambios en proyectos con PB ya creado e iniciado.
- Informes de seguimiento más sofisticados
- Herramienta visual de tipo “kanban” para facilitación de reuniones
- Manejo de tareas por historia, proporcionando más granularidad a la labor a realizar
- Uso de distintos calendarios. Mejora en el manejo de festivos por parte de la aplicación
- Ajustes automáticos en base a la resolución de las pantallas donde se utilice la aplicación

Anexo I: Procedimiento de Instalación

Para la instalación de la aplicación deberán seguirse los siguientes pasos:

- Deberá tener claro donde quiere instalar cada componente de la aplicación
- Deberá asegurarse de que los usuarios tienen suficientes permisos de acceso a las ubicaciones en las que decida instalar los componentes de la aplicación
- Los componentes de la aplicación a instalar son:
 - SCW.exe La aplicación
 - SCW.exe.config El fichero de configuración de la aplicación
 - Ftg.mdb El SGBD
- “SCW.exe” y “SCW.exe.config” deben ubicarse juntos
- Si considera que el usuario puede manipular “SCW.exe.config” protéjalo para evitar su manipulación
- Ubique “ftg.mdb” donde desee y anote la ruta
- Modifique la ruta incluida en “SCW.exe.config” por la anotada (Vea el apartado 7.5. Fichero de configuración “app.config” de esta memoria si tiene dudas)
- Se incluye también el fichero de instalación del Microsoft .NET Framework 4.5 por si tuviese que instalarlo, aunque se recomienda se descargue la última versión disponible del mismo de la página de Microsoft.

Nótese que la base de datos “ftg.mdb” inicialmente se proporciona vacía de contenido, salvo por el usuario “admin” y los roles comentados en esta memoria.

Anexo II: Detalle de las tablas de la aplicación

Este anexo pormenoriza de cada una de las tablas empleadas⁷:

Equipos		
Campo	Descripción	Tipo
IdEquipo	Identificador único de equipo de trabajo (DT)	Entero
NombreEquipo	Nombre que se da al equipo de trabajo (DT)	Carácter
IdPersona	Identificador único de usuario	Entero
IdRolPersona	Identificador de Rol de la Persona	Entero

Tabla 7: Detalle de la Tabla Equipos

Roles		
Campo	Descripción	Tipo
IdRol	Identificador único de rol que puede desempeñar un usuario	Entero
NombreRol	Nombre del Rol	Carácter

Tabla 8: Detalle de la Tabla Roles

Usuarios		
Campo	Descripción	Tipo
IdUser	Identificador único de usuario	Entero
Nombre	Nombre del usuario	Carácter
Clave	Contraseña del usuario	Carácter

Tabla 9: Detalle de la Tabla Usuarios

Notificaciones		
Campo	Descripción	Tipo
IdNotificacion	Identificador único de notificación	Entero
Aviso	Texto de la notificación	Carácter
IdProyecto	Identificador único de Proyecto	Entero
IdHistoria	Identificador único de Historia	Entero
IdSprint	Identificador único de Sprint	Entero
Fecha	Fecha en la que el aviso caduca	Fecha
Quien	Identificador único de usuario al que va dirigido el mensaje	Entero
Critico	Determina si el mensaje es crítico o no lo es	Boolean
Leido	Indica si el mensaje ha sido marcado como leído por	Boolean

⁷ Marcado en amarillo pálido las PK de las tablas. Cuando se marca más de una celda significa que la PK es la concatenación de ellas

Notificaciones		
Campo	Descripción	Tipo
	el usuario o no	
Overdued	Cuando el aviso ya ha caducado	Boolean
Hora	Hora a la que se producirá la reunión cuando la notificación es de reunión	Fecha
Duración	Duración de la reunión cuando la notificación es de reunión	Fecha

Tabla 10: Detalle de la Tabla Notificaciones

SBacklog (Sprint Backlog)		
Campo	Descripción	Tipo
IdSbacklog	Identificador único de SB	Entero
IdHistoria	Identificador único de Historia	Entero
IdProyecto	Identificador único de Proyecto	Entero
IdUSer	Identificador único de usuario	Entero
IdSprint	Identificador único de Sprint	Entero
Duracion	Duración real del sprint	Entero
Estado	Estado de la historia	Carácter
DetalleDias	Contiene el esfuerzo en horas por día. En un formato: [XX/YY], donde XX es el día e YY las horas de ese día	Carácter
NombreHistoria	El nombre que se ha dado a la Historia	Carácter

