



Universidad de Valladolid

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

CREACIÓN DE UN AULA VIRTUAL EN SHAREPOINT 2016

Autor

Pablo Castellanos Duque

Tutor
Pablo de la Fuente Redondo

Agradecimientos

En primer lugar, me gustaría dar las gracias a una persona que, aunque ya no está entre nosotros, sigue presente en todos y cada uno de los momentos importantes en mi vida. Con este proyecto termino mi carrera y es gracias a ti, Mamá. Gracias por haberme cuidado y haberme convertido en la persona que soy hoy. Estoy seguro de que allá donde estés, te sentirás orgullosa de mí tanto como lo estoy yo de ti. Gracias, una vez más, por todo lo que me has dado.

A mi pequeña Mar, la chica que me acompaña desde hace ya más de seis años y que me ha demostrado que siempre está ahí cuando la necesito. Gracias por cada momento a tu lado.

A mi abuela Nina, mi abuela Espe, mi padre David y mi hermano Mario, por cuidarme y apoyarme desde que era pequeño y por todo lo que os habéis preocupado por mí.

A mis tíos y mis primos, por haber estado ahí en las buenas y, sobre todo, en las malas. Gracias por estar siempre dispuestos a echar una mano cuando hace falta.

A mis amigos, especialmente a Josemi, Manuel, Héctor y David, gracias por la dura tarea de aguantarme todos estos años.

A mis compañeros de trabajo, especialmente a Carlos, Pepe y Pablo, por haberme tratado como uno más y haber tenido la paciencia de enseñarme todo lo que sé.

A todo el resto de personas que me olvidé de nombrar. Gracias a todos!

Resumen

Mi trabajo como desarrollador en SharePoint en los primeros meses de mi carrera profesional me hizo plantearme una pregunta: ¿Es realmente SharePoint una alternativa viable para una universidad? Este proyecto es fruto de la búsqueda de esa respuesta.

El proyecto es una prueba de concepto para comprobar cómo podría implementarse un campus virtual de una universidad, utilizando SharePoint 2016. A través de una pequeña granja con soluciones a medida, se tratará de mostrar una propuesta para un sistema de este tipo, analizando su funcionamiento y las ventajas e inconvenientes que este puede tener.

Índice general

Agradecimientos	I
Resumen	II
1. Introducción	1
1.1. Estructura de la memoria	2
1.2. Objetivos del proyecto	2
1.3. Motivaciones del proyecto	3
2. SharePoint	4
2.1. Descripción de la herramienta	5
2.2. Historia del producto	7
2.3. Modelos de implementación	8
2.3.1. SharePoint On-premises	8
2.3.2. SharePoint Online	9
2.3.3. Implementaciones híbridas	9
2.4. Arquitectura	10
2.4.1. Roles de servidor y MinRole	10
2.5. Requisitos mínimos	12
2.6. Modelo de objetos	13
2.6.1. Jerarquía de sitios	13
2.6.2. Columnas y tipos de contenido	13
2.6.3. Listas y bibliotecas	15
2.6.4. Permisos	16
2.7. Desarrollo de soluciones	16
3. Escenario y alternativas	18
3.1. Descripción del escenario	19
3.2. Análisis del sistema actual	21
3.3. Análisis del sistema propuesto	22
3.4. Comparativa de sistemas	24
4. Planteamiento de la solución	27
4.1. Alcance de la propuesta	28
4.2. Infraestructura y herramientas	29
4.3. Estructura de la solución	30
4.4. Organización de la información	31
4.4.1. Información a almacenar	31
4.4.2. La información en SharePoint	34
4.4.3. Listas, tipos de contenido y columnas	36

4.4.4.	Niveles de permisos	37
5.	Planificación	38
5.1.	Metodología utilizada	39
5.2.	Product Backlog	39
5.3.	Planificación inicial	41
5.4.	Ejecución de la planificación	42
5.4.1.	Sprint 1	42
5.4.2.	Sprint 2	43
5.4.3.	Sprint 3	44
5.4.4.	Sprint 4	45
5.4.5.	Sprint 5	46
5.4.6.	Sprint 6	47
5.4.7.	Sprint 7	48
5.4.8.	Sprint 8	49
6.	Provisionamiento y configuración	50
6.1.	Script creación de colecciones de sitios	51
6.2.	Script creación de sitios	51
6.3.	Script creación de columnas	52
6.4.	Script de creación de tipos de contenido	52
6.5.	Script de creación de listas	53
6.6.	Configuración de servicios	53
6.6.1.	Servicio de búsqueda	53
6.6.2.	Servicio de perfiles de usuario	54
6.6.3.	Servicio de metadatos administrados	55
7.	Desarrollo de soluciones	56
7.1.	Webpart Enlaces asignaturas	57
7.1.1.	Requisitos funcionales	57
7.1.2.	Funcionalidad utilizada	57
7.1.3.	Diseño técnico	58
7.1.4.	Precondiciones y control de excepciones	59
7.1.5.	Batería de pruebas	60
7.2.	Webpart Mis calificaciones	61
7.2.1.	Requisitos funcionales	61
7.2.2.	Funcionalidad utilizada	62
7.2.3.	Diseño técnico	63
7.2.4.	Precondiciones y control de excepciones	64
7.2.5.	Batería de pruebas	65
7.3.	Webpart Listado de grupos	66
7.3.1.	Requisitos funcionales	66
7.3.2.	Funcionalidad utilizada	66
7.3.3.	Diseño técnico	67
7.3.4.	Precondiciones y control de excepciones	69
7.3.5.	Batería de pruebas	70
7.4.	Webpart Propiedades entrega	71

7.4.1.	Requisitos funcionales	71
7.4.2.	Funcionalidad utilizada	71
7.4.3.	Diseño técnico	72
7.4.4.	Precondiciones y control de excepciones	73
7.4.5.	Batería de pruebas	73
7.5.	Webpart Gestor de tareas	74
7.5.1.	Requisitos funcionales	74
7.5.2.	Funcionalidad utilizada	74
7.5.3.	Diseño técnico	75
7.5.4.	Precondiciones y control de excepciones	77
7.5.5.	Batería de pruebas	78
7.6.	Webpart Calificar	79
7.6.1.	Requisitos funcionales	79
7.6.2.	Funcionalidad utilizada	79
7.6.3.	Diseño técnico	80
7.6.4.	Precondiciones y control de excepciones	82
7.6.5.	Batería de pruebas	83
7.7.	Webpart Creación Entregas	84
7.7.1.	Requisitos funcionales	84
7.7.2.	Funcionalidad utilizada	84
7.7.3.	Diseño técnico	85
7.7.4.	Precondiciones y control de excepciones	87
7.7.5.	Batería de pruebas	88
8.	Conclusiones y trabajo futuro	89
8.1.	Conclusiones	90
8.1.1.	Conclusiones a nivel de proyecto	90
8.1.2.	Conclusiones personales	91
8.2.	Trabajo futuro	92
9.	Bibliografía	93
Anexos		95
Anexo I.	Script de creación de colecciones de sitio	95
Anexo II.	Script de creación de sitios	96
Anexo III.	Script de creación de columnas	98
Anexo IV.	Script de creación de tipos de contenido	101
Anexo V.	Script de creación de listas	105
	Contenido del CD	108

Índice de cuadros

2.1. Topologías recomendadas	11
2.2. Topología mínima MinRole	11
2.3. Requisitos hardware mínimos standalone	12
2.4. Requisitos mínimos granja	12
4.1. Funcionalidades de los alumnos	28
4.2. Funcionalidades de los alumnos	29
5.1. Tareas Sprint Inicial	41
5.2. Fechas Sprint 1	42
5.3. Sprint 1	42
5.4. Fechas Sprint 2	43
5.5. Sprint 2	43
5.6. Fechas Sprint 3	44
5.7. Sprint 3	44
5.8. Fechas Sprint 4	45
5.9. Sprint 4	45
5.10. Fechas Sprint 5	46
5.11. Sprint 5	46
5.12. Fechas Sprint 6	47
5.13. Sprint 6	47
5.14. Fechas Sprint 7	48
5.15. Sprint 7	48
5.16. Fechas Sprint 7	49
7.1. Elementos web: Webpart Enlaces	58
7.2. Elementos web: Webpart Calificaciones	63
7.3. Elementos web: Webpart Listado Grupos	67
7.4. Elementos web: Webpart Propiedades Entrega	72
7.5. Elementos web: Gestor Tareas	75
7.6. Elementos web: Webpart Calificar	80
7.7. Elementos web: Webpart Calificar	85

Capítulo 1

Introducción

A la hora de definir un proyecto de estas características, donde es el propio autor quien define el alcance y los requisitos del mismo y quien establece las metas a las que llegar, es importante tener en cuenta qué es lo que se quiere lograr con el proyecto, qué razones llevan a su ejecución y qué se espera conseguir con él.

En esta breve sección en la que se comenzará describiendo la estructura de la propia memoria, se tratará de dar respuesta a esos interrogantes, estableciendo los objetivos del proyecto y exponiendo las motivaciones del mismo.

1.1. Estructura de la memoria

La memoria está compuesta por 8 capítulos de contenido, precedidas por los agradecimientos y un breve resumen del trabajo:

- **Introducción:** Sección introductoria donde se describen los objetivos del proyecto.
- **SharePoint:** Sección en la que se describirá la herramienta utilizada.
- **Escenario y alternativas:** Sección en la que se describirá el escenario del problema y las alternativas de solución.
- **Planteamiento de la solución:** Sección en la que se describirá el alcance del proyecto, y la solución que se propondrá.
- **Planificación:** Sección en la que se describirá la planificación utilizada, la planificación inicial y el seguimiento de la planificación a lo largo del proyecto.
- **Desarrollo de scripts:** Sección en la que se describirán los scripts desarrollados en las distintas etapas del desarrollo.
- **Desarrollo de soluciones:** Sección en la que se describirán las soluciones desarrolladas, con una sección por desarrollo. Para cada desarrollo, se elaborará una batería de pruebas individual y, posteriormente, se realizará una batería de pruebas de integración para comprobar el funcionamiento de todos los componentes de forma conjunta.
- **Conclusiones y trabajo futuro:** Sección en la que se expondrán las conclusiones obtenidas en el trabajo y algunas de las posibilidades que ofrece SharePoint.

La memoria se completará con dos secciones adicionales:

- **Bibliografía:** Sección de referencias bibliográficas.
- **Anexos:** Documentación adicional que no es lo suficientemente importante como para considerarlo una sección de la memoria, pero que merece la pena incluir como documentación adicional.

1.2. Objetivos del proyecto

El trabajo de fin de grado consistirá en la creación de un entorno de colaboración entre alumnos y profesores, utilizando SharePoint Server 2016. Este entorno estará basado en la experiencia obtenida trabajando con los entornos Moodle tanto de la Universidad de Valladolid (Campus Virtual UVa) como de la Escuela Técnica

Superior de Ingeniería Informática (Aula Virtual ETSII).

El principal objetivo del proyecto es demostrar que SharePoint es una herramienta que proporciona una alternativa viable para la utilización en entornos educativos, específicamente en el caso de un aula virtual. El proyecto tratará de mostrar, mediante una implementación a pequeña escala, cómo podría funcionar un sistema de este tipo.

Como objetivo secundario, se tratará de mostrar cómo reaccionaría SharePoint en caso de problemas típicos de este tipo de sistemas, tales como la disponibilidad, escalabilidad, seguridad, eficiencia... unido a un aspecto que se considera fundamental en sistemas de este tipo y que cada día resulta más importante: el tratamiento de usuarios y sus datos personales.

Como objetivo adicional, se tratará de demostrar que una buena estructuración del sistema unida al uso de buenas prácticas como filosofía, permiten crear sistemas usables, cuyo coste de mantenimiento y evolución es muy inferior al de un sistema mal planteado. Este objetivo estará presente durante todo el planteamiento de la solución.

1.3. Motivaciones del proyecto

Gran parte de los proyectos a los que un desarrollador de software tiene que enfrentarse, son proyectos de mantenimiento o mejora sobre sistemas ya creados, que condicionan el proceso a unas bases, principios o restricciones que impiden, en muchos casos, poder trabajar con total libertad. El desarrollo debe adaptarse a unas condiciones que no siempre son las más adecuadas y que, habitualmente, no pueden ser modificadas por cuestiones de alcance.

Un mal planteamiento en el sistema o una forma incorrecta de ejecutar el mismo, genera lo que coloquialmente se denomina efecto bola de nieve, desarrollando código cada vez menos mantenible y con prácticas menos recomendadas.

Una de las principales causas de una mala ejecución en el desarrollo de software, proviene del desconocimiento del sistema sobre el que se trabaja y el enfoque de los desarrolladores hacia el qué y no hacia el cómo, o dicho de otro modo, la búsqueda del cumplimiento de los objetivos sin prestar mayor atención a la calidad del trabajo producido.

El objetivo personal marcado para este proyecto, es el de obtener una visión mucho más amplia sobre lo que SharePoint representa, participando activamente en todas las fases relevantes dentro de un proyecto de desarrollo, tratando de entender el por qué de las decisiones tomadas y las implicaciones que estas pueden tener.

Capítulo 2

SharePoint

La información y su gestión, son dos de los recursos más críticos que posee una organización, pues representan el conocimiento y los procesos de la misma. Cómo es creado, gestionado y compartido y cómo los usuarios colaboran a través de la utilización de ese contenido, ha evolucionado desde la gestión de contenido empresarial tradicional, a los servicios de contenido dinámico actuales. Las plataformas de servicios de contenido son la siguiente etapa de la gestión de contenido empresarial y representan un cambio, de sistemas y repositorios autocontenidos, a servicios abiertos. Para obtener un incremento en la productividad de la organización, se debe balancear la manejabilidad de estas plataformas con su facilidad de uso. Es en este contexto donde Microsoft apuesta por SharePoint.

En este capítulo se tratará de dar una visión global del producto, con especial atención a los puntos que se consideran claves para el proyecto.

2.1. Descripción de la herramienta

Obtener una definición clara y concisa sobre qué es SharePoint es una tarea que, a día de hoy, ni siquiera el propio Microsoft ha logrado. Lo que en su origen fue una plataforma de colaboración empresarial que permitía la construcción de intranets y la automatización de procesos de negocio orientada a empresas, hoy es una plataforma mucho más abierta y con muchas más funcionalidades. Este hecho provoca la existencia de dos definiciones complementarias: SharePoint como producto software y SharePoint como entorno de colaboración.

- Como entorno de colaboración, SharePoint ofrece servicios como la plataforma esencial de gestión de contenido de Office 365. Es un conjunto de herramientas que incluyen gran cantidad de funcionalidades, cuya finalidad dentro de una organización es proporcionar a sus usuarios los recursos necesarios para:
 - Trabajar en un entorno colaborativo, tanto entre usuarios de la misma organización, como usuarios externos a ella.
 - Compartir información de forma organizada y estructurada, a la par que ágil, flexible, segura y escalable.
 - Gestionar y presentar los contenidos.
 - Gestionar los procesos de la organización.

- Como producto, SharePoint es una herramienta de colaboración empresarial basada en tecnologías web, que proporciona características completas para la administración de contenidos, implementa procesos empresariales y proporciona acceso a información esencial para los objetivos y procesos organizativos. Proporciona una plataforma integrada para planear, implementar y administrar aplicaciones de intranet, extranet e Internet.

La primera de las definiciones corresponde a una visión más purista de la herramienta, muy asociada con el propósito inicial de la misma, mientras que la segunda es una definición tan abierta como se pueda imaginar. Lo que comenzó siendo una herramienta con un propósito muy definido, centrado en la creación de intranets para empresas, ha evolucionado hasta su adaptación para intranets, extranets e incluso hasta portales web públicos de muchas organizaciones con o sin ánimo de lucro, tales como universidades, ONGs. . .

Aunque existiesen versiones previas, la primera de las que realmente se identifica con SharePoint y que marcó el verdadero punto de partida del producto fue SharePoint Portal Server 2003.

SharePoint está presente en más de 300.000 organizaciones en todo el mundo, entre las que se encuentra el 85 % de las empresas que forman parte del Fortune 500.

Durante el año 2016, el número de usuarios activos de SharePoint se ha incrementado en un 90 %, y los datos almacenados en Sharepoint Online lo han hecho en un 250 %. El cuadrante mágico de Gartner, ha reconocido a Microsoft como líder en la categoría de “Plataforma de servicios de contenido” en el año 2017. Además de ser uno de los únicos tres líderes en dicha categoría, Microsoft está considerado como el líder con mayor capacidad de ejecución.

Los principales factores que proporcionan este liderazgo a SharePoint son:

- **Simplicidad:** SharePoint es rápido, con una interfaz simple y sencilla de configurar, lo que proporciona al usuario una herramienta de contenidos sencilla, pero muy potente.
- **Servicios de Office 365:** SharePoint está incluido en la capa de servicios de contenido de Office 365, lo que brinda a los usuarios un conjunto consistente de funcionalidades a través de aplicaciones que se complementan, creando una experiencia de usuario que es capaz de cubrir gran parte de las necesidades que éstos tienen, junto con las garantías en materia de seguridad y control que proporciona una compañía como Microsoft.
- **Soporte amplio de tipos de archivo:** Junto con los archivos de Office, SharePoint puede almacenar cualquier archivo y soporta la vista de más de 270 tipos de archivos, entre los que se encuentran Adobe Photoshop (PSD), Illustrator (AI), Acrobat (PDF), así como video, formatos 3D, e imágenes DICOM.
- **Búsqueda:** La búsqueda de SharePoint es inteligente, rápida y sencilla, permitiendo encontrar y filtrar resultados que incluyan todo tipo de contenido, como archivos, sitios, gente, noticias y elementos de lista. SharePoint incluso indexa objetos, texto y escritura a mano dentro de otras imágenes.
- **Escalabilidad:** SharePoint está diseñado para clientes que van desde pequeñas empresas a organización con cientos de miles de usuarios, con una capacidad máxima de hasta 30 billones de documentos.
- **Seguridad y confianza en TI:** SharePoint aprovecha los mecanismos de seguridad de Microsoft como “Gobierno de datos avanzados para retención y gestión de registros”, “Prevención de pérdida de datos (DLP), eDiscovery, y Customer Key con controles consistentes a través de todo Office 365.
- **Flexibilidad de implementación:** Los clientes pueden elegir su modelo de implementación entre on-premises, híbrida o en la nube.
- **Liderazgo de nube y cumplimiento de normas:** Formado por más de 100 centros de datos globales y la red global de Microsoft, combinados con los estándares de cumplimiento de normas, que incluyen ISO 27001, FISMA, y EU Model Clauses.

2.2. Historia del producto

A pesar de los intentos previos de Microsoft por crear su herramienta de colaboración, no fue hasta el año 2001 cuando se lanzó al mercado la primera versión de SharePoint, Microsoft SharePoint Portal Server 2001. Las primeras versiones fueron gratuitas y se utilizaron como medio para recoger los requisitos de negocio de las empresas que decidieron apostar por ellas, hasta que, en su versión 2003, comenzaron a distribuirse las primeras versiones comerciales con una licencia de pago.

La primera versión realmente relevante, llegó en el año 2007, donde se introdujeron los pilares fundamentales en los que se apoyan las versiones posteriores. Cabe destacar entre ellas, la creación de Tipos de contenido para representar tipos de datos con una estructura común, la aparición de SharePoint Designer como herramienta de edición de contenido web, y el inicio de la integración de SharePoint como parte del paquete de Office.

La versión Microsoft SharePoint Server 2010, fue la versión en la que realmente SharePoint triunfó como plataforma de colaboración empresarial. Las claves de dicho éxito fueron la creación del concepto de la cinta de opciones o ribbon, la creación de la API de PowerShell para la administración, y el acceso a la API desde cliente. De forma adicional, se incluyeron las primeras funcionalidades sociales de la plataforma.

En paralelo, la creciente demanda de servicios en la nube dio origen, en 2011, a la versión SharePoint Online, que proporcionaba los mismos servicios que la versión On-Premises, pero con una infraestructura totalmente gestionada por Microsoft a través de Azure. En un principio, la elevada inversión en infraestructura y desarrollos a medida realizada por las empresas, unida con la elevada limitación en las personalizaciones hizo que esta alternativa no fuese muy popular, y pareciese más bien un intento fallido, aunque posteriormente, con la creación del paquete Office 365, esta opción cada vez cobra una mayor importancia.

Como en todo gestor documental, era necesario un buen motor de búsqueda, con una gran eficiencia a la hora de localizar los contenidos. SharePoint no fue capaz de resolver este aspecto de una forma sólida, por lo que varias empresas optaron por desarrollar sus propios motores de búsqueda para SharePoint, con un coste de licenciamiento adicional. Entre ellos, el motor de búsqueda FAST fue el que obtuvo mayor aceptación por parte de los usuarios, por lo que Microsoft tomó la decisión de adquirir la empresa que lo desarrollaba, y añadirlo como parte de su versión Microsoft SharePoint Server 2013. Unido a ello, esta versión proporciona soporte de navegadores como Chrome o Firefox, más características sociales, integración con OneDrive y el concepto de "Drag and drop". Esta versión es la más popular en la actualidad.

Posteriormente y sin grandes modificaciones reseñables, surge SharePoint 2016, cuyo principal objetivo es dar una mayor consistencia a todos los conceptos introducidos en versiones anteriores y abrir camino a la nube, proporcionando la posibilidad

de creación de entornos híbridos entre SharePoint On-Premises y SharePoint Online, haciendo transparente al usuario la gestión de contenidos.

2.3. Modelos de implementación

En esta sección, se presentarán las tres formas de implementar SharePoint, su versión On-premises, su versión Online y una versión híbrida entre ambas. Si bien es cierto que existe una cuarta manera de implementar SharePoint, consistente en instalar un SharePoint On-premises en servidores Azure, se considerará como parte de la implementación On-premises y no como un modelo adicional.

2.3.1. SharePoint On-premises

Es la versión más tradicional de SharePoint. El cliente adquiere unos servidores propios, bien de forma física o en la nube, junto con las licencias de Windows Server, SQL Server, SharePoint... necesarias para la instalación. Es el propio cliente quien se hace responsable de la instalación y configuración de su granja SharePoint. Este método es el tradicional y más habitual hasta la fecha.

Microsoft pone a disposición de los clientes un servicio de evaluación del estado de la implementación, conocido como RAP, encargado de validar el cumplimiento de las buenas prácticas dictadas por la compañía, en la granja. El no cumplimiento de algunas de estas normas, puede llegar a anular la garantía de Microsoft sobre el producto, eximiendo a la misma de toda responsabilidad. Los clientes podrán ejecutar el servicio tantas veces como lo consideren oportuno.

El mantenimiento y la administración de los servidores es también responsabilidad del cliente. Uno de los aspectos más relevantes a tener en cuenta, es la actualización de los servidores con los parches oficiales, pues Microsoft no garantiza el funcionamiento de las soluciones a medida tras la aplicación de los mismos, así como tampoco garantiza la persistencia de los cambios en las configuraciones estándar que el cliente haya podido realizar. Estos dos detalles, unidos a la excesiva complejidad de administración de la herramienta, han sido las principales dificultades que ha experimentado el producto a la hora de encontrar profesionales de IT capacitados.

La gran ventaja de este modelo es el control total de los servidores. Al depender directamente de cliente, los servidores son mucho más personalizables, pudiendo crear todo tipo de configuraciones y soluciones a medida que se adapten a las necesidades del cliente, incluso pudiendo fácilmente combinar SharePoint con otros servicios externos.

2.3.2. SharePoint Online

La dificultad a la hora de encontrar profesionales cualificados en administración de granjas SharePoint, unido a la creciente demanda de servicios en la nube, fomentó la aparición de SharePoint Online.

Es la versión más innovadora de SharePoint, incluida dentro del paquete de Office 365. Consiste en la combinación entre SharePoint y Azure para proporcionar al cliente granjas de SharePoint preinstaladas y con unos recursos a medida que pueden aumentar o disminuir bajo demanda.

En este caso, tanto la instalación como la configuración, el mantenimiento y gran parte de la administración son responsabilidad de Microsoft, a cambio de una cantidad mensual en función de los servicios contratados.

La principal desventaja de este modelo es su bajo grado de personalización. Los servidores son propiedad de Microsoft y, por tanto, prácticamente es imposible realizar ningún gran cambio en su configuración. La creación de soluciones a medida, se limita únicamente al uso de las APIs de cliente, mucho más limitadas que las de servidor, que no siempre permiten hacer todo lo que un SharePoint On-Premises sí permite, o que dificultan mucho tareas que en servidor se consideran inmediatas. Este hecho hace que únicamente las organizaciones que se adapten al modelo de negocio estándar de SharePoint se puedan permitir utilizarlo en su totalidad.

2.3.3. Implementaciones híbridas

La gran distancia que supone el salto desde SharePoint On-Premises a SharePoint Online, hizo que muchas de las empresas rechazaran la posibilidad de migrar sus contenidos a la nube, bien por inversiones en infraestructura / desarrollo, bien por requisitos no materializables. Esto hizo que, a partir de la versión 2016, SharePoint permitiese la creación de entornos híbridos, como medio de transición entre On-Premises y Online.

El principal objetivo de esta medida es el acercamiento a la nube de clientes tradicionales, migrando los contenidos no dependientes de una implementación On-Premises, y manteniendo en ella lo estrictamente necesario. La implementación híbrida es capaz de funcionar como una única granja de forma transparente al usuario, por lo que está siendo la alternativa más viable en la actualidad.

2.4. Arquitectura

En esta sección se tratará de dar una visión genérica de los elementos que forman parte de una granja de SharePoint y de su estructuración en los distintos servidores.

A nivel arquitectónico, una granja es el conjunto de todos los servidores físicos necesarios para la implementación de un entorno SharePoint. Las granjas pueden estar compuestas:

- Un único servidor que proporcione todos los servicios. Este tipo de granjas se conoce como standalone y se suelen utilizar para entornos de desarrollo o pruebas.
- Varios servidores entre los que se reparten la totalidad de servicios. Este tipo de granjas son las utilizadas en entornos productivos.

2.4.1. Roles de servidor y MinRole

Los servicios dentro de los servidores, se agrupan formando roles de servidor, con unas aplicaciones de servicio, servicios y componenetes específicos. En un servidor pueden coexistir varios roles. Existen cinco roles:

- **Front-end:** Servidor encargado de atender las peticiones de los usuarios. Se optimizará para proporcionar una baja latencia y un bajo tiempo de respuesta en cuando a solicitudes de usuario.
- **Aplicación:** Servidor encargado de atender las solicitudes back-end. Se optimizará para tener un alto rendimiento en sus procesos.
- **Caché distribuída:** Servidor encargado de la coordinación de diferentes cachés entre nodos de la granja.
- **Búsqueda:** Servidor dedicado únicamente a tareas relacionadas con el servicio de búsqueda.
- **Personalizado:** Servidor con una configuración totalmente personalizable. Se utiliza en casos en los que los anteriores no se adapten correctamente. Los administradores tendrán control total sobre este servidor.

Role	Requerido en granjas de contenido	Requerido en granjas de servicios compartidos	Requerido en granjas de búsqueda
Front-End	Si	No	No
Aplicación	Si	Si	No
Caché distribuída	Si	Si	No
Búsqueda	Si, si hospeda la búsqueda	Si, si hospeda la búsqueda	Si
Personalizado	Opcional	Opcional	Opcional

Cuadro 2.1: Topologías recomendadas

A partir de SharePoint 2016, las topologías se definen utilizando el concepto de MinRole, basado en dividir la granja en servidores con un único rol dentro de los cuatro definidos anteriormente, sin contar con el personalizado. Las ventajas de esta topología son:

- **Simplificación en la implementación:** Los administradores de únicamente tendrán que activar los servicios correspondientes a cada nodo y asignarle un rol dentro de la granja y SharePoint se encargará de configurarlo.
- **Rendimiento más optimizado:** Los servicios de SharePoint están optimizados debido a la colocación estratégica de cada uno de ellos dentro de la granja.
- **Planificación de la escalabilidad más sencilla:** Es muy sencillo añadir un servidor nuevo a la granja, únicamente configurándolo como un rol nuevo.

Servidor	Rol
Frontal 1	Front-End
Frontal 2	Front-End
Aplicaciones 1	Aplicación
Aplicaciones 2	Aplicación
Caché 1	Caché distribuida
Búsqueda 1	Búsqueda

Cuadro 2.2: Topología mínima MinRole

Esta visión de Microsoft es muy poco flexible, ya que únicamente tiene en cuenta los servidores de cara a SharePoint y no a la totalidad del sistema ni sus particularidades. Entre las principales desventajas:

- La implementación no se adapta a las necesidades reales del sistema, sino a las necesidades teóricas de SharePoint.

- Es necesario contar con un mínimo de siete servidores distribuidos tal y como se muestra en la tabla.
- No siempre es necesario dedicar un único servidor a cada rol.
- La escalabilidad es sencilla, pero no siempre se adapta a las necesidades. En caso de necesitar más recursos, es necesario añadir más servidores, no pudiendo aprovechar los recursos libres de otros servidores.

2.5. Requisitos mínimos

SharePoint es un producto con una alta exigencia en cuanto a recursos hardware, pues está basado en transacciones SQL y su consumo de CPU y RAM es bastante elevado.

La configuración de recursos óptima para los servidores de una granja es un aspecto muy complejo y ,habitualmente, requiere múltiples cambios y ajustes respecto a la planificación inicial. Para facilitar esta tarea, Microsoft proporciona una serie de recomendaciones en función de los recursos disponibles, fijando unos requisitos mínimos para que el sistema pueda funcionar correctamente.

Los requisitos hardware mínimos para una instalación de una granja de un único servidor son los siguientes:

Componente	Valor mínimo recomendado
Procesador	64 bits, 4 núcleos
Memoria RAM	16-24 GB dependiendo de los servicios configurados
Disco duro	80 GB libres para la instalación 80 GB libres como almacenamiento secundario

Cuadro 2.3: Requisitos hardware mínimos standalone

Los requisitos hardware mínimos para cada servidor en una instalación de una granja SharePoint son los siguientes:

Componente	Valor mínimo recomendado
Procesador	64 bits, 4 núcleos
Memoria RAM	16-24 GB dependiendo de los servicios configurados
Disco duro	80 GB libres para la instalación 80 GB libres como almacenamiento secundario

Cuadro 2.4: Requisitos mínimos granja

2.6. Modelo de objetos

En sus orígenes, SharePoint comenzó utilizando una API servidor en Visual Basic, con una funcionalidad limitada. La evolución del producto ha dado origen a que, en la actualidad, coexistan APIs servidor en C# y PowerShell, combinadas con APIs cliente en C#, PowerShell y tecnologías Javascript. Cada una de ellas se utiliza en un contexto y de una forma diferente, pero todas tienen algo en común: todas utilizan el modelo de objetos de SharePoint.

2.6.1. Jerarquía de sitios

A nivel lógico, la información en SharePoint se organiza de forma jerárquica, siguiendo un modelo de objetos propio de la plataforma. Los objetos más representativos son los siguientes:

- **Granja (Farm):** Representa la totalidad de la granja SharePoint. Engloba todos los servidores, servicios y soluciones instaladas, lo que lo sitúa en lo más alto de la jerarquía. Las funcionalidades asociadas a este objeto son, fundamentalmente, aquellas que tienen que ver con tareas de configuración. En la versión online, recibe el nombre de Tenant.
- **Aplicación web (Web application):** Representa cada aplicación web independiente dentro del entorno SharePoint. Una aplicación web es un conjunto de sitios web agrupados en base a criterios de funcionalidad, administración o negocio. Cada aplicación web está relacionada con una o varias URLs y uno o varios proveedores de autenticación.
- **Colección de sitios (Site collection):** Representa un contenedor de sitios web, agrupados por criterios de funcionalidad, administración o negocio. Su funcionalidad está relacionada con la configuración común a todos los sitios web que forman parte de ellas.
- **Sitio (Site):** Representa un sitio web, formado por páginas y componentes SharePoint. El objetivo fundamental de los sitios web, es proporcionar una partición del contenido total del sistema que, debido a criterios de funcionalidad, seguridad, estructura o negocio, funcionen como una unidad con cierto grado de independencia.

2.6.2. Columnas y tipos de contenido

En esta sección se tratará de exponer cómo se estructura la información dentro de SharePoint, haciendo una analogía con una visión muy genérica de la programación orientada a objetos que es, en definitiva, la base de su estructura. Se insiste en recalcar que la comparativa será muy superficial y se basará en los siguientes conceptos:

- Las entidades se representan mediante clases que se instancian creando objetos.
- La definición de una clase incluye la definición de atributos y métodos.
- Los atributos y métodos pueden ser públicos o privados.
- Las clases pueden heredar de otras clases, especificando estas.
- Todas las clases heredan de una clase superior, habitualmente representada como Object.
- Las clases pueden redescibir ciertos métodos de su clase padre.
- Los objetos pueden agruparse en contenedores de objetos de la misma clase, como arrays o listas.

Estos conceptos son suficientes para proporcionar una visión genérica del producto en las próximas secciones.

Las **columnas** representan los atributos de una clase, definiendo un nombre y un tipo de datos esperado. Cabe destacar que en SharePoint, no se permite que una columna almacene más de un valor, por lo que no podría crearse una columna cuyo tipo de datos esperado fuese, por ejemplo, un array.

Los tipos de columna más habituales en SharePoint son:

- **Línea de texto:** Cadena de caracteres corta.
- **Múltiples líneas:** Cadena de caracteres larga.
- **Si/no:** Campo boolean.
- **Número:** Número de cualquier tipo.
- **Moneda:** Valor monetario.
- **Persona o grupo:** Campo de usuario.
- **Imagen o link:** Enlace o imagen (las imágenes se almacenan como enlaces al lugar donde se encuentran).

Los **tipos de contenido** representan las clases, definidos como una agrupación de columnas. Al igual que las clases, los tipos de contenido siempre heredan, al menos, de System (equivalente a Object) y pueden heredar de cualquier otro tipo de contenido.

SharePoint permite a los tipos de contenido sobrescribir métodos de su clase padre, fundamentalmente eventos, pero no permite añadir nuevos métodos. Es por ello que, además de System, SharePoint pone a disposición del usuario tipos de contenido básicos, como el documento o el elemento de lista, que disponen de las operaciones necesarias para según qué necesidades.

La recomendación de Microsoft dicta que se debe tratar de maximizar la reutilización a la hora de crear columnas y tipos de contenido, es decir, tratar de crear el mínimo conjunto de columnas posibles que permita representar los datos con el nivel de detalle necesario y utilizar la herencia entre tipos de contenido en lugar de la creación de tipos de contenido independientes con múltiples columnas comunes.

2.6.3. Listas y bibliotecas

En el contexto de SharePoint, los objetos o instancias de los tipos de contenido, son persistentes y deben almacenarse dentro de contenedores llamados listas. Una **lista** no es otra cosa que agrupación de objetos con los mismos atributos y cuyo contenido se almacena en la memoria física. Las instancias de dichos objetos, se denominan elementos de lista.

La recomendación de Microsoft dicta que una lista debe almacenar elementos de un único tipo de contenido. En la práctica, una lista puede estar compuesta de varios tipos de contenido o incluso directamente de columnas pero, en este proyecto, esa alternativa no será contemplada.

Una **biblioteca** en SharePoint, es un tipo de lista concreto que almacena elementos cuyo tipo de contenido necesariamente hereda de Documento. La importancia que los documentos tienen en SharePoint, como gestor documental, hace que se hable de listas y bibliotecas como dos conceptos distintos, cuando realmente no lo son.

Microsoft recomienda almacenar varios tipos de contenido distintos en la misma biblioteca, únicamente cuando estos sean de distinta naturaleza y deban tratarse como si no lo fueran. Un ejemplo claro y que se utilizará en este proyecto será el de almacenar documentos y conjuntos de documentos en la misma biblioteca, siendo tipos de contenido distintos. El conjunto de documentos no es otra cosa que un documento, que contiene varios documentos en su interior, a pesar de funcionar como uno solo. Un ejemplo típico de su uso, es el de un expediente que necesita actuar como una única entidad, pero almacenar múltiples documentos en su interior.

2.6.4. Permisos

Los permisos en SharePoint se asignan a usuarios o grupos. El concepto de usuario, se asocia a una única entidad, mientras que el concepto de grupos se divide entre grupos SharePoint, creados y gestionados desde la propia plataforma, y grupos de dominio, creados y gestionados externamente desde Active Directory. La conceptualización de estos elementos en el modelo de objetos es un tanto confusa:

- **Grupo (Group):** Representa un grupo estrictamente de SharePoint.
- **Usuario (User):** Representa tanto a un usuario individual como a un grupo de dominio. Aunque parezca confuso, SharePoint trata los grupos de dominio como si fuesen un usuario individual. SharePoint es capaz de conocer a qué grupos de dominio pertenece cada usuario, pero no es capaz de conocer los miembros de un determinado grupo de dominio. Para distinguir si un usuario corresponde a un usuario real o es un grupo de dominio, se utiliza una propiedad boolean del objeto (típicamente llamada IsDomainGroup).

2.7. Desarrollo de soluciones

El desarrollo de soluciones a medida sobre SharePoint, es una alternativa que sirve para cubrir aquellas necesidades de los usuarios, a las que el producto no se adapta. Consiste en la creación de funcionalidades personalizadas que se pueden utilizar en diferentes ámbitos dentro de la granja.

Los desarrollos se estructuran en paquetes (soluciones WSP) que, posteriormente, son instalados en toda la granja. El contenido típico de una solución puede incluir clases, páginas de aplicación, ficheros javascript, ficheros de recursos, hojas de estilos... Existen dos elementos que deben existir necesariamente en una solución SharePoint:

- **Paquete (Package):** Es un manifiesto donde se declara cada uno de los elementos que incluye la solución, así como todas aquellas referencias y requisitos que el paquete necesita para su correcto funcionamiento. Deberá existir un único manifiesto package en cada solución.
- **Característica (Feature):** Es una agrupación de una partición de los elementos del paquete, correspondiente a una funcionalidad específica. Dentro de cada solución pueden existir múltiples características y estas pueden ser tan específicas como se requiera.

Las características SharePoint, tienen un **ámbito de activación**, que indica el nivel al que la característica será habilitada e implicará ciertas modificaciones asociadas. Los 4 posibles valores son:

- **Farm:** La característica se habilita para toda la granja. Son muy pocas las características que se habilitan a este nivel y suelen estar relacionadas con parámetros de configuración. La creación de este tipo de características sólo está justificada en el caso de que deban ser habilitadas y deshabilitadas con una cierta frecuencia, y no como medio para realizar configuraciones de infraestructura. En la actualidad, este tipo de características tienden a ser sustituidas por scripts PowerShell.
- **Web application:** La característica se habilita para una aplicación web. Son pocas las características que se habilitan a este nivel y suelen estar relacionadas con parámetros de configuración de los portales o despliegue de utilidades comunes a toda la aplicación como el log. De forma adicional, son desplegadas a este nivel aquellas características que deban estar presentes en todos y cada uno de los sitios del portal, sin excepción. En la actualidad, las características que desplegaban configuración a este nivel, tienden a ser sustituidas por scripts PowerShell.
- **Site:** La característica se habilita para toda la colección de sitios y todos los sitios que forman parte de ella. Es la opción más común a la hora de desarrollar soluciones. En este ámbito, se despliegan las funcionalidades comunes a todos los sitios de la colección.
- **Web:** La característica se habilita para un único sitio o subsitio. En este ámbito, se despliegan características que, por definición, sólo deben estar habilitadas en determinados sitios y no en toda la colección.

Las tareas más habituales en proyectos de desarrollo SharePoint son las siguientes:

- **Definición de listas:** Definición y creación de las columnas de sitio, tipos de contenido, listas y bibliotecas. Tradicionalmente se generaba como una solución adicional, pero su dudosa reutilización hizo que Microsoft apostase por PowerShell como medio para ello.
- **Creación de WebPart:** Un webpart es un elemento web reutilizable, que puede colocarse dentro de una página.
- **Creación de Display Templates:** Las display templates son plantillas de resultados de búsqueda que permiten presentar al usuario la información de una consulta específica o de una caja de búsqueda, de forma personalizada. Se utilizan no sólo en buscadores, sino como medio para presentar todo tipo de contenidos, utilizando una lógica de consulta estándar de SharePoint, con webparts de caja.
- **Creación de TimerJob:** Un timerjob es un procedimiento que se ejecuta siguiendo una planificación específica y que realiza una determinada función en el sistema.

Capítulo 3

Escenario y alternativas

El mundo de la informática está en constante crecimiento. Las tecnologías que se utilizan en los distintos ámbitos evolucionan en función de las necesidades de los usuarios, adaptándose a ellas. Los usuarios cada día son más exigentes, demandando la obtención de servicios que, poco a poco, se asumen como necesarios cuando poco tiempo atrás no eran sino simples ideas innovadoras que en un futuro podrían triunfar.

El objetivo de este trabajo de fin de grado es analizar cuáles son esas necesidades en el ámbito universitario, tanto desde el punto de vista del alumno, como del resto de actores que forman parte del mismo, y proporcionar una alternativa viable de crecimiento a futuro, que permita satisfacer todas ellas con un coste asumible.

En este capítulo, se analizará el sistema existente, sus ventajas e inconvenientes, y se analizará la alternativa que este trabajo propone, Sharepoint, tratando de encontrar las ventajas que esta última ofrece sobre la actual.

3.1. Descripción del escenario

La universidad es un entorno colaborativo, donde la comunicación y la compartición de información entre profesores y alumnos son aspectos fundamentales. Además de ello, las universidades son centros de referencia que, mediante la investigación, trabajan día a día para conocer más sobre el mundo que nos rodea y, si es posible, hacerlo un lugar mejor.

En la actualidad, resulta inevitable hablar de comunicación, compartición, colaboración y acceso a la información, sin hablar de informática. Desde hace muchos años, con la aparición de los primeros sistemas informáticos, las universidades han ido evolucionando, proporcionando facilidades a sus miembros para el desarrollo de su actividad. Desde la aparición del correo electrónico, hasta el concepto actual de intranet, que a día de hoy, sigue evolucionando de forma vertiginosa.

La gran mayoría de universidades, sino todas, cuentan con un sistema de correo electrónico y una intranet donde profesores y alumnos pueden compartir material para las asignaturas pero, ¿es esto suficiente? La respuesta es no, nunca es suficiente, siempre se puede mejorar. Los usuarios actuales hacen uso del sistema que la universidad proporciona, y añaden a él, herramientas de terceros que completan aquellas carencias que dicho sistema posee.

Tratemos de analizar las necesidades típicas de profesores y alumnos en la actualidad, en la Universidad de Valladolid. Para ello, tratemos de analizar aquellos comunes a todas las asignaturas, tratando de exponer las facilidades que el sistema existente proporciona, y cómo los alumnos reaccionan ante ellas.

Al inicio del curso académico, los alumnos de una determinada asignatura, son añadidos como miembros de ella en el sistema informático, obteniendo el acceso a los contenidos de la misma durante ese dicho curso. Típicamente, en asignaturas con un número determinado de alumnos, éstos son divididos en grupos de teoría y en grupos de prácticas. Esta tarea suele realizarse siguiendo criterios como el orden alfabético de sus apellidos... Los profesores comunican de forma presencial / correo electrónico, los miembros de dichos grupos pero, de cara al sistema, no existe tal diferenciación, ni llega a existir una consecuencia palpable como, por ejemplo, mostrar ciertos contenidos, avisos.. a los usuarios de un subgrupo y no mostrarlos a todos.

En las prácticas grupales, habitualmente se definen grupos de trabajo, típicamente de 2 a 4 personas, que trabajarán de forma colectiva para obtener una calificación conjunta en los distintos elementos evaluables. El concepto de pertenencia a un grupo, suele estar definido en un documento enviado a los alumnos por el profesor, pero en el sistema no figura ningún tipo de relación de pertenencia a grupo por parte de los usuarios.

Los dos párrafos anteriores desembocan irremediabilmente, en un exceso de in-

formación inútil para los alumnos, pues toda la información exclusiva de un grupo (seminarios, prácticas, documentos. . .), es visible para todos al no existir en el sistema una forma de diferenciarlos.

Una vez asignados los alumnos de un grupo, es responsabilidad de los mismos comunicarse entre sí, para realizar dichas prácticas. Esto desemboca en una solicitud masiva de correos electrónicos personales, pues el correo que proporciona la universidad no es muy cómodo, y números de teléfono personales para realizar llamadas de voz y, sobre todo, poder comunicarse a través de aplicaciones de mensajería instantánea, tales como Whatsapp o Telegram, pues no existe ninguna alternativa viable a ello dentro del conjunto de aplicaciones que la universidad ofrece.

A la hora de poder trabajar en equipo, los integrantes del grupo deben decidir cómo poder trabajar en grupo. Para ello, se utilizan herramientas típicas tales como Google Drive / Office 365 para editar documentos de forma simultánea, Google Drive / Microsoft Onedrive / Dropbox para almacenar ficheros en la nube y gestionar el versionado de los mismos, Skype / Hangouts / Discord para llamadas grupales y compartición de escritorio, Trello para gestión de tareas o proyectos, Whatsapp / Telegram para comunicarse.

Es habitual que, fuera del horario de clase, los alumnos necesiten comunicarse con su profesor. Para ello, el profesor pone a disposición del alumno unos horarios de tutoría, donde poder atender de forma presencial, en despacho, las dudas que éste pueda tener sobre la asignatura. Adicional a ello, los alumnos disponen de una dirección de correo electrónico del profesor, donde podrán comunicarse con este y esperar respuesta. La realidad que se esconde detrás de esto, es que la mayoría de las horas destinadas a las tutorías no son aprovechadas por los alumnos por incomodidad, y los correos electrónicos con dudas puntuales, no siempre son tan eficaces, y muchas veces retrasan demasiado la respuesta frente a un servicio de mensajería instantánea.

Durante el transcurso de las clases, el profesor utiliza, típicamente, unas diapositivas que sirven como guía, sobre las que aporta unas explicaciones que deben ser tomadas como apuntes por los alumnos. Algunos profesores utilizan herramientas de terceros para tomar notas sobre las diapositivas y, posteriormente comparten estas notas con los alumnos adjuntando otro archivo distinto al original. Estas prácticas propician una pérdida de información para el alumno si, por algún motivo, no le resulta posible acudir a clase, y dificulta la compartición de dicha información al no disponer de una versión digital de los documentos con notas.

La intranet de la universidad es muy poco personalizable, obligando al usuario a adaptarse a una forma de trabajar definida por el sistema, sobre la que no tiene ninguna capacidad de decisión. Esto es un concepto cada vez más importante, los sistemas deben proporcionar al usuario una cierta flexibilidad de organización que actualmente no existe.

3.2. Análisis del sistema actual

La gestión de la docencia en la Universidad de Valladolid se realiza a través de una plataforma, Campus Virtual, basada en Moodle y para la que la universidad ha desarrollado sus propias personalizaciones.

Es una herramienta de Learning Content Management (LCMS), de distribución libre y escrita en PHP, cuyo objetivo fundamental es proporcionar a los educadores una forma de crear comunidades de aprendizaje online.

Hasta julio de 2008, la base de usuarios registrados incluye más de 21 millones, distribuidos en 46.000 lugares, en los 91 idiomas a los que la herramienta está traducida.

Ventajas de Moodle como software

- **Es un software libre.** Además de ser gratuito, esto permite a los desarrolladores, implementar sus propias soluciones personalizadas, adaptándose a los problemas particulares de la comunidad educativa para la que trabajan.
- El hecho de ser un software libre, propicia la **existencia de una comunidad abierta**, que permite intercambiar ideas, información... De esta comunidad, los desarrolladores pueden colaborar para mejorar el software.
- **Es escalable**, permitiendo su implementación tanto en organizaciones pequeñas como grandes.
- Implementa mecanismos de **seguridad**.

Ventajas de Moodle para la docencia

- **Simplicidad** en su uso y **facilidad de acceso y administración**.
- Permite la **gestión de perfiles de usuario**.
- **Módulo de tareas entregables**, que permite la entrega, revisión y calificación de las mismas.
- **Módulo de consulta**, que facilita la creación de votaciones para un tema determinado.
- **Módulo foro**, donde poder abrir discusiones entre los miembros de un determinado grupo.
- **Módulo diario**, donde poder compartir información privada entre alumno y profesor.
- **Módulo cuestionario**, donde poder almacenar una base de datos de preguntas y posteriormente realizar exámenes que serán calificados automáticamente.

- **Módulo recurso**, que permite la presentación de contenido digital relativo a la asignatura por parte del profesor, hacia sus alumnos.
- **Módulo wiki**, que permite el trabajo conjunto de varios alumnos sobre el mismo documento.
- **Módulo taller**, que permite la creación de tareas para ser evaluadas entre pares.

3.3. Análisis del sistema propuesto

La solución propuesta en este trabajo consiste en utilizar SharePoint para tratar de crear un aula virtual. En la actualidad, SharePoint es un producto que se distribuye junto a Office 365, por lo que resulta mucho más interesante orientar este punto hacia el valor que aporta Office 365 en conjunto, como plataforma educativa.

Office 365 es la versión en la nube de la suite de productividad de Microsoft. Sus servicios se adquieren en forma de suscripción, con distintas modalidades de pago (mensual, anual. . .) y proporciona el acceso, tanto a las aplicaciones de Office On-premise tradicionales, que actualmente son utilizadas por 1.200 millones de usuarios en el mundo, como a aplicaciones más modernas orientadas a organizaciones.

Según declaraciones de Satya Nadella, CEO de Microsoft, Office 365 es uno de los productos con mayor potencial de crecimiento en la historia de la compañía. Esta afirmación se sustenta en las impresionantes cifras de crecimiento. En 2017, Office 365 contaba con 25 millones de usuarios activos mensuales en sus versiones comerciales, mientras que tan sólo un año después, en 2018, estos usuarios superan la cifra de 100 millones.

Ventajas de Office 365 como software

- Dispone de una **suscripción mensual gratuita** para los usuarios de organizaciones del ámbito educativo, como universidades, que incluye todas las aplicaciones necesarias.
- **Es software propietario**. Esto implica el respaldo de una gran empresa, Microsoft, que pone a disposición de los usuarios todo tipo de facilidades de uso de sus productos (documentación, soporte técnico. . .) y a los desarrolladores una amplia gama de documentación técnica oficial, catálogo de buenas prácticas. . . que facilita mucho la labor de desarrollo sobre los mismos.
- Dispone de **una de las comunidades más grandes del mundo**, con miles de usuarios, desarrolladores. . .
- **Utiliza tecnologías propietarias** basadas en .NET, que están continuamente actualizándose para adaptarse a las necesidades específicas de sus productos.

Ventajas de Office 365 para la docencia

- Es un conjunto de aplicaciones que **integra gran parte de las necesidades** de los usuarios en el ámbito educativo.
- **Outlook y Exchange**, proporcionan al usuario los servicios de correo electrónico y calendario.
- **Word, Excel y PowerPoint**, permiten al usuario la creación de documentos, tanto locales como online, y permiten el acceso y modificación concurrente de los mismos, lo que permite, por ejemplo, la realización de una práctica de forma simultánea por todos los miembros del grupo.
- **Onenote** proporciona al usuario un bloc de notas online, donde tomar apuntes de las asignaturas, prácticas... tanto de forma individual como conjunta, inclusive con el profesor.
- **Sharepoint** proporciona al usuario el concepto de intranet al que está acostumbrado, permitiendo, entre otras muchas cosas, el almacenamiento, estructuración y compartición de información.
- **Onedrive** proporciona al usuario almacenamiento en la nube de todo tipo de archivos y compartición de los mismos. Realmente este Onedrive no es otra cosa que la vista, en modo explorador de archivos, del contenido almacenado en Sharepoint.
- **Skype Empresarial**, permite a los usuarios la comunicación directa mediante mensajería instantánea y videollamada. Entre sus funciones más destacadas está la de presentar escritorio, donde el usuario receptor puede ver en tiempo real la pantalla del emisor, e incluso tomar el control de la misma siempre que el emisor lo permita.
- **Teams** surge como la evolución natural de Skype para adaptarse a las nuevas necesidades. Además de permitir todo lo que ya permite Skype, permite la creación de grupos fijos, con unos integrantes. Estos grupos, a su vez, pueden contener subgrupos identificados con un nombre y un propósito más específico. Cada grupo / subgrupo tiene un chat global y, opcionalmente, podrá disponer de su propio Onedrive, Onenote, Trello, Planner... Integrado en la misma aplicación de Teams. Esta aplicación es relativamente joven, saliendo al mercado en 2017 y es una de las herramientas en las que microsoft trabaja con más insistencia, añadiendo constantemente funcionalidades nuevas.
- **Sway** permite impartir clases online en tiempo real
- **Forms** permite la creación de formularios.
- **Flow** permite la automatización de tareas en flujos de trabajo.
- **PowerApps** permite la creación de aplicaciones modulares, que se distribuyen, con o sin coste, en la store de Office.

- **School Data Sync**, que permite la automatización del proceso de creación de grupos para asignaturas. Simplificando mucho sus funcionalidades, crea los grupos de Teams, el espacio Onedrive, el bloc en Onenote, el sitio Sharepoint. . .

3.4. Comparativa de sistemas

Desde hace años, la Universidad de Valladolid mantiene una suscripción anual de Microsoft, que proporciona a sus profesores y alumnos el acceso a una gran variedad de licencias de herramientas de desarrollo, sistemas operativos y otros productos de la compañía de forma totalmente gratuita, a través de su portal Microsoft Imagine, también conocido como DreamSpark. Junto con lo anteriormente mencionado, esta licencia de campus, proporciona acceso a suscripciones totalmente gratuitas de Office 365 para profesores y alumnos. El uso de esta suscripción, pondría a disposición de alumnos y profesores un abanico de posibilidades mucho más amplio al que actualmente están acostumbrados, solucionando gran parte de los problemas planteados en puntos anteriores de esta sección.

El gran debate que surge necesariamente antes de implantar una metodología de trabajo basada en herramientas de Microsoft es el siguiente: ¿es realmente una buena práctica que la universidad inste a los alumnos a utilizar los productos de una determinada compañía en lugar de utilizar otras alternativas orientadas al software libre? Esta cuestión es un debate abierto, con partidarios y detractores, que queda fuera del alcance de este trabajo. Es por tanto que se tratará de justificar la decisión, basándose únicamente en las ventajas e inconvenientes que la implantación del mismo conllevarían para la universidad.

Combinando Outlook y los servicios de Exchange, profesores y alumnos acceso a cuentas de correo electrónico y un calendario en una única aplicación. Una buena gestión del calendario permitiría a los usuarios estar al tanto de todos los eventos asociados a las asignaturas, favoreciendo la organización. Destacar que un buen uso del planificador temporal incluido en Outlook, permitiría saber, bajo calendario, las franjas horarias disponibles de un usuario para, por ejemplo, proponer un horario para una tutoría.

Word, Excel y Powerpoint permitirían el trabajo conjunto de los miembros de un determinado grupo sobre un mismo documento, hoja de cálculo, presentación.

Onenote permitiría a los profesores, además de crear blocs de notas personales, compartir con los alumnos un bloc de notas global para la asignatura, donde poder incluir las distintas anotaciones que se consideren oportunas para el desarrollo de la asignatura. Además, permitiría a los alumnos unificar en una aplicación el concepto de la toma de apuntes, facilitando la compartición de los mismos con sus compañeros, pudiendo crear blocs conjuntos. Además, los alumnos pueden crear blocs de notas

que sirvan como cuaderno de bitácora para el desarrollo de las prácticas, a los que el propio profesor podrá tener acceso si lo considera oportuno.

Onedrive proporciona a cada usuario, de forma gratuita, un espacio de almacenamiento en la nube de 1TB, donde podrá almacenar todos los archivos que considere oportunos sin, presumiblemente, ninguna preocupación por el espacio disponible.

Teams es una de las aplicaciones más jóvenes y a la vez más potentes en la suite de Office 365. Surge como una evolución de Skype, bajo la motivación de ofrecer una experiencia de conversaciones modernas, con charla persistente y enfocada en mantener a todos los participantes involucrados activamente. Permite la mensajería instantánea tanto individual como colectiva, y la creación de grupos permanentes, con un chat propio donde poder intercambiar mensajes, con un sistema de enfatizaciones basado en menciones y etiquetado. Cada uno de los grupos, podrá tener su propio espacio de almacenamiento en Onedrive, bloc de notas Onenote, tablero de Trello / Planner... Su uso en la universidad, permitiría la creación de un grupo para cada asignatura, y la creación de subgrupos para gestionar los grupos de laboratorio, los grupos de prácticas... sustituyendo el uso de aplicaciones de mensajería instantánea / llamadas de voz externas. Esto proporciona un medio para que los alumnos se comuniquen entre sí, y fomenta mucho la comunicación alumno / profesor, pudiendo dar lugar a tutorías interactivas, incluso grupales, de forma remota.

School Data Sync permitiría, mediante una buen conector con la aplicación de matrículas de la universidad, la automatización del proceso de creación de los grupos en el sistema. Con una buena configuración sería posible crear cada curso, de forma automática, los grupos estándar de Teams para cada asignatura, grupos de Sharepoint...

Sway facilitaría la impartición de cursos a distancia, algo que cada vez es más común entre las universidades de referencia.

Flow permitiría la automatización de tareas para procedimientos internos, ayudando a mejorar los procesos burocráticos.

PowerApps permitiría adquirir, desde la store de Office, aplicaciones gratuitas o de pago, que pudiesen completar las funcionalidades existentes.

Todas las funcionalidades descritas en este punto, serían un añadido al sistema existente pero no podemos dejar a un lado el hecho de que el concepto de intranet que proporciona Moodle, debe ser cubierto. Es en este contexto donde aparece la figura de la herramienta más potente de la suite, Sharepoint.

Sharepoint proporciona a los usuarios el concepto de intranet, pudiendo proporcionar todas las funcionalidades de Moodle, y hacerlas compatibles con el resto de funcionalidades. El resto de aplicaciones mencionadas, proporcionan una funcionalidad estandarizada, que no necesita de desarrollos a medida, pero para conseguir que

SharePoint cubra las funcionalidades que actualmente abarca Moodle, será necesaria la creación y estructuración de un entorno colaborativo con soluciones a medida. Éste es el verdadero objetivo de este trabajo de fin de grado, mostrar cómo funciona Sharepoint y cómo se adaptaría para proporcionar una intranet a la Universidad de Valladolid.

Capítulo 4

Planteamiento de la solución

Como en todo sistema informático, y especialmente cuando se trata de sistemas encargados de gestionar grandes volúmenes de información, el plantamiento inicial condiciona la totalidad del sistema, pues establece los cimientos sobre los que posteriormente se edificará.

En este capítulo se analizará el planteamiento de la solución, estudiando cómo adaptar las necesidades del sistema, con los recursos que SharePoint ofrece.

4.1. Alcance de la propuesta

Las funcionalidades de un sistema de este tipo son demasiado extensas. El alcance de este proyecto únicamente abarcará la gestión de una asignatura, tratando de representar cómo podrían interactuar profesores y alumnos y mostrar el potencial de SharePoint en un entorno de este tipo. Las funcionalidades que se proporcionarán en este proyecto estarán fundamentalmente orientadas al alumno, desarrollándose funcionalidades relativas a los profesores siempre y cuando sean necesarias para el correcto funcionamiento de las anteriores.

A continuación, se presentará un listado de las funcionalidades a desarrollar para los alumnos:

Funcionalidad	Descripción	Prioridad
Enlaces asignatura	Mostrar una lista de enlaces a las asignaturas en las que el alumno está matriculado, para que éste pueda navegar entre ellas	1
Biblioteca de documentación	Crear una biblioteca que permita a los alumnos tener acceso a los materiales de la asignatura.	1
Entregas	Crear la estructura necesaria para que un alumno pueda tener acceso a las prácticas de la asignatura y realizar entregas.	1
Entregas	Crear un tablero de tareas, donde los miembros de una misma entrega puedan gestionar el estado de la misma.	3
Entregas	Crear un sistema de comentarios, para permitir que los usuarios de una misma entrega puedan compartir información.	4
Entregas	Crear una utilidad calendario, que permita mostrar aquellas tareas con una fecha de vencimiento cercana.	4
Calificaciones	Mostrar al usuario una lista con las calificaciones obtenidas en los distintos evaluables de la asignatura.	1
Listado de grupos	Mostrar al usuario la pertenencia a los distintos grupos de la asignatura.	1
Listado de grupos	Mostrar al usuario una lista con todos los grupos de la asignatura y sus miembros.	2
Noticias	Crear una lista de noticias, que permita mostrar los avisos / noticias relevantes a la misma.	3

Cuadro 4.1: Funcionalidades de los alumnos

Las funcionalidades a desarrollar, para los profesores, serán más limitadas y enfocadas a permitir el correcto flujo de trabajo del resto:

Funcionalidad	Descripción	Prioridad
Calificaciones	Calificar un evaluable para un alumno o grupo concreto.	2
Entregas	Crear una entrega para los miembros de la asignatura.	2
Noticias	Crear noticias para la asignatura.	3

Cuadro 4.2: Funcionalidades de los alumnos

4.2. Infraestructura y herramientas

Debido a la naturaleza del proyecto, no es posible contar con la infraestructura necesaria para la creación de una granja de múltiples servidores, por lo que se optará por la creación de una granja de servidor único, virtualizada mediante Hyper-V en una máquina local.

Para el desarrollo del proyecto, se utilizarán las licencias que la Universidad de Valladolid pone a disposición de los alumnos en la plataforma Microsoft Imagine. Dichas licencias proporcionan todas las herramientas necesarias para el proyecto, por lo que no será necesario realizar ninguna inversión en este aspecto.

El software utilizado para la realización del proyecto es el siguiente:

- **Windows Server 2012 R2:** Sistema operativo sobre el que se instalarán los componentes. Se ha escogido la versión 2012 en lugar de 2016 por razones de fiabilidad.
- **SQL Server 2016 Developer:** Versión para desarrolladores del software de bases de datos de Microsoft, donde se almacenarán todos los contenidos de SharePoint. Se ha escogido la versión 2016 en lugar de la 2017 por razones de fiabilidad.
- **SharePoint Server 2016:** Última versión On-Premises del producto.
- **Visual studio Enterprise 2015:** Entorno de desarrollo para soluciones SharePoint. Se ha escogido la versión 2015 en lugar de 2017 únicamente por razones de comodidad.
- **U2U CAML Query builder:** Herramienta de apoyo para la creación de consultas en lenguaje CAML, utilizadas para listas SharePoint.
- **Search Query Tool:** Herramienta de apoyo para la realización de consultas de búsqueda basadas en API REST.
- **Google Chrome:** Navegador web principal, utilizado para acceder a SharePoint.

- **Internet explorer:** Navegador web secundario, utilizado para realizar pruebas adicionales.

4.3. Estructura de la solución

El proyecto estará compuesto por soluciones a medida que incluyen webparts, scripts PowerShell para definiciones de columnas, tipos de contenido, listas, bibliotecas... y archivos HTML para la definición de las display templates.

En cuanto al código, se ha optado por generar un único proyecto, llamado Webparts, con una única feature que despliega todos los webparts en la colección de sitios. La justificación de esto se basa en la funcionalidad, ya que todos los webparts serán utilizados en el mismo contexto y, en cualquier sitio que se despliegue uno de los webparts, podría perfectamente utilizarse otro, por lo que no tiene sentido separarlos en características distintas. En futuros evolutivos sobre esta plataforma, sería conveniente dividir los webparts en diferentes características dentro del mismo paquete, agrupándolos a nivel funcional.

En cuanto a scripts, se proporcionarán junto con el código organizados de la siguiente manera:

- **Columnas de sitio:** Un único script que contiene la definición de todas las columnas de sitio existentes en la solución y cuyo nombre será SiteColumns.ps1
- **Tipos de contenido:** Un script por cada tipo de contenido, cuya nomenclatura será la siguiente: CType-*Nombre.ps1
- **Listas y bibliotecas:** Un script por cada lista o biblioteca, cuya nomenclatura será la siguiente: List-*Nombre.ps1 para listas, y Library-*Nombre.ps1 para bibliotecas.
- **Sitios:** Un script que sirva como plantilla para crear un sitio para una asignatura en base a cuatro parámetros de configuración: nombre, URL, URL del site collection y descripción. Su nombre será WebDefinition.ps1
- **Colecciones de sitios:** Un script que sirva como plantilla para crear una colección de sitios para un grado en base a tres parámetros de configuración: nombre, URL y descripción. Su nombre será SiteDefinition.ps1

Las display templates, se presentarán en una carpeta adicional de la solución, incluyendo únicamente un fichero HTML por cada una de ellas, con la siguiente nomenclatura: DisplayTemplate-Nombre.html.

4.4. Organización de la información

Para el funcionamiento del Campus Virtual, es necesario almacenar los datos de las distintas carreras, asignaturas... Establecer una correcta jerarquía, será uno de los puntos más clave a la hora de garantizar la eficiencia del sistema. En este aspecto, existen dos elementos fundamentales que deberán actuar como uno sólo, por un lado está la jerarquía propia del negocio, en este caso la Universidad, y la jerarquía de la plataforma, en este caso SharePoint.

4.4.1. Información a almacenar

En el siguiente diagrama, se muestran las entidades que se han identificado dentro del dominio del problema, y las relaciones entre ellas:

- **Campus virtual:** Representa la totalidad del Campus Virtual.
- **Grado:** Representa un grado con todas sus menciones.
- **Asignatura:** Representa una asignatura asociada a un grado.
- **Usuario:** Representa un usuario físico genérico del Campus Virtual.
- **Administrador:** Representa un usuario con permisos de administrador sobre el grado o alguna de sus asignaturas. Será el encargado de realizar labores de administración propias de un técnico, para las que se necesitan permisos mayores a los de un usuario normal.
- **Profesor:** Representa un profesor de la universidad. Será el encargado de gestionar la documentación de la asignatura, así como de crear los evaluables y sus calificaciones.
- **Alumno:** Representa un alumno de la universidad. Será el encargado de realizar los evaluables correspondientes a las asignaturas y obtendrá calificaciones por parte del profesor.
- **Grupo:** Representa una agrupación alumnos de la universidad, asociado a una asignatura.
- **Grupo teoría:** Representa un grupo de teoría asociado a una asignatura.
- **Grupo laboratorio:** Representa un grupo de laboratorio asociado a una asignatura. Se asociará con evaluables de tipo laboratorio.
- **Grupo prácticas:** Representa un grupo de trabajo asociado a una asignatura. Se asociará con evaluables de tipo práctica.
- **Evaluable:** Representa todo elemento calificable asociado a una asignatura.
- **Práctica:** Representa una entrega de una asignatura.

- **Examen:** Representa un examen asociado a una asignatura.
- **Laboratorio:** Representa una prueba de laboratorio asociada a una asignatura.
- **Seminario:** Representa un seminario evaluable asociado a una asignatura.
- **Otros:** Representa cualquier elemento distinto de los cuatro anteriores que sea susceptible de ser calificado.
- **Calificación:** Representa la calificación de un evaluable asociado a una asignatura.
- **Documento:** Representa cualquier documentación asociada a la asignatura.

Las relaciones entre las entidades previamente descritas son las siguientes:

- El campus virtual tendrá asociados 0 o más grados, ya que es posible que exista un campus sin grados en el momento de su creación. Un grado estará asociado únicamente a un campus virtual.
- El campus virtual estará asociado a 0 o más usuarios, ya que es posible que exista un grado sin usuarios en el momento de su creación. El/los usuario/s administrador/es del campus no se tienen en cuenta como parte de la entidad usuario, ya que no representan a ninguna persona física.
- Un grado estará asociado a 0 o más asignaturas, ya que es posible que exista un grado sin asignaturas en el momento de su creación. Una asignatura estará asociada a un único grado y, en caso de que sea impartida en más de un grado, se tratarán como asignaturas independientes.
- Un grado estará asociado a 1 o más usuarios, ya que es necesario que exista al menos un administrador del grado en el momento de su creación. Un usuario estará asociado a 0 o más grados, ya que los usuarios pueden existir en el campus sin necesidad de estar asociados a ningún grado.
- Los usuarios administrador, profesor y alumno, heredarán de usuario y tendrán permisos específicos para cada uno de sus roles.
- Una asignatura estará asociada a 1 (principal), o 2 (principal y secundario) administradores. y un administrador estará asociado a 0 o más asignaturas.
- Una asignatura estará asociada a 0 o más profesores o alumnos, ya que puede existir una asignatura sin profesores o alumnos en un momento determinado (fin de curso...). Un profesor o alumno estará asociado a 0 o más asignaturas, ya que puede que exista un profesor que no imparta docencia en ninguna asignatura o un alumno que no esté matriculado en ninguna asignatura (durante el proceso de matrícula...).

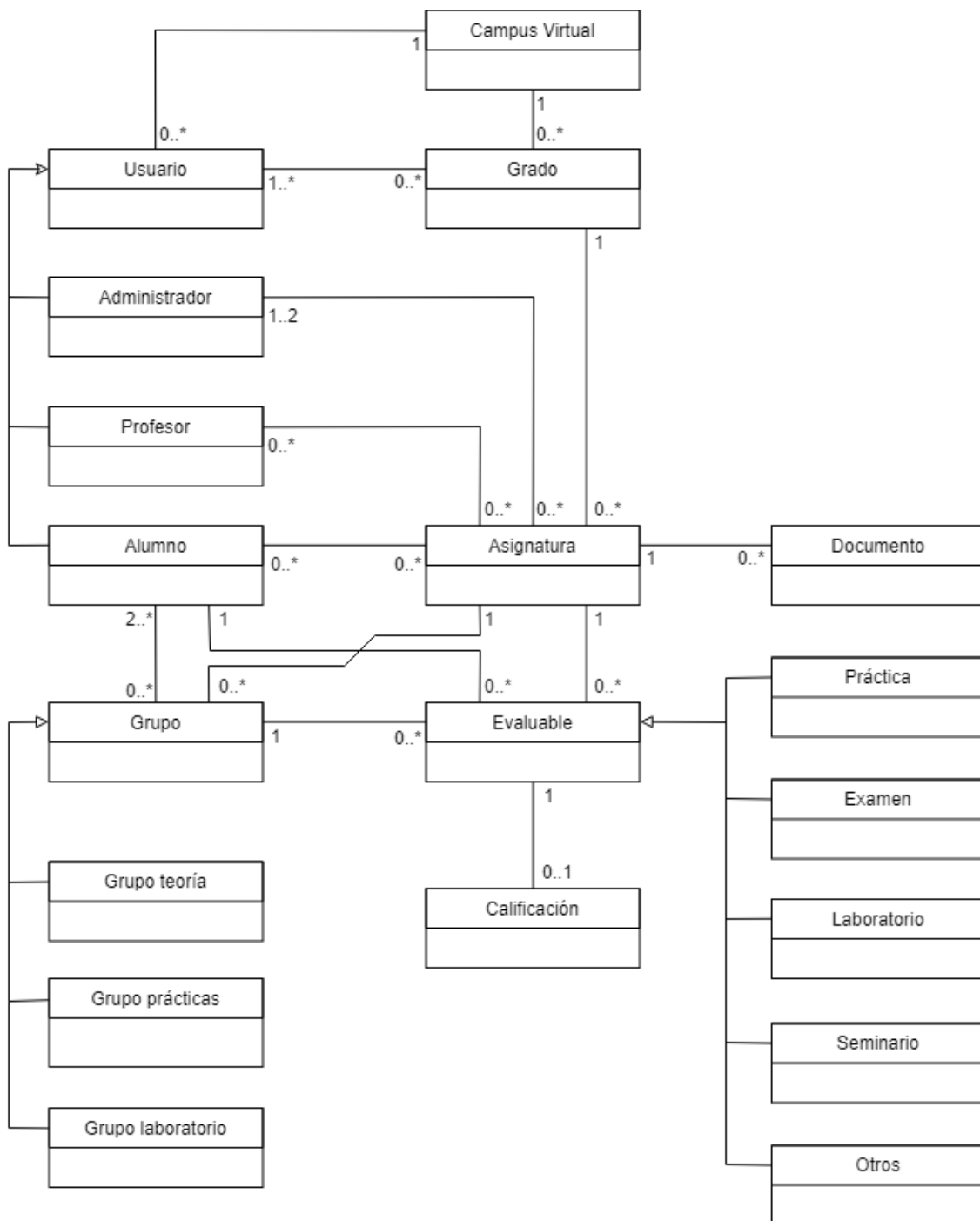


Figura 4.1: Entidades del dominio

- Una asignatura estará asociada a 0 o más grupos, ya que pueden existir asignaturas que no tengan ningún grupo específico. Un grupo estará asociado a una asignatura, ya que de existir el mismo grupo en dos asignaturas distintas, se tratará como dos grupos independientes.
- Los grupos de teoría, laboratorio y prácticas, heredarán de grupo y tendrán funcionalidades diferenciadas.
- Una asignatura estará asociada a 0 o más evaluables, ya que puede existir una asignatura que no tenga ningún evaluable (inicio de curso...). Un evaluable estará asociado a una única asignatura.
- Una asignatura estará asociada a 0 o más documentos, ya que puede existir una asignatura sin documentación, y un documento estará asociado a una única asignatura, ya que de estar relacionado con más de una, se tratará como documentos independientes.
- Un evaluable estará asociado a un único alumno o a un único grupo. Un grupo o alumno estará asociado a 0 o más evaluables.
- Un evaluable estará asociado a 0 o una calificación, ya que puede estar o no calificado. Una calificación corresponderá a un único evaluable.
- Los evaluables de tipo práctica, examen, laboratorio, seminario y otros, heredarán de evaluable y tendrán funcionalidades distintas.

4.4.2. La información en SharePoint

Una buena estructuración de la información dentro de SharePoint, es un punto fundamental a la hora de facilitar la tarea del desarrollo. La existencia de una jerarquía clara y coherente, es requisito necesario en toda granja. A la hora de decidir cómo adaptar el modelo de objetos, se ha de tener en cuenta las distintas posibilidades.

En cualquiera de las alternativas, resulta lógico pensar en una **aplicación web** única, que represente todo el campus virtual, pues no existe ninguna razón que justifique dividirlo en varias.

En cuanto a **colecciones de sitio**, surgen dos alternativas:

- **Una colección de sitios por cada departamento** dentro de la universidad, que actúe como contenedor de las asignaturas. Esta opción podría ser viable en otro tipo de aplicaciones relacionadas con la investigación pero, en este caso, no resulta muy adecuada debido a que difumina el concepto de grado, que pasaría a ser una asociación de sitios de distintas colecciones de sitios. Adicional a ello, no queda muy claro dónde colocar asignaturas como las prácticas en empresa o el trabajo de fin de grado.

- **Una colección de sitios por cada grado y por cada mención.** Esta opción plantea incluir las asignaturas comunes en una colección de sitios genérica, y las asignaturas de las distintas menciones en una colección de sitios específica. A nivel conceptual, existe una cuestión fundamental que lleva a rechazar esta opción, y es la existencia de asignaturas de una mención específica, incluidas dentro de otra. Colocar estas asignaturas como genéricas sería un error porque no tienen por qué estar presentes en todas las menciones, pero colocarlas en una mención específica presentaría la incógnita de cómo relacionar otra mención distinta con ella.
- **Una colección de sitios por cada grado.** Se trataría de una colección de sitios única para todo el grado, que incluya todas las asignaturas de todas las menciones. Esta alternativa es la más viable para evitar problemas de localización de los contenidos y parece la más adecuada a nivel conceptual.

Respecto a los **sitios**, resulta bastante lógico asignar a cada asignatura, un sitio independiente. Al no existir relaciones de jerarquía entre asignaturas, no se considera necesario utilizar el concepto de **subsitio**, al menos de forma inicial.

Los usuarios estarán almacenados en Active Directory y se sincronizarán con SharePoint mediante un procedimiento planificado que se ejecuta constantemente que generará los correspondientes **usuarios SharePoint**. Para cada asignatura, se creará un grupo SharePoint para profesores y otro para alumnos. De forma adicional, se crearán grupos SharePoint para grupos de laboratorio, prácticas y teoría, a razón de un grupo SharePoint por cada grupo existente. El hecho de que los usuarios estén almacenados en Active Directory, implica que un cambio en cualquiera de los datos de un usuario, se verá reflejado en ese usuario SharePoint en la próxima sincronización, evitando problemas de datos duplicados y facilitando tareas como el cambio de contraseña.

Las documentos se crearán como **elementos de lista dentro una biblioteca de documentos** única para cada asignatura. Los evaluables se crearán como **elementos de lista con un tipo de contenido para cada uno**, dentro de la biblioteca de la asignatura. De forma adicional, un evaluable generará un **elemento de lista con un tipo de contenido específico** en una lista que almacenará los datos del mismo.

Las calificaciones se almacenarán como **elementos de lista con un tipo de contenido específico**, en una lista específica para las calificaciones.

4.4.3. Listas, tipos de contenido y columnas

Columnas de sitio

Para la representación de la totalidad de la información, se ha identificado la necesidad de crear las siguientes columnas:

- **Alumno:** Columna de tipo usuario que almacenará alumnos o grupos de prácticas.
- **Código:** Columna que almacenará un código para identificar unívocamente un evaluable.
- **Calificación:** Columna que almacenará la calificación de un alumno en un determinado evaluable.
- **Tipo:** Columna que almacenará el tipo de evaluable entre una lista fija de posibles valores.
- **Fecha:** Columna que almacenará una fecha.
- **Ponderación:** Columna que almacenará una ponderación entre 0 y 1, sobre el total de la asignatura, para la calificación de un evaluable.
- **Nombre:** Columna que almacenará el nombre de un evaluable.
- **Descripción:** Columna que almacenará la descripción de un evaluable.
- **Estado entrega:** Columna que almacenará el estado de una entrega entre una lista fija de posibles valores.
- **Comentarios:** Columna que almacenará los datos comentarios a una entrega.
- **Tarea:** Columna que almacenará el nombre de una tarea.
- **Estado tarea:** Columna que almacenará el estado de una tarea asociada a una entrega.
- **ID:** Id único de un elemento.

Tipos de contenido y listas

Dado que, como se ha expuesto, se utilizará un único tipo de contenido por cada lista, se expondrán listas y tipos de contenido como un único concepto.

Para la representación de la totalidad de la información, se ha detectado la necesidad de creación de las siguientes listas y bibliotecas:

- **Calificaciones:** Lista que almacena calificaciones. Compuesta por Alumno, Código y Calificación
- **Evaluables:** Lista que almacena los datos de los evaluables. Compuesta por Nombre, Descripción, Código, Tipo, Fecha y Ponderación.
- **Tarea:** Lista que almacena los datos de las tareas. Compuesta por Tarea, Estado Tarea e Id.
- **Documentación:** Biblioteca de documentos estándar para almacenar la documentación genérica de una tarea.
- **Entregas:** Biblioteca de documentos que almacena entregas y documentos simples. Las entregas serán conjuntos de documentos con las siguientes columnas: Código, Alumno y Comentarios.

4.4.4. Niveles de permisos

Dentro de cada sitio, se definirán cuatro roles de permisos:

- **Visitante:** Persona ajena a la asignatura que necesita, por algún motivo, tener acceso a los contenidos genéricos de la misma. Deberá obtener permisos de lectura únicamente de los contenidos genéricos de la asignatura.
- **Alumno:** Alumno de la asignatura. Deberá obtener permisos de lectura sobre los contenidos genéricos de la asignatura y permisos específicos sobre aquellos evaluables asignados que deba modificar.
- **Profesor:** Profesor de la asignatura. Deberá obtener permisos para modificar los contenidos genéricos de la asignatura.
- **Administrador:** Administrador de SharePoint, con permisos de super usuario.

Cabe destacar que el hecho de que un usuario no tenga permisos para realizar una acción no significa que no pueda realizarla, sino que no puede realizarla utilizando la interfaz estándar de SharePoint. Una buena implementación de SharePoint se basa en el principio de menor privilegio posible y se utilizan mecanismos para proporcionar a un usuario permisos específicos únicamente para realizar acciones concretas y esto es muy utilizado en los desarrollos.

Capítulo 5

Planificación

Antes de comenzar cualquier proyecto, es necesario realizar una planificación del mismo. Para este proyecto, se considera la utilización de una metodología ágil adaptada a las circunstancias.

En este capítulo se describirá la forma de trabajar en el proyecto, partiendo de la planificación inicial y terminando con una sección en la que se describirá brevemente el resultado de dicha planificación una vez ejecutado el proyecto.

5.1. Metodología utilizada

Para la planificación del proyecto, se seguirá una metodología similar a SCRUM, adaptada a un proyecto de fin de carrera. El equipo de desarrollo estará formado únicamente por el alumno y el tutor asumirá un doble rol, por un lado como cliente, y por otro como responsable de proyecto.

El proyecto se organizará en sprints con una duración de 14 días. Se define el inicio del sprint el martes de la primera semana, y su finalización el lunes de la segunda. Al principio de cada sprint, el martes, se llevará a cabo una reunión con el tutor, donde se realizará una evaluación del resultado del sprint anterior y se planificará el siguiente en función de los resultados obtenidos.

La planificación inicial consistirá en la creación de un product backlog con todos los entregables a generar y la definición, en base a dicho backlog, de las tareas a realizar en el primer sprint. La definición de los siguientes dependerá del resultado anterior, por lo que no tiene sentido realizar una planificación en un marco temporal más amplio.

5.2. Product Backlog

El análisis de la solución propuesta, genera una serie de tareas a realizar, agrupadas en entregables que serán representados como elementos del backlog. Los elementos son los siguientes:

- **Documentación:** Creación de la memoria final de entrega de trabajo de fin de grado.
- **Creación máquina virtual:** Creación de máquina virtual Hyper-V e instalación de Windows Server 2012 R2.
- **Instalación de prerequisites:** Instalación de los prerequisites de SharePoint 2016, como .NET Framework.
- **Instalación de características:** Instalación de las características necesarias para utilizar SharePoint, destacando entre ellas la creación de un controlador de dominio Active Directory.
- **Instalación de SQL Server:** Instalación de SQL Server 2016 developer en la máquina virtual, junto con las características y configuración recomendada para SharePoint 2016.
- **Instalación SharePoint:** Instalación y configuración inicial de SharePoint Server 2016, que implica la creación de una granja de servidor único.
- **Instalación Visual Studio:** Instalación de Visual Studio Enterprise 2015, junto con los componentes necesarios para desarrollar soluciones SharePoint, como el paquete Office developer tools for Visual Studio.

- **Configuraciones finales:** Configuraciones finales para conseguir una granja completamente funcional.
- **Creación de usuarios:** Creación de perfiles de usuario de prueba en Active Directory, con los que interactuar con SharePoint.
- **Creación de Web Application:** Creación y configuración de la aplicación web del Campus Virtual, e inclusión de la URL en el fichero hosts.
- **Script creación Site Collection:** Script para crear las colecciones de sitio según el formato establecido.
- **Script creación de Sites:** Script para crear los sitios según el formato establecido.
- **Script creación de Site Columns:** Script para crear las columnas de sitio en el sitio raíz de una colección de sitios.
- **Script creación de Content Types:** Script para crear los tipos de contenido en el sitio raíz de una colección de sitios.
- **Script creación de listas y bibliotecas:** Scripts para crear cada una de las listas y bibliotecas utilizadas en el desarrollo, en un sitio concreto.
- **Webpart Calificaciones:** Creación de webpart para mostrar las calificaciones de un alumno.
- **Webpart Listado Grupos:** Creación de webpart para mostrar los grupos de la asignatura, listando sus miembros y algunas de sus propiedades.
- **Webpart Notas:** Creación de un webpart que permita el intercambio de notas entre usuario basado en el webpart de caja de SharePoint.
- **Webpart propiedades Entrega:** Creación de webpart para mostrar las propiedades de una entrega a modo tabla.
- **Webpart Noticias:** Combinación del webpart de búsqueda de SharePoint con una DisplayTemplate para mostrar las noticias de una asignatura.
- **Webpart Gestor Tareas:** Creación de webpart para permitir la creación de tarjetas clasificadas por estado, con las tareas asociadas a una entrega, y gestionar las mismas.
- **Webpart Enlaces:** Webpart que proporcione a los alumnos un listado con los enlaces a los sitios de las asignaturas en las que están matriculados.
- **Webpart Calificar:** Webpart que permita a los profesores la calificación de un evaluable.
- **Webpart Creación Entrega:** Webpart que permita la creación de una entrega, generando los correspondientes elementos para cada usuario.

- **Webpart Creación Noticias:** Webpart que permita la creación de noticias para una asignatura, por parte de un profesor.
- **Display Template Noticias:** Creación de una plantilla para mostrar resultados de búsqueda.
- **TimerJob Entrega Proyectos:** Creación de una tarea programada que finalice las tareas cuando la fecha de entrega haya vencido.
- **Evento ItemUpdating Enentregas:** Creación de eventos para la gestión de la actualización de las entregas.
- **Maquetación de sitio de asignatura:** Creación de sitio para presentar la asignatura.

5.3. Planificación inicial

El primer sprint comenzará el martes 27 de marzo de 2018 y finalizará el 9 de abril de 2018. Se establecerá la reunión mencionada el martes 10 de abril de 2018.

Los elementos del backlog asociados al primer sprint serán los siguientes:

Backlog	Tarea
Creación máquina virtual	Creación de máquina Hyper-V Instalación de Windows
Instalación prerequisites	Instalación prerequisites
Instalación características	Instalación características Creación Active Directory Configuración de un dominio
Instalación SQL Server	Instalación SQL Server Instalación características adicionales
Instalación SharePoint	Instalación SharePoint Creación granja Configuración granja
Instalación Visual Studio	Instalación Visual Studio Instalación Office Dev Tools
Configuraciones finales	Configuración granja Configuración características

Cuadro 5.1: Tareas Sprint Inicial

5.4. Ejecución de la planificación

En esta sección se expondrán las tareas asociadas a los distintos sprints, junto con el resultado de los mismos y los motivos existentes para la desviación de la planificación si la hubiese.

5.4.1. Sprint 1

Fecha de inicio	27 de marzo de 2018
Fecha de fin	10 de abril de 2018

Cuadro 5.2: Fechas Sprint 1

Backlog	Tarea	Estado
Creación máquina virtual	Creación de máquina Hyper-V	OK
	Instalación de Windows	OK
Instalación prerequisites	Instalación prerequisites	OK
	Instalación características	OK
Instalación características	Creación Active Directory	OK
	Configuración de un dominio	OK
Instalación SQL Server	Instalación SQL Server	OK
	Instalación características adicionales	OK
Instalación SharePoint	Instalación SharePoint	OK
	Creación granja	OK
	Configuración granja	OK
Instalación Visual Studio	Instalación Visual Studio	OK
	Instalación Office Dev Tools	OK
Configuraciones finales	Configuración granja	OK
	Configuración características	OK

Cuadro 5.3: Sprint 1

El sprint 1 fue un sprint inicial que sirvió como toma de contacto y se completó sin complicaciones.

5.4.2. Sprint 2

Fecha de inicio	10 de abril de 2018
Fecha de fin	23 de abril de 2018

Cuadro 5.4: Fechas Sprint 2

Backlog	Tarea	Estado
Creación de usuarios	Creación de dominio	OK
	Creación usuarios	OK
Script creación colecciones sitios	Análisis	OK
	Planteamiento de alternativas	OK
	Selección de alternativa	OK
	Creación script	OK
	Pruebas	OK
Script creación sitios	Análisis	OK
	Planteamiento de alternativas	OK
	Selección de alternativa	OK
	Creación script	OK
Script creación columnas	Pruebas	OK
	Análisis	OK
	Planteamiento de alternativas	OK
	Selección de alternativa	OK
Script creación tipo contenido	Creación script	OK
	Pruebas	OK
	Selección de alternativa	OK
	Planteamiento de alternativas	OK
Documentación	Creación de estructura memoria	OK
	Sección introducción	KO
	Sección SharePoint	KO

Cuadro 5.5: Sprint 2

En este sprint, el exceso de carga obligó a retrasar gran parte de la tarea de documentación al siguiente sprint, principalmente justificado por razones de índole laboral.

5.4.3. Sprint 3

Fecha de inicio	24 de abril de 2018
Fecha de fin	7 de mayo de 2018

Cuadro 5.6: Fechas Sprint 3

Backlog	Tarea	Estado
Webpart calificaciones	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
Webpart listado de grupos	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	KO
	Crear CSS	KO
	Poblar listas prueba	KO
Realizar prueba	KO	
Documentación	Sección Introducción	KO
	Sección SharePoint	KO
	Sección Sección Planteamiento	KO

Cuadro 5.7: Sprint 3

El sprint 3 fue el sprint más complicado dentro del proyecto, ya que coincidió con el final de un proyecto en la empresa y la situación ocupó gran parte del tiempo que estaba asignado al trabajo de fin de grado, es por ello que únicamente se pudo elaborar un webpart y plantear otro, dejando la tarea de la documentación para futuros sprints.

5.4.4. Sprint 4

Fecha de inicio	8 de mayo de 2018
Fecha de fin	21 de mayo de 2018

Cuadro 5.8: Fechas Sprint 4

Backlog	Tarea	Estado
Webpart listado de grupos	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
Webpart propiedades entrega	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
Documentación	Sección Introducción	OK
	Sección SharePoint	OK
	Sección Sección Planteamiento	KO

Cuadro 5.9: Sprint 4

El sprint 4 se planificó con una carga de trabajo mucho más inferior a los anteriores, puesto que la situación laboral todavía no permitía dedicar todo el tiempo posible al proyecto. Se terminó de realizar el trabajo pendiente en el sprint 3 y se creó alguno de los webparts más sencillos. Se comenzó también a rellenar parte de la memoria.

5.4.5. Sprint 5

Fecha de inicio	22 de mayo de 2018
Fecha de fin	4 de junio de 2018

Cuadro 5.10: Fechas Sprint 5

Backlog	Tarea	Estado
Webpart gestor de tareas	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
Webpart creación de noticias	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
Documentación	Sección Planteamiento	OK
	Sección Desarrollo (webparts ya realizados)	OK
	Sección Planificación	KO

Cuadro 5.11: Sprint 5

Este sprint fue el de mayor carga de desarrollo, pues el webpart de gestor de tareas es el que mayor coste en horas de desarrollo conllevaba. Se realizó una carga tan elevada para compensar la falta de trabajo de los dos anteriores.

5.4.6. Sprint 6

Fecha de inicio	5 de junio de 2018
Fecha de fin	18 de junio de 2018

Cuadro 5.12: Fechas Sprint 6

Backlog	Tarea	Estado
Webpart creación de entregas	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
Webpart calificar	Análisis	OK
	Planteamiento	OK
	Completar script columnas	OK
	Completar script tipos contenido	OK
	Crear script listas	OK
	Crear ASCX	OK
	Crear Code Behind	OK
	Crear CSS	OK
	Poblar listas prueba	OK
	Realizar prueba	OK
DisplayTemplate noticias	Análisis	OK
	Creación del servicio de búsqueda	OK
	Rastreo para generar crawled properties	OK
	Creación de las managed properties	OK
	Creación del origen de contenidos	OK
	Rastreo para poblar managed properties	OK
	Creación de html template	KO
	Creación de webpart resultados de búsqueda	KO

Cuadro 5.13: Sprint 6

En este sprint, se llevaron a cabo los webparts de calificación de un evaluable y generación de entregas para todos los alumnos. Se intentó realizar las display templates, pero únicamente se pudo llevar a cabo la parte de configuración del servicio de búsqueda relativa, por falta de conocimiento a la hora de crear las plantillas. No se realizó ninguna tarea de documentación en este sprint.

5.4.7. Sprint 7

Fecha de inicio	19 de junio de 2018
Fecha de fin	2 de julio de 2018

Cuadro 5.14: Fechas Sprint 7

Backlog	Tarea	Estado
ItemUpdating entregas	Análisis	OK
	Planteamiento	OK
	Crear Execute	OK
	Poblar listas prueba	OK
	Realizar prueba ejecución	OK
Documentación	Sección Introducción (modificación)	OK
	Sección SharePoint (modificación)	OK
	Sección Planteamiento (modificación)	OK
	Sección Planificación	OK
	Sección Desarrollo de scripts	KO
	Sección Desarrollo	OK
	Sección Bibliografía	KO
	Sección Conclusiones y trabajo futuro	KO
	Agradecimientos	KO
	Resumen	KO
Anexos	KO	
DisplayTemplate noticias	Creación de html template	KO
	Subida a SharePoint de la template	KO
	Creación de webpart resultados de búsqueda	KO
	Creación query de búsqueda	KO
	Asignación de la plantilla a los resultados	KO
	Poblar listas prueba	KO
	Realizar prueba	KO

Cuadro 5.15: Sprint 7

En un inicio, este sprint incluía el fin de la primera convocatoria, por lo que debía ser el sprint final. La carga de este sprint estuvo basada en la finalización de los desarrollos y la documentación completa. En la práctica, la entrega se retrasó hasta el 11 de julio, generando un nuevo sprint.

5.4.8. Sprint 8

Fecha de inicio	2 de julio de 2018
Fecha de fin	11 de julio de 2018

Cuadro 5.16: Fechas Sprint 7

Este sprint fue el sprint final, con una duración más corta y mucha carga de trabajo. Durante el sprint se finalizaron todas las tareas necesarias para completar el proyecto, unido a tareas de gestión, tales como generar los entregables y solicitar la documentación pertinente. La comunicación con el tutor durante este sprint ha sido continua, manteniendo varias reuniones.

La creación de las noticias, unido a **la creación de las DisplayTemplates tuvo que descartarse** en las fases finales del proyecto, debido a la limitación de recursos del mismo. El uso del servicio de búsqueda exige un mínimo de 24GB de memoria RAM para un correcto desempeño, una cifra muy alejada de los recursos disponibles. Es por ello, que se decidió **excluir el webpart de creación de noticias del proyecto**, dado que sin las DisplayTemplates, este no tenía sentido.

A pesar de las dificultades y del tiempo tan ajustado, se pudieron completar con éxito todas las tareas y se consideró el proyecto como cerrado.

Capítulo 6

Provisionamiento y configuración

La correcta creación de una estructura en SharePoint, implica el desarrollo de multitud de scripts PowerShell para el ajuste de parámetros, y la configuración de distintos servicios propios de la plataforma.

En esta sección se describirán los scripts más relevantes para el despliegue de la estructura, junto con las configuraciones realizadas en los servicios para poder utilizarlos en desarrollos posteriores.

El contenido de los scripts se adjuntará como anexo a la memoria, ya que no se considera que incluirlo en esta sección sea apropiado.

6.1. Script creación de colecciones de sitios

A partir de los parámetros de entrada, el script debe ser capaz de crear una nueva colección de sitios, representando un grado, en la aplicación web del Campus Virtual, con lenguaje 1033 (EN-US) y cuya plantilla sea la STS-0 (Team Site).

Los parámetros de entrada serán los siguientes:

- URL de la colección de sitios.
- Alias de la colección de sitios, que se utilizará para formar la URL.
- Título de la colección de sitios.

En el script, se designará un administrador principal de la colección de sitios para todas y cada una de las colecciones de sitio creadas, ya que es necesario para la creación. La asignación de un administrador secundario podrá realizarse a posteriori, utilizando este usuario.

La creación de la colección de sitios implica la creación de un sitio raíz, con la URL de la propia colección. Dicho sitio deberá estar creado con la misma plantilla y lenguaje de la colección de sitios.

6.2. Script creación de sitios

A partir de los parámetros de entrada, el script debe ser capaz de crear un nuevo sitio dentro de una colección de sitios, representando una asignatura, con lenguaje 1033 (EN-US) y cuya plantilla sea la STS-0 (Team Site).

Los parámetros de entrada serán los siguientes:

- URL de la colección de sitios.
- Alias del sitio, que se utilizará para formar la URL.
- Título del sitio.

Después de la creación del sitio, el script deberá crear en él los grupos de profesores y alumnos, asignando a estos los roles de permisos contribute y read, respectivamente. El administrador de dichos grupos debe ser el propio administrador principal de la colección de sitios.

6.3. Script creación de columnas

A partir de los parámetros de entrada, el script deberá ser capaz de crear todas las columnas de sitio utilizadas en los desarrollos del proyecto, sobre el sitio raíz de una colección de sitios.

Los parámetros de entrada serán los siguientes:

- URL de la colección de sitios.

Las columnas se crearán indicando el tipo de datos que almacenan, su nombre y si son requeridas o no a la hora de crear un elemento que incluya las mismas.

Siguiendo la indicación de Microsoft, las columnas de sitio deberán ser creadas con un `InternalName` (definido en la creación y que posteriormente no puede ser cambiado) que comience con un prefijo que sirva para identificarlas.

Posterior a la creación, se modificará la propiedad `DisplayName` para asignar a las columnas el valor correspondiente, y además, se utilizará la propiedad `Group` para clasificar las mismas con un nombre distintivo.

6.4. Script de creación de tipos de contenido

A partir de los parámetros de entrada, el script deberá ser capaz de crear todos los tipos de contenido utilizados en los desarrollos del proyecto, partiendo de las columnas de sitio creadas en el script del punto anterior, sobre el sitio raíz de una colección de sitios.

Los parámetros de entrada serán los siguientes:

- URL de la colección de sitios.

Los tipos de contenido de lista (calificación, evaluable y tarea) heredarán del tipo de contenido `System`, mientras que el tipo de contenido de conjunto de documentos (entrega) heredará de `Document Set`.

Posterior a la creación, se modificará la propiedad `Group` para clasificar los tipos de contenido con un nombre distintivo.

6.5. Script de creación de listas

A partir de los parámetros de entrada, el script deberá ser capaz de crear todas las listas utilizadas en los desarrollos del proyecto, partiendo de los tipos de contenido creados en el script del punto anterior, sobre un sitio concreto relativo a una asignatura.

Los parámetros de entrada serán los siguientes:

- URL del sitio.

Como parte adicional, el script deberá romper la herencia de permisos de la lista de calificaciones, ya que los alumnos no deberían poder visualizar el contenido de la misma.

Siguiendo las recomendaciones de Microsoft, todas las listas utilizarán la plantilla de lista personalizada (Custom List) y utilizará un único tipo de contenido por lista para su creación, sin utilizar columnas simples dentro de la estructura.

6.6. Configuración de servicios

En este apartado se realizará una pequeña descripción de las configuraciones realizadas en las aplicaciones de servicio necesarias para los desarrollos.

6.6.1. Servicio de búsqueda

El servicio de búsqueda es uno de los elementos más importantes en SharePoint debido a su gran eficiencia a la hora de indexar y localizar contenidos. Dispone de una aplicación de servicio propia dentro de SharePoint que debe ser configurada. Esta aplicación es bastante complicada de configurar cuando hablamos de granjas de varios servidores, ya que existen 5 roles en la topología de búsquedas que deberán ser ubicados en los distintos servidores. En el caso de una granja Standalone, esto no suele generar problemas.

La búsqueda de SharePoint se basa en la realización de un rastreo y una indexación previa, para posteriormente localizar los contenidos de una forma mucho más eficiente. Esto provoca una elevada carga de trabajo en el servidor que una máquina Standalone no soporta muy bien, por lo que únicamente se podrá realizar un ejemplo sencillo y con un rastreo bajo demanda y no programado, adaptado a las posibilidades reales.

Una vez creada la aplicación de servicio, es necesario tener en cuenta una serie de parámetros:

- **Origen de contenidos:** Representa el conjunto de sitios o colecciones de sitios que serán rastreadas. Se indica mediante una o varias URLs y, habitualmente, se rastrean todos los contenidos debajo de ellas. SharePoint no es capaz de rastrear de forma concurrente sobre un mismo origen de contenidos, por lo que resulta fundamental una buena definición de orígenes de contenido, unida a una buena programación temporal de los mismos, para garantizar un equilibrio de carga y maximizar la eficiencia. Se creará un origen de contenidos simple para toda la aplicación web, aunque en un entorno productivo esto no debería ser así.
- **Crawled Property:** Representa cada una de las propiedades de cada una de las columnas existentes en todas las listas de todos los sitios de la granja. De forma resumida, una crawled property es todo metadato que puede ser rastreable en algún elemento SharePoint. Estos elementos obtendrán los valores específicos de cada documento durante un rastreo (y se crearán si no existen). Los desarrolladores no pueden alterar estas propiedades directamente.
- **Managed Property:** Dentro del conjunto de las crawled properties, la mayor parte de ellas nunca se utilizarán con filtro en una búsqueda. Para evitar un consumo excesivo de recursos en la indexación, SharePoint utiliza Managed Properties. Cada una de ellas referencia a una o varias crawled properties, indicando a SharePoint que dichas propiedades deben ser indexadas y poniendo a disposición de los desarrolladores / usuarios la propiedad como filtro de búsqueda. En el ejemplo que se desarrollará con las DisplayTemplates, será necesario utilizar 3 propiedades.

6.6.2. Servicio de perfiles de usuario

El servicio de perfiles de usuario gestiona la conexión de los datos de los usuarios entre SharePoint y el Active Directory, permitiendo una sincronización unilateral o bilateral. Los perfiles de usuario almacenan perfiles de usuario como listados de propiedades mapeadas con Active Directory en relaciones 1 a 1, es decir, una propiedad de SharePoint debe tener una propiedad equivalente en Active Directory, no pudiendo generar propiedades calculadas...

Los objetos usuario en SharePoint almacenan alguna de estas propiedades más importantes (Name, LoginName...) por lo que es fundamental que esto esté configurado correctamente a la hora de, por ejemplo, mostrar el nombre correcto de un usuario. Para el acceso a propiedades más específicas, es necesario abrir una conexión implícitamente dentro del desarrollo.

Esta aplicación de servicio es muy extensa, pues proporciona entre otras cosas las funcionalidades sociales. Únicamente se utilizará para crear una conexión de sincronización con el dominio asignado al TFG y poder recuperar alguna de las propiedades de los usuarios a modo de demostración.

6.6.3. Servicio de metadatos administrados

El servicio de metadatos administrados es un servicio core de SharePoint que se utiliza para muchas de las funcionalidades que se desarrollarán en el trabajo, aunque ninguna de ellas lo hará de forma directa.

El servicio se utiliza fundamentalmente para almacenar palabras clave y variables de configuración. En la sección de trabajo futuro se expondrán algunas posibilidades que ofrecería el servicio bien utilizado, en una aplicación de este tipo.

Capítulo 7

Desarrollo de soluciones

El desarrollo de soluciones en SharePoint, al menos la parte atañe a este proyecto, se basa en la creación de elementos cuya funcionalidad es independiente del resto de elementos. Es por ello que su desarrollo se realiza en un ámbito cerrado, entendiéndose por ello que el análisis, diseño, implementación y pruebas, se realizan de forma individual y sólo al final del proyecto, se realizará una prueba de integración de los elementos de forma conjunta.

En cuanto a la toma de requisitos, se realizará únicamente la parte de requisitos funcionales, ya que los no funcionales vienen determinados, en su mayoría, por propio sistema y no dependen del desarrollo concreto. En este contexto, se dará por entendido que todos los desarrollos implican requisitos no funcionales básicos en el desarrollo web, tales como un tiempo de respuesta entre los límites tolerables para los usuarios. La eficiencia en la mayoría de los casos está unida al uso correcto de las herramientas proporcionadas por Microsoft, por lo que se asumirá como requisito no funcional, el desarrollo siguiendo las recomendaciones de Microsoft en su catálogo de buenas prácticas.

En este capítulo se dedicará un punto a cada una de las funcionalidades creadas, y un punto adicional a las pruebas de integración.

7.1. Webpart Enlaces asignaturas

El objetivo de este webpart es mostrar, para el usuario actual, un listado de las asignaturas en las que está matriculado dentro de la colección de sitios correspondiente.

7.1.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos funcionales:

- El listado se presentará al usuario como una pequeña tabla formada por enlaces a los correspondientes sitios de cada una de las asignaturas.
- El listado únicamente contendrá enlaces a sitios o subsitios dentro de la misma colección de sitios en la que está incluido el webpart.
- El texto de dichos enlaces, será el mismo que el título del sitio correspondiente a cada uno de ellos.
- El vínculo del enlace deberá redirigir a la página principal del sitio de la asignatura a la que referencia.
- En caso de que no se encuentre ninguna asignatura en la que el alumno esté matriculado, se deberá informar al alumno mediante un mensaje de error.

7.1.2. Funcionalidad utilizada

Para obtener los datos, se realizará una consulta a todos los sitios de la colección de sitios, buscando en cada uno de ellos si el usuario pertenece al grupo de alumnos de la asignatura y, en caso afirmativo, se añadirá a la lista un enlace con el título del sitio, que apunte a la dirección del mismo.

Para poder realizar las consultas sobre cada uno de los sitios, es necesario que el usuario posea determinados permisos que, atendiendo a la definición, un alumno no debería tener. Dichos permisos son aquellos que permiten consultar los grupos de SharePoint asociados a un determinado sitio. Dado que los usuarios únicamente dispondrán del rol de permisos de lectura, no podrán listar los permisos de un sitio, así como tampoco ver los grupos del mismo.

SharePoint proporciona un mecanismo, la ejecución de código con permisos elevados, que permite a un usuario ejecutar un fragmento de código determinado, con permisos de administrador. Utilizando este mecanismo, se logra que únicamente el código incluido dentro del método, se ejecute con el nivel de permisos propio del administrador del pool de aplicaciones, por lo que el usuario podrá obtener acceso a datos a los que no podría en una ejecución normal, sin realizar ningún tipo de modificación en sus permisos específicos.

El uso de bloques de código ejecutados con privilegios elevados, debe ser lo más estricto y riguroso posible, ya que podría generar un agujero de seguridad en el sistema. Es por ello que se recomienda utilizar esta técnica únicamente en aquellas partes del código en las que sea estrictamente necesario.

Se ha de tener en cuenta, de forma adicional, que el usuario administrador del pool de aplicaciones tampoco debe utilizarse como comodín, es decir, debe obtener permisos únicamente como administrador del entorno SharePoint y NUNCA como administrador local del servidor, pues se estaría comprometiendo el sistema de forma innecesaria. Es un caso muy típico de granjas SharePoint, la asignación de permisos arbitrariamente a dicho usuario, como solución a muchos de los problemas que se plantean, bien por una mala gestión de permisos, bien por el uso de técnicas inadecuadas.

7.1.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Título del webpart	Texto estático	Título HTML
Mensaje de error	Texto estático	ASP Label
Listado de enlaces	Listado no ordenado de enlaces	ASP Repeater

Cuadro 7.1: Elementos web: Webpart Enlaces

Se crearán las siguientes clases adicionales:

- **EnlaceAsignatura:** Clase auxiliar, ajena al dominio del problema, que representa un enlace a una asignatura. Se utilizará como contenedor de los datos necesarios para rellenar un elemento del repeater. Sus atributos son:
 - Nombre: Representará el título del enlace
 - Enlace: Representará la URL del enlace

El flujo de carga de la página será el siguiente:

- Se creará una lista vacía de objetos de tipo EnlaceAsignatura.
- Se utilizará el objeto SPContext para obtener los objetos de colección de sitios y usuario actuales, sobre los que se realizará la consulta.
- Se iterará a lo largo de todos los sitios y subsitios pertenecientes a la colección de sitios actual, y se ubicará, dentro de cada uno de ellos, el grupo cuyo nombre contenga “alumnos”.
- Se buscará en el listado de grupos del usuario, si existe el grupo de alumnos del punto anterior y, en caso afirmativo, se añadirá a la lista de EnlaceAsignatura, un nuevo objeto EnlaceAsignatura con:

- Nombre: Título del sitio
 - Enlace: URL del sitio
- Al terminar la iteración, si la lista de EnlaceAsignatura contiene al menos un elemento, se asociará la misma como fuente de datos del repeater y se mostrarán los datos por pantalla. En caso contrario, se ocultará el repeater y se mostrará el mensaje de error.

7.1.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y los sitios y subsitios asociados a la asignatura, deben estar correctamente creados y accesibles.
- Los sitios y subsitios de las asignaturas, deben contener un título y enlace que cumplan los estándares definidos.
- En caso de estar matriculado en una asignatura, el alumno deberá figurar como miembro del grupo de alumnos, correctamente creado siguiendo los estándares.
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de sitios o subsitios asociados a una colección de sitios que represente un grado.

El sistema debe reaccionar con un comportamiento específico ante las siguientes excepciones:

- Imposibilidad de acceder a la colección de sitios: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a un sitio específico o una propiedad dentro del mismo necesaria para la obtención de los datos: Escribir la excepción en el log de SharePoint, omitir dicho sitio y continuar comprobando el resto de los sitios restantes.
- Alumno no matriculado en ninguna asignatura: Mostrar al usuario el mensaje de error.

7.1.5. Batería de pruebas

Se definirán 6 casos de prueba para este webpart, en función del tipo de usuario que esté visualizando el webpart o de las circunstancias del mismo:

- **Usuario administrador:** El administrador no debe estar matriculado en ninguna asignatura, por lo que el webpart debe mostrar un mensaje indicando que no se encuentran asignaturas para el usuario. **Resultado OK**
- **Usuario profesor:** El profesor no debe estar matriculado en ninguna asignatura, por lo que el webpart debe mostrar un mensaje indicando que no se encuentran asignaturas para el usuario. **Resultado OK**
- **Usuario alumno no matriculado:** El webpart debe mostrar un mensaje indicando que no se encuentran asignaturas para el usuario. **Resultado OK**
- **Usuario alumno matriculado en una asignatura:** El webpart debe mostrar un enlace a la asignatura correspondiente. **Resultado OK**
- **Usuario alumno matriculado en varias asignaturas dentro de la colección de sitios:** El webpart debe mostrar tantos enlaces como asignaturas matriculadas posea el alumno. **Resultado OK**
- **Usuario alumno matriculado en asignaturas, tanto dentro como fuera de la colección de sitios:** El webpart debe mostrar tantos enlaces como asignaturas matriculadas posea el alumno, únicamente dentro de la propia colección de sitios, ignorando el resto. **Resultado OK**

7.2. Webpart Mis calificaciones

El objetivo de este webpart es mostrar, para el usuario actual y la asignatura actual, un listado de las calificaciones obtenidas en los distintos elementos evaluables de la asignatura.

7.2.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos:

- El listado mostrará al usuario las calificaciones obtenidas en la asignatura correspondiente al sitio en el que el webpart esté ubicado.
- El listado se presentará al usuario en forma de tarjetas, incluyendo en primera posición una tarjeta con la nota media de la asignatura, seguido de tantas tarjetas como notas tenga el alumno.
- El formato de la tarjeta con la calificación global, debe ser distintivo respecto al resto de tarjetas, y contendrá los siguientes datos:
 - Calificación media de la asignatura sobre 10 puntos.
 - Calificación media de la asignatura sobre el total de puntos evaluados hasta la fecha.
- El formato de las tarjetas con calificaciones individuales, deberá ser igual para todas ellas, y contendrá los siguientes datos:
 - Título del evaluable
 - Tipo de evaluable
 - Fecha de evaluación
 - Ponderación de la calificación sobre el total de la asignatura, expresada en tanto por 1
 - Calificación del evaluable sobre 10 puntos
 - Calificación ponderada del evaluable, basada en la proporción sobre la nota total de la asignatura
- El listado de calificaciones únicamente deberá mostrar aquellos evaluables que ya hayan sido calificados.
- En caso de que no se encuentre ninguna calificación en la asignatura, se deberá informar al alumno mediante un mensaje.

7.2.2. Funcionalidad utilizada

La funcionalidad de este webpart, recogerá los datos de dos listas de SharePoint localizadas en el sitio donde esté ubicado, y utilizará el usuario actual para realizar un filtrado. Dichas listas son:

- **Lista Calificaciones:** Recogerá las calificaciones de todos los usuarios de la asignatura, a razón de un elemento de lista por cada calificación, de cada usuario, en cada evaluable.
- **Lista Evaluables:** Recogerá los datos de todos los evaluables de la asignatura, a razón de un elemento de lista por cada evaluable existente en la asignatura.

La lista de calificaciones guarda una relación entre evaluable y calificación, mediante la columna código.

Para poder realizar las consultas sobre las listas propuestas, será necesario que los alumnos obtengan permisos de lectura sobre sus elementos. En el caso de la lista de evaluables, la obtención de permisos de lectura no implicaría riesgo alguno, ya que estos son públicos y se pueden obtener mediante el mismo webpart. Un caso bien distinto sería el de la lista de calificaciones, que almacena las calificaciones de todos los usuarios de la asignatura, datos que un alumno no debería conocer.

Ante esta problemática surgen dos opciones:

- Romper la herencia de los elementos para proporcionar permisos de lectura a cada usuario únicamente sobre los elementos de la lista en los que se haga referencia al mismo.
- No dar permisos sobre la lista a ningún alumno, y realizar las consultas bajo sentencias de ejecución con privilegios elevados.

La primera de las opciones requiere un desarrollo a medida adicional, que dispare un evento en el momento en el que se añadan elementos a la lista, rompa la herencia de permisos del elemento y otorgue los permisos correspondientes al usuario. Esta tarea, además de costosa a nivel funcional, puede generar inconsistencias si, por ejemplo, se produjese una excepción a la hora de dar los permisos correspondientes, añadir el elemento... No parece la opción más viable.

La segunda de las opciones únicamente implicaría el uso de una sentencia con privilegios elevados sobre la lista de calificaciones, una consulta totalmente acotada y que no generaría ninguna posibilidad de modificar dichos datos. Esta opción parece mucho menos costosa, más correcta y más segura, por lo que se ha optado por ella.

7.2.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Título del webpart	Texto estático	Título HTML
Mensaje de error	Texto estático	ASP Label
Listado de calificaciones	Listado de cards BootStrap	ASP Repeater

Cuadro 7.2: Elementos web: Webpart Calificaciones

Se crearán las siguientes clases adicionales:

- **Calificacion:** Clase auxiliar, ajena al dominio del problema, que representa una calificación de la asignatura. Se utilizará como contenedor de los datos necesarios para rellenar cada tarjeta. Sus atributos son:
 - Titulo: Título del evaluable
 - Tipo: Tipo de evaluable
 - Fecha: Fecha de entrega del evaluable
 - Fraccion: Ponderación de la nota
 - Nota: Calificación del evaluable
 - NotaProp: Calificación ponderada del evaluable

l flujo de carga de la página será el siguiente:

1. Se creará una lista vacía de objetos de tipo Calificacion.
2. Se utilizará el objeto SPContext para obtener los objetos sitio y usuario actuales, sobre los que se realizará la consulta.
3. Se realizará una consulta a la lista de calificaciones, basada en el ID del usuario actual, donde se extraerán todas las calificaciones de las que el usuario dispone en el sitio.
4. Para cada una de las calificaciones, se utilizará el código de evaluable para obtener los datos del mismo mediante una consulta a la lista de evaluables.
5. Una vez recogidos los datos, se pasarán como parámetro a un método que construye el objeto Calificacion correspondiente.
6. Si el objeto Calificacion obtenido es válido, se añade a la lista inicial de objetos Calificacion.
7. Al terminar la iteración, si la lista de Calificacion contiene al menos un elemento, se asociará la misma como fuente de datos del repeater y se mostrarán los datos por pantalla. En caso contrario, se ocultará el repeater y se mostrará el mensaje de error.

7.2.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y el sitio asociado a la asignatura, deben estar correctamente creados y accesibles.
- El sitio debe contener las listas Calificaciones y Evaluables, accesibles y correctamente creadas según los estándares.
- Todas las calificaciones deben estar asociadas correctamente a un código de evaluable, que debe tener su correspondiente elemento dentro de la lista de evaluables.
- La suma de todas las proporciones de los evaluables de una asignatura debe ser igual al total de la asignatura, no siendo necesario que este sea igual o menor a uno (debido a que pueden existir tareas adicionales para subir nota).
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de sitios o subsitios asociados a una asignatura.

El sistema debe reaccionar con un comportamiento específico ante las siguientes excepciones:

- Imposibilidad de acceder al sitio: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a la lista de calificaciones o evaluables: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de encontrar calificaciones para el usuario: Mostrar al usuario el mensaje de error definido.
- Imposibilidad de encontrar el evaluable correspondiente a una calificación: Escribir la excepción en el log de SharePoint, omitir dicho elemento y continuar comprobando el resto de los elementos restantes.
- Imposibilidad de crear el objeto Calificacion correspondiente: Escribir la excepción en el log de SharePoint, omitir dicho elemento y continuar comprobando el resto de los elementos restantes.

7.2.5. Batería de pruebas

Se definirán 5 casos de prueba para este webpart, en función del tipo de usuario que esté visualizando el webpart o de las circunstancias del mismo:

- **Usuario administrador:** El administrador no debe tener calificaciones en la asignatura, por lo que el webpart debe mostrar un mensaje indicando que no se encuentran calificaciones para el usuario. **Resultado OK**
- **Usuario profesor:** El profesor nno debe tener calificaciones en la asignatura, por lo que el webpart debe mostrar un mensaje indicando que no se encuentran calificaciones para el usuario. **Resultado OK**
- **Usuario alumno sin calificaciones:** El webpart debe mostrar un mensaje indicando que no se encuentran calificaciones para el usuario. **Resultado OK**
- **Usuario alumno con calificaciones:** El webpart debe mostrar una lista con las calificaciones del usuario, precedida por la nota total actual de la asignatura. **Resultado OK**
- **Varios usuarios alumnos con calificaciones:** El webpart debe mostrar una lista con las calificaciones únicamente del usuario actual, precedida por la nota total actual de la asignatura. **Resultado OK**

7.3. Webpart Listado de grupos

El objetivo de este webpart es mostrar, para la asignatura actual, el total de alumnos y profesores matriculados en la asignatura y la división en grupos de prácticas, teoría, laboratorio, mostrando los miembros que pertenecen a cada uno de ellos.

7.3.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos:

- El listado mostrará al usuario, mediante cinco pantallas distintas, los alumnos matriculados en la asignatura, los profesores a cargo de la misma, la división en grupos de teoría y laboratorio y los grupos de prácticas.
- Las vista de alumnos matriculados en la asignatura y la de profesores, se realizará mediante una tabla que mostrará obligatoriamente el nombre, y opcionalmente el teléfono y el correo electrónico de cada usuario si es que este lo facilita.
- Las vistas de grupos de teoría y grupos de laboratorio se presentarán mediante tarjetas, con un máximo de dos tarjetas por fila, cuyo encabezado será el nombre del grupo de teoría y su contenido serán los nombres de los integrantes del mismo.
- La vista de grupos de prácticas se presentará mediante tarjetas, con un máximo de cuatro tarjetas por fila, cuyo encabezado será el nombre del grupo de teoría y su contenido serán los nombres de los integrantes del mismo.

7.3.2. Funcionalidad utilizada

La funcionalidad de este webpart, obtendrá los datos de los usuarios pertenecientes a cada uno de los grupos, de los grupos SharePoint del propio sitio. La búsqueda de los grupos se realizará desde los grupos SharePoint previamente creados para cada asignatura.

Para poder realizar las consultas sobre el sitio y obtener los permisos necesarios para listar los grupos y los usuarios que pertenecen a ellos, es necesario contar con una serie de permisos que alumno o profesor, por definición, no deberían poseer. Es por ello que es necesario ejecutar dicho código dentro de una sentencia con privilegios elevados.

La consulta de propiedades de usuarios se realizará buscando en el servicio de perfiles de usuario de todos los usuarios de la asignatura. Entre los permisos de SharePoint, un usuario puede consultar los datos de su propio perfil de usuario, pero no debería ser capaz de consultar perfiles de otros usuarios, por lo que vuelve a ser necesario el uso de una sentencia con privilegios elevados.

7.3.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Título del webpart	Texto estático	Título HTML
Mensaje de error	Texto estático	ASP Label
Título listado	Texto estático	Título HTML
Desplegable tablas	Desplegable	ASP DropDownList
Tabla alumnos	Tabla simple	ASP Gridview
Listado grupos T/P	Listado de cards BootStrap	ASP Repeater

Cuadro 7.3: Elementos web: Webpart Listado Grupos

Se crearán las siguientes clases adicionales:

- Grupo: Representa cualquiera de los grupos de teoría, laboratorio o prácticas de la asignatura, formado por dos atributos:
 - Numero: Representa el número del grupo
 - Miembros: Cadena que contiene los miembros del grupo

El flujo de carga de la página será el siguiente:

1. Se fijará el el texto de la etiqueta correspondiente a los alumnos.
2. Se invocará al método encargado de rellenar los alumnos de la asignatura.
3. Si la tabla de usuarios contiene al menos un usuario, se fijará como origen de contenidos del grid y se mostrarán los usuarios de la asignatura.
4. En caso de que no se encuentre ningún alumno para la asignatura, se mostrará el mensaje de error.
5. Se utilizará el objeto SPContext para obtener el objeto sitio actual, sobre los que se realizará la consulta.

De forma adicional, se crearán dos métodos que servirán para rellenar los alumnos de la asignatura, uno para alumnos o profesores, y otro para grupos de teoría y laboratorio.

En caso de los alumnos o profesores:

1. Se creará un objeto DataTable vacío, al que se aplicará el formato de celdas específico.
2. Se utilizará el objeto SPContext para obtener el sitio actual.
3. Se iterará dentro de los grupos del sitio actual, hasta encontrar el de alumnos o profesores, según el parámetro de entrada del mismo.

4. Se listarán los alumnos o profesores del grupo y se incluirá una nueva fila en el DataTable con los datos de cada miembro, realizando una consulta al servicio del User Profile para obtener el email o teléfono si existiesen.
5. Se devolverá la tabla formateada con todos los alumnos localizados, o una tabla vacía en caso de no encontrar ninguno.

En caso de los grupos de laboratorio, teoría o prácticas:

1. Se creará una lista de objetos Grupo vacío.
2. Se utilizará el objeto SPContext para obtener el sitio actual.
3. Se iterará dentro de los grupos del sitio actual, hasta encontrar el de prácticas, laboratorio o teoría, según el parámetro de entrada del mismo y se creará un objeto Grupo para cada uno.
4. Se asignará al atributo Numero, el número del grupo.
5. Se listarán los alumnos del grupo y se concatenará el nombre de cada uno de los miembros en el atributo Miembros o se asignará un mensaje de error en caso de no localizar miembros en dicho grupo.
6. Se añadirá el objeto Grupo a la lista de objetos Grupo.
7. Se devolverá la lista de objetos.

En el caso del dropdown de opciones:

1. Cada vez que el objeto seleccionado cambie, se producirá una actualización en la página basada en el nuevo valor que se obtenga.
2. En función de dicho valor, un switch será el encargado de mostrar u ocultar la tabla o los repeaters y de, utilizando los métodos anteriores y gestionando el parámetro de entradas de los mismos, establecer los orígenes de contenido de los elementos.
3. Al igual que en el flujo de carga normal de la página, si no se encuentran usuarios / grupos, se mostrará el mensaje de error, ocultando el resto de contenedores.

7.3.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y los sitios y subsitios asociados a la asignatura, deben estar correctamente creados y accesibles.
- Los sitios y subsitios de las asignaturas, deben contener un título y enlace que cumplan los estándares definidos.
- En caso de estar matriculado en una asignatura, el alumno deberá figurar como miembro del grupo de alumnos, correctamente creado siguiendo los estándares.
- En caso ser profesor de una asignatura, el usuario deberá figurar como miembro del grupo de profesores, correctamente creado siguiendo los estándares.
- En caso pertenecer a un grupo de teoría, laboratorio o prácticas de una asignatura, el alumno deberá figurar como miembro del grupo correspondiente, correctamente creado siguiendo los estándares.
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de un sitio que represente una asignatura.

El sistema deberá reaccionar con un comportamiento específico ante las siguientes **excepciones**:

- Imposibilidad de acceder al sitio o a sus grupos: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a un grupo específico o una propiedad necesaria dentro del mismo: Escribir la excepción en el log de SharePoint, omitir dicho sitio y continuar comprobando el resto de los sitios restantes.
- Imposibilidad de encontrar grupos que cumplan los criterios seleccionados: Mostrar al usuario el mensaje de error.

7.3.5. Batería de pruebas

Se definirán 4 casos de prueba para este webpart, en función de los grupos existentes en la asignatura.

- **No existen grupos de teoría, laboratorio o prácticas:** Cuando sea seleccionada la opción de grupos de teoría, laboratorio o prácticas y no existan grupos del tipo seleccionado, el webpart debe mostrar un mensaje indicando que no se han encontrado grupos. **Resultado OK**
- **No existen el grupo de alumnos o profesores:** Cuando sea seleccionada la opción de alumnos o profesores y no existan grupos del tipo seleccionado, el webpart debe mostrar un mensaje indicando que no se han encontrado grupos. **Resultado OK**
- **Existen grupos de teoría, laboratorio o prácticas:** Cuando sea seleccionada la opción de grupos de teoría, laboratorio o prácticas y existan grupos del tipo seleccionado, el webpart debe mostrar el listado de grupos y miembros de cada uno, de la opción seleccionada. **Resultado OK**
- **Existen el grupo de alumnos o profesores:** Cuando sea seleccionada la opción de alumnos o profesores y existan grupos del tipo seleccionado, el webpart debe mostrar una tabla con los miembros seleccionados, mostrando su correo electrónico y teléfono si existe en el perfil de usuario. **Resultado OK**

7.4. Webpart Propiedades entrega

El objetivo de este webpart es mostrar, para la entrega actual, una tabla de propiedades con aquellas que se consideran más importantes dentro de la propia entrega.

7.4.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos:

- El listado mostrará al usuario, mediante una tabla, las propiedades de la entrega sobre la que está actualmente navegando.
- El encabezado de la tabla mostrará las siguientes propiedades:
 - Título de la entrega
 - Descripción de la entrega
- El contenido de la tabla mostrará las siguientes propiedades:
 - Estado de la entrega
 - Calificación
 - Fecha de entrega
 - Tiempo restante
 - Miembros
 - Última modificación: Fecha y autor
 - Comentarios de la entrega
- Los elementos de la tabla se mostrarán horizontalmente en pares propiedad-valor.

7.4.2. Funcionalidad utilizada

La funcionalidad de este webpart está basada en un tipo de contenido especial de SharePoint, el Document Set. De forma estándar, el Document Set dispone de un webpart encargado de mostrar las propiedades similar a como lo hará el webpart que se desarrollará. La decisión de implementar un webpart a medida radica en que la forma de presentar los datos que existe por defecto, no se ajusta al formato esperado.

Para la obtención de los datos, únicamente será necesario consultar al propio elemento de la biblioteca por sus propiedades. Es por ello que, teniendo en cuenta

que el nivel de seguridad en SharePoint únicamente es capaz de llegar a nivel de elemento y no de sus propiedades, si este webpart está ubicado dentro de un documento al que el usuario ha tenido acceso, también tendrá acceso a todas sus propiedades.

7.4.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Título entrega	Texto estático	ASP Label
Descripción entrega	Texto estático	ASP Label
Propiedades entrega	Tabla	Table HTML ASP Label / propiedad

Cuadro 7.4: Elementos web: Webpart Propiedades Entrega

El flujo de carga de la página será el siguiente:

1. Se llamará al método de carga de propiedades de la entrega.

El método de carga de propiedades de la entrega, generará el siguiente flujo de trabajo:

1. Se llamará al método encargado de obtener el objeto evaluable correspondiente.
2. Se llamará al método de obtención de la calificación correspondiente.
3. Se utilizará el objeto SPContext para conseguir la entrega actual.
4. Se rellenarán los datos de la tabla con los datos de los tres objetos anteriores.

El método de obtención del evaluable generará el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para recuperar el sitio actual.
2. Se realizará una consulta a la lista de evaluable para obtener el evaluable asociado a la entrega.
3. Se devolverá el objeto evaluable correspondiente si se encuentra.

El método de obtención de la calificación generará el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para recuperar el sitio actual.
2. Se realizará una consulta a la lista de calificaciones para obtener la calificación de la entrega.
3. Se devolverá el objeto calificación correspondiente si se encuentra.

7.4.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y los sitios y subsitios asociados a la asignatura, deben estar correctamente creados y accesibles.
- Los sitios y subsitios de las asignaturas, deben contener un título y enlace que cumplan los estándares definidos.
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de un evaluable de tipo entrega.
- El evaluable ha debido ser creado mediante el procedimiento definido para ello, lo que implica que debe tener un elemento asociado en la tabla de evaluables.

El sistema debe reaccionar con un comportamiento específico ante las siguientes excepciones:

- Imposibilidad de acceder al sitio o a sus grupos: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a las propiedades del elemento: Escribir la excepción en el log de SharePoint y mostrar la tabla vacía.
item Imposibilidad de acceder al evaluable asociado a la entrega: Escribir la excepción en el log de SharePoint y mostrar la tabla vacía.

7.4.5. Batería de pruebas

Debida la sencillez de este webpart, la única prueba a realizar sobre el mismo, es la de comprobar si muestra correctamente las propiedades al acceder a un documento ya que, tal y como está diseñado el sistema, si el usuario accede a dicho documento, el mismo tendrá propiedades para mostrar y el usuario tendrá permisos para mostrarlas correctamente. **Resultado OK**

7.5. Webpart Gestor de tareas

El objetivo de este webpart es permitir a un alumno o grupo de alumnos, el seguimiento de una entrega mediante un simple gestor de tareas asociado a la propia entrega. Dicho gestor permitirá la creación de tareas y su clasificación por estado.

7.5.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos:

- El webpart mostrará al usuario un panel con las tareas creadas por cualquiera de los miembros del grupo sobre dicha entrega.
- El panel de tareas organizará las mismas en cuatro columnas en función del estado de la tarea: To Do, DOING, BLOCKED o DONE.
- Las tareas se presentarán como una tarjeta con un color de fondo, en la que se incluirá el texto de la tarea.
- El color de fondo de las tareas estará determinado por el estado de las mismas:
 - TO DO: Amarillo
 - DOING: Azul
 - BLOCKED: Rojo
 - DONE: Verde
- El webpart permitirá a los usuarios añadir nuevas tareas.
- El webpart permitirá a los usuarios modificar el estado de las tareas existentes.
- El webpart permitirá a los usuarios eliminar tareas.

7.5.2. Funcionalidad utilizada

La funcionalidad de este webpart utiliza una lista SharePoint, que se rellena con un elemento por cada tarea creada en cada uno de los documentos. Dado que la lista almacenará todos los datos relativos a todas las entregas de todos los alumnos de la asignatura, será necesario establecer la relación tarea-documento de alguna forma.

SharePoint identifica unívocamente un elemento cualquiera dentro de la granja mediante un parámetro GUID. Dichos elementos pueden ser tanto elementos de lista, como los propios servidores de la granja. El GUID es asignado de forma aleatoria durante la creación del objeto y no se puede modificar (salvo que se realice mediante una consulta UPDATE en el SQL, anulando automáticamente la garantía

de Microsoft).

Utilizando este parámetro se puede realizar una asociación tarea-entrega que permita la creación, lectura, modificación y borrado de las mismas.

7.5.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Panel tareas	Panel de cards BootStrap	ASP Repeater
Desplegable tareas creadas	Desplegable	ASP DropDownList
Desplegable estado tareas	Desplegable	ASP DropDownList
Botón actualizar	Botón	ASP Button
Botón eliminar	Botón	ASP Button
Text box nueva tarea	Cuadro de texto	ASP Textbox
Desplegable nuevo estado	Desplegable	ASP DropDownList
Botón añadir	Botón	ASP Button

Cuadro 7.5: Elementos web: Gestor Tareas

El flujo de carga de la página será el siguiente:

1. Se llamará al método de carga de tareas para mostrar las tareas actuales de la entrega, siempre que la página se cargue por primera vez.

La funcionalidad de carga de tareas se implementa en una función independiente, ya que será llamada en múltiples ocasiones. Su flujo de trabajo es el siguiente:

1. Se creará una lista de tareas, que servirá como origen de contenido al repeater del panel de tareas, junto con una lista de strings que servirá como origen de contenidos del desplegable de tareas.
2. Se utilizará el objeto SPContext para obtener el elemento de lista actual, que será la entrega.
3. Se realizará una consulta a la lista de tareas, para obtener todas las tareas asociadas a la entrega actual.
4. Para cada una de las tareas, se creará un nuevo objeto Tarea.
5. Se rellenará la propiedad correspondiente al estado de la tarea, en el objeto Tarea creado, dejando el resto de propiedades vacías.
6. Se rellenará la lista de strings con el texto de la tarea como un nuevo elemento.

7. Si al terminar la iteración, las listas anteriormente mencionadas contienen al menos un elemento, se fijará como origen de contenidos de sus respectivos elementos y se mostrará el resultado por pantalla.

En caso de añadir una tarea:

1. Se utilizará el objeto SPContext para obtener el elemento de lista actual, que será la entrega.
2. Se obtendrá el valor escrito en el textbox como texto de la tarea a crear.
3. Se realizará una consulta a la lista de tareas, para obtener todas las tareas asociadas a la entrega actual y se comprobará si la tarea que se desea añadir ya existe.
4. En caso de que no exista, se creará un nuevo elemento en la lista de tareas, con el texto introducido en el textbox, el GUID de la entrega actual y el estado obtenido a partir del desplegable de nuevo estado.
5. Se llamará al método de carga de tareas.

En caso de modificar una tarea:

1. Se utilizará el objeto SPContext para obtener el elemento de lista actual, que será la entrega.
2. Se obtendrá el valor seleccionado en el desplegable de tareas creadas como texto de la tarea a modificar.
3. Se realizará una consulta a la lista de tareas, para obtener todas las tareas asociadas a la entrega actual y se obtendrá el elemento correspondiente a la tarea a modificar.
4. En caso de que exista, se modificará el elemento en la lista de tareas con el estado obtenido del desplegable de estado de las tareas.
5. Se llamará al método de carga de tareas.

En el caso de eliminar una tarea:

1. Se utilizará el objeto SPContext para obtener el elemento de lista actual, que será la entrega.
2. Se obtendrá el valor seleccionado en el desplegable de tareas creadas como texto de la tarea a eliminar.

3. Se realizará una consulta a la lista de tareas, para obtener todas las tareas asociadas a la entrega actual y se comprobará si la tarea que se desea eliminar existe.
4. En caso de que no exista, se eliminará elemento en la lista de tareas.
5. Se llamará al método de carga de tareas.

7.5.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y los sitios y subsitios asociados a la asignatura, deben estar correctamente creados y accesibles.
- La lista de tareas de la asignatura debe estar correctamente creada y accesible.
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de la vista de un evaluable de tipo Entrega.

El sistema debe reaccionar con un comportamiento específico ante las siguientes excepciones:

- Imposibilidad de acceder al sitio o a sus grupos: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a la lista de tareas o a la propia entrega: Escribir la excepción en el log de SharePoint y no realizar modificaciones.
- Imposibilidad de añadir una tarea: Escribir la excepción en el log de SharePoint y no realizar modificaciones.
- Imposibilidad de modificar una tarea: Escribir la excepción en el log de SharePoint y no realizar modificaciones.
- Imposibilidad de eliminar una tarea: Escribir la excepción en el log de SharePoint y no realizar modificaciones.

7.5.5. Batería de pruebas

Se definirán 8 casos de prueba para este webpart, en función de la acción a realizar y del número de tareas asociadas a la entrega.

- **No existen tareas durante la carga:** El webpart deberá mostrar un tablero vacío con la estructura del mismo. **Resultado OK**
- **Existen tareas durante la carga:** El webpart deberá mostrar el tablero de entregas con las entregas existentes clasificadas por estado y color dentro del mismo. Adicionalmente, se deberá rellenar el desplegable de actualizar entregas con una entrada por cada entrega. **Resultado OK**
- **Actualizar tarea sin ningún valor seleccionado en el desplegable:** El webpart no deberá realizar ninguna acción y procederá a la recarga de la página. **Resultado OK**
- **Actualizar tarea seleccionada:** El webpart deberá actualizar el estado de la tarea, si procede, según el valor del desplegable de estado y recargar la página. **Resultado OK**
- **Eliminar tarea sin ningún valor seleccionado en el desplegable:** El webpart no deberá realizar ninguna acción y procederá a la recarga de la página. **Resultado OK**
- **Eliminar tarea seleccionada:** El webpart deberá eliminar la tarea y recargar la página. **Resultado OK**
- **Añadir tarea sin ningún valor en el textbox:** El webpart no deberá realizar ninguna acción y procederá a la recarga de la página. **Resultado OK**
- **Añadir tarea con textbox relleno:** El webpart deberá crear una nueva tarea, con el estado correspondiente al obtenido en el desplegable de estado y recargar la página. **Resultado OK**

7.6. Webpart Calificar

El objetivo de este webpart es permitir a un profesor, añadir una calificación a un evaluable de un alumno o grupo.

7.6.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos:

- El webpart permitirá calificar un evaluable para un alumno o grupo dentro de la asignatura.
- El webpart permitirá seleccionar al profesor el evaluable a calificar y el usuario o grupo al que debe calificar.
- El webpart permitirá calificar a un único alumno o grupo a la vez.

7.6.2. Funcionalidad utilizada

Este webpart utiliza la API de Document Set de SharePoint. El Document Set es un tipo de contenido especial de SharePoint cuyo uso permite tratar un grupo de documentos como un único elemento de lista. En la práctica, un document set no es más que una carpeta que contiene documentos que, a su vez, tiene propiedades similares a las que puede tener un documento.

Aunque en esencia, un Document Set pueda ser tratado como una carpeta, la recomendación de Microsoft para evitar conflictos, es utilizar la clase DocumentSet.

De forma adicional se utilizarán consultas a listas y operaciones similares a las expuestas en los anteriores desarrollos para relacionar los documentos con evaluables y calificaciones.

7.6.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Panel selección evaluable	Desplegable	ASP Panel ASP DropDownList
Panel selección entrega	Texto Deplegable	ASP Panel HTML p ASP DropDownList
Panel calificar	Texto Cuadro de texto Botón	ASP Panel HTML p ASP TextBox ASP Button

Cuadro 7.6: Elementos web: Webpart Calificar

El flujo de carga de la página será el siguiente:

1. Se llamará al método de carga del primer desplegable en caso de que sea la carga inicial de la página.

La funcionalidad de carga del primer desplegable se implementará en una función independiente. Su flujo de trabajo es el siguiente:

1. Se creará un objeto DataTable que servirá como origen de contenidos para el desplegable de evaluables.
2. Se utilizará el objeto SPContext para obtener el sitio actual.
3. Se extraerá la lista de evaluables y entregas del sitio actual y se realizará una consulta para obtener todas las carpetas que estén en la raíz de la lista y que coincidan con el nombre de un evaluable, pues cada una representa una práctica.
4. Por cada carpeta encontrada, se rellenará una entrada del DataTable con el nombre de la carpeta como texto, y su URL como valor.
5. Se marcará como origen de contenidos del desplegable de evaluables, el DataTable obtenido.

Cuando se seleccione alguno de los elementos del desplegable de evaluables, se recogerá dicho evento provocando el siguiente flujo de trabajo:

1. Se creará un objeto DataTable que servirá como origen de contenidos para el desplegable de entregas.
2. Se utilizará el objeto SPContext para localizar el sitio actual.

3. Sobre el sitio actual, se localizará la carpeta correspondiente a la URL almacenada en el elemento seleccionado del desplegable.
4. Se localizarán todos los elementos cuyo tipo de contenido sea entrega, dentro de dicha carpeta.
5. Por cada elemento encontrado, se rellenará una entrada del DataTable con el nombre del elemento como texto, y su URL como valor.
6. Se marcará como origen de contenidos del desplegable de entregas, el DataTable obtenido.

Cuando se seleccione alguno de los elementos del desplegable de entregas, se recogerá dicho evento provocando el siguiente flujo de trabajo:

1. Se mostrará el panel de calificar.

Cuando se utilice el botón dentro del panel de calificar, se producirá un evento que generará el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para obtener el sitio actual.
2. Sobre el sitio actual, se localizará el document set correspondiente a la URL almacenada en el elemento seleccionado del desplegable.
3. Se obtiene el campo usuario o grupo asociado a la entrega.
4. Si el campo contiene un único usuario, se llama al método encargado de calificar entregas para ese usuario.
5. Si el campo contiene un grupo, se llama al método de calificar entregas para cada uno de los usuarios del grupo.

El método encargado de calificar a un usuario, genera el siguiente flujo de trabajo:

- Se utilizará el objeto SPContext para obtener el sitio actual.
- Sobre el sitio actual, se localizará la lista de calificaciones de la asignatura.
- Se tratará de buscar si existe una calificación del usuario para el evaluable que se ha de calificar.
- En caso de encontrar dicha calificación, se actualizará el valor de la misma.
- En caso de no encontrar dicha calificación, se creará una calificación nueva para el usuario, en el evaluable.

7.6.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y los sitios y subsitios asociados a la asignatura, deben estar correctamente creados y accesibles.
- Las listas de entregas, calificaciones y evaluables deben estar correctamente creadas y accesibles.
- Las entregas deben estar correctamente creadas siguiendo el método definido para ello.
- El profesor tiene permisos de colaborador sobre toda la asignatura y sus contenidos.
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de un sitio asociado a una asignatura.

El sistema debe reaccionar con un comportamiento específico ante las siguientes excepciones:

- Imposibilidad de acceder al sitio o a sus grupos: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a la lista de entregas, calificaciones o evaluables: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de localizar prácticas creadas: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de localizar entregas creadas para una práctica: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de modificar o añadir una calificación: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.

7.6.5. Batería de pruebas

Se definirán 7 casos de prueba para este webpart, en función de los grupos existentes en la asignatura.

- **No existen prácticas definidas:** Cuando durante la carga de la página no se localice ninguna práctica creada en la biblioteca de entregas, no se deberá mostrar ningún valor en el primer desplegable y el resto deberán estar ocultos. **Resultado OK**
- **Existen prácticas definidas:** Cuando durante la carga de la página se localicen prácticas creadas en la biblioteca de entregas, se deberá rellenar el primer desplegable con una entrada por cada una. **Resultado OK**
- **No existen entregas asociadas a una práctica:** Cuando durante el evento de cambio en el primer desplegable, no se localice ninguna entrega asociada a la práctica creada en la carpeta correspondiente en la biblioteca de entregas, no se deberá mostrar ningún valor en el segundo desplegable y el resto deberán estar ocultos. **Resultado OK**
- **Existen entregas asociadas a una práctica:** Cuando durante el evento de cambio en el primer desplegable, se localice alguna entrega asociada a la práctica creada en la carpeta correspondiente en la biblioteca de entregas, se deberá rellenar el segundo desplegable con una entrada por cada una. **Resultado OK**
- **Se selecciona una entrega en el desplegable:** Se deberá mostrar el panel de calificar. **Resultado OK**
- **Se pulsa el botón calificar con el cuadro de texto vacío o incorrecto:** Se deberá mostrar un mensaje de error a la derecha del cuadro de texto, que impida avanzar. **Resultado OK**
- **Se pulsa el botón calificar con un valor correcto:** Se deberá crear, para cada usuario, un elemento calificación con el valor introducido. Si ya existe una calificación con el mismo código y alumno, únicamente se deberá modificar su valor. Se deberá mostrar un mensaje que indique que la operación se ha realizado con éxito. **Resultado OK**

7.7. Webpart Creación Entregas

El objetivo de este webpart es automatizar la tarea de creación de una práctica. Este proceso implica el registro de la misma en la lista de evaluables, la creación de una carpeta contenedora en la biblioteca de entregas y la creación de las correspondientes entregas dentro de dicha carpeta, para cada usuario o grupo de prácticas.

7.7.1. Requisitos funcionales

El webpart deberá cumplir con los siguientes requisitos:

- El webpart permitirá la creación de una práctica.
- El evaluable deberá ser creado con los siguientes datos:
 - Nombre: Nombre del evaluable.
 - Descripción: Descripción del evaluable.
 - Ponderación: Ponderación sobre la nota global de la asignatura, expresado en tanto por 1.
 - Fecha: Fecha del evaluable.
 - Alumnos: Alumno o grupo asociado a la práctica.
- La creación de un evaluable implica la creación de una carpeta con el mismo nombre del evaluable dentro de la biblioteca de entregas.
- La creación de un evaluable implica la creación de una entrega por usuario o grupo dentro de la carpeta anteriormente creada, con un nombre formado a partir del loginname de cada usuario del grupo.
- El usuario o los usuarios del grupo asociados a una entrega deben obtener permisos de edición únicamente sobre su documento.
- Los usuarios no asociados a una práctica no deben poder ver el resto de prácticas creadas en la asignatura.

7.7.2. Funcionalidad utilizada

Este webpart utiliza la API de Document Set de SharePoint ya expuesta en el desarrollo anterior. En el caso de este webpart, será necesario crear un Document Set desde cero, utilizando una tabla Hash para proporcionar las propiedades en lugar del tradicional sistema de propiedades de elemento. Se utilizará la clase Document Set y se observará cómo de ella se puede obtener tanto el elemento carpeta como el propio elemento de lista de SharePoint.

De forma adicional y para evitar que otros usuarios puedan acceder a prácticas de las que no sean propietarios, se deberá romper la herencia de permisos de un elemento y asignar únicamente los permisos estrictamente necesarios. Esto es una práctica muy habitual que requerirá la ejecución de código con privilegios elevados.

7.7.3. Diseño técnico

La estructura html de la página, contendrá los siguientes elementos:

Elemento	Descripción	Elemento web
Texto nombre	Texto estático	HTML p
Textbox Nombre	Cuadro de texto	ASP Textbox
Texto descripción	Texto estático	HTML p
Textbox descripción	Cuadro de texto	ASP Textbox
Texto ponderación	Texto estático	HTML p
Textbox ponderación	Cuadro de texto	ASP Textbox
Texto fecha entrega	Texto estático	HTML p
Textbox fecha entrega	Control de fecha	SP DateTimeControl
Texto alumnos / grupos	Texto estático	HTML p
People picker	People picker	SP PeopleEditor

Cuadro 7.7: Elementos web: Webpart Calificar

El flujo de carga de la página será el siguiente:

1. Se llamará al método encargado de obtener un código único para identificar al nuevo evaluable.
2. Se llamará al método encargado de crear una carpeta en la biblioteca de entregas para almacenar el evaluable, con el mismo nombre que el propio evaluable.
3. Se llamará al método de creación de entregas con los datos del código y la carpeta obtenidos previamente.

El método de obtención del código de la entrega tendrá el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para obtener el sitio actual sobre el que se ejecuta el webpart.
2. Se realizará una consulta a la lista de evaluables para obtener el código más alto dentro de la misma.
3. Se devolverá el código anterior incrementado en 1, que será el código del nuevo evaluable.

La creación del elemento evaluable generará el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para obtener el sitio actual sobre el que se ejecuta el webpart.
2. Se creará un nuevo elemento evaluable con el código previamente obtenido y los datos recogidos en el formulario.

El método de creación de la carpeta de la entrega generará el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para obtener el sitio actual sobre el que se ejecuta el webpart.
2. Se creará una nueva carpeta en la lista de entregas con el mismo nombre que el evaluable.

El método de creación de entregas generará el siguiente flujo de trabajo:

1. Se evaluarán los valores resueltos por el people picker y, en función del tipo de entidad resuelta (usuario o grupo) se llamará al método de crear entregas para cada uno.

El método de creación de una entrega individual generará el siguiente flujo de trabajo:

1. Se utilizará el objeto SPContext para obtener el sitio actual sobre el que se ejecuta el webpart.
2. Se obtendrá la carpeta creada de la biblioteca de entregas.
3. Se creará un nuevo document set con los datos de la entrega, dentro de la carpeta.
4. Se romperá la herencia de permisos en el document set.
5. Se eliminarán todos los permisos en el elemento y se asignarán permisos de contribución sobre el mismo a todos los alumnos miembros de la entrega y al grupo de profesores de la asignatura.

7.7.4. Precondiciones y control de excepciones

El sistema deberá cumplir las siguientes precondiciones:

- La colección de sitios asociada a la carrera y los sitios y subsitios asociados a la asignatura, deben estar correctamente creados y accesibles.
- Las listas de entregas y evaluables deben estar correctamente creadas y accesibles.
- El profesor tiene permisos de colaborador sobre toda la asignatura y sus contenidos.
- El usuario administrador del pool de aplicaciones debe contar con los permisos estándar.
- El webpart debe ser utilizado dentro de un sitio asociado a una asignatura.

El sistema debe reaccionar con un comportamiento específico ante las siguientes excepciones:

- Imposibilidad de acceder al sitio o a sus grupos: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de acceder a la lista de entregas, o evaluables: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de crear una carpeta en la lista: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de localizarla carpeta creada en la lista: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Imposibilidad de crear una entrega dentro de la carpeta: Escribir la excepción en el log de SharePoint y mostrar al usuario el mensaje de error definido.
- Alguno de los campos del formulario, excepto el people picker, está vacío: Mostrar un mensaje de error al lado del campo incorrecto y no permitir la ejecución.
- El campo de ponderación no tiene un valor entre 0 y 1: Mostrar un mensaje de error al lado del campo incorrecto y no permitir la ejecución.

7.7.5. Batería de pruebas

Se definirán 3 casos de prueba para este webpart, en función de los grupos existentes en la asignatura.

- **Datos incompletos en el formulario de creación:** Se deberá mostrar el mensaje de error correspondiente y no permitir que continúe el flujo