Tabla 11: Detalle de la Tabla Sprint Backlog

PBacklog (Product Backlog)		
Campo	Descripción	Tipo
IdProyecto	Identificador único de Proyecto	Entero
IdHistoria	Identificador único de Historia	Entero
NHistoria	Nombre de la historia	Carácter
Dias	Duración estimada de la historia	Entero
Estado	Estado de la historia	Carácter
CosteEstimado	Coste estimado de la historia en euros	Entero
ValorEstimado	Valor estimado que aporta la historia	Entero
NumSprint	Sprint en el que se realiza la historia	Entero
Prioridad	Determina la prioridad de la historia	Entero
Comentarios	Campo libre que permite introducir lo que se desee	Carácter

Tabla 12: Detalle de la Tabla Product Backlog

Sprints		
Campo	Descripción	Tipo
IdSprint	Identificador único de Sprint	Entero
IdProyecto	Identificador único de Proyecto	Entero
FComienzo	Fecha en la que comienza el Sprint	Fecha
Duracion	Duración del sprint	Entero
Objetivo	Almacena el objetivo del sprint	Carácter
IdSbacklog	Identificador único de SB	Entero
Cerrado	Indica si un sprint está cerrado	Boolean
Notificado	Indica si se ha realizado la notificación de las actividades del sprint	Boolean
Cancelado	Indica si el Sprint ha sido cancelado	Boolean
Auxiliar	Campo auxiliar cuyo valor determina qué notificaciones se han realizado. Se debe interpretar como un valor en decimal de 4 bits. Donde: 8 = SPM, 4 = DS, 2 = SRM y 1 = SRetroM. La combinación de ellos indica qué notificaciones se han realizado y cuáles están aún pendientes.	Entero
Activo	Indica si el sprint es el activo, es decir, el que está en curso en este momento	Boolean

Tabla 13: Detalle de la Tabla Sprints

Proyectos		
Campo	Descripción	Tipo
IdProyecto	Identificador único de Proyecto	Entero
NombrePry	Nombre que se da al proyecto	Carácter
FechaInicio	Fecha de inicio del proyecto	Fecha
FechaFin	Fecha de finalización del proyecto	Fecha
NumDeSprints	Número de sprints del proyecto	Entero
Terminado	Indica si el proyecto está terminado	Boolean
Cancelado	Indica si el proyecto se ha cancelado	Boolean
HorasDia	Indica el número de horas al día que se trabaja por defecto	Entero
IdEquipo	Identificador único de equipo de trabajo (DT)	Entero
IdProductBacklog	Contiene el identificador de PB	Entero

Tabla 14: Detalle de la Tabla Proyectos

Releases		
Campo	Descripción	Tipo
IdRelease	Identificador único de la Release del proyecto	Entero

Releases		
Campo	Descripción	Tipo
IdProyecto	Identificador único de Proyecto	Entero
NombreRelease	Nombre que se da a la Release	Carácter
IdHistoria	Identificador único de Historia	Entero
Liberado	Indica si liberado o no	Boolean

Tabla 15: Detalle de la Tabla Releases

Referencias y enlaces

- Agile Manifesto. (s.f.). *Manifiesto Ágil*. Recuperado el 2015, de <http://agilemanifesto.org/iso/es/manifesto.html>
- Microsoft .Net. (s.f.). *Información General acerca de .NET Framework*. Recuperado el 2015, de [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)
- Microsoft Dev. (s.f.). *Developer Network*. Recuperado el 2015, de [https://msdn.microsoft.com/es-es/library/system.data.oledb.oledbcommand\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.data.oledb.oledbcommand(v=vs.110).aspx)
- Microsoft. (s.f.). *Download Center*. Recuperado el 2015, de <https://www.microsoft.com/es-ES/download/details.aspx?id=34673>
- Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. Prentice Hall.
- Scrum Alliance. (s.f.). *Scrum Alliance*. Recuperado el 2015, de <https://www.scrumalliance.org/>
- Scrum Methodology. (s.f.). *Scrum Methodology*. Recuperado el 2015, de <http://scrummethodology.com/>
- Scrum.org. (s.f.). *Scrum.Org.The home of Scrum*. Recuperado el 2015, de <https://www.scrum.org/>
- Soporte Microsoft . (s.f.). *Soporte de Microsoft*. Recuperado el 2015, de <https://support.microsoft.com/en-us/kb/239114>
- Sutherland, J., & Schwaber, K. (Julio de 2013). *La Guía Definitiva de Scrum: Las reglas del Juego*. Recuperado el 2015, de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Varios. (s.f.). *Manifiesto Ágil*. Recuperado el 2015, de <http://agilemanifesto.org/iso/es/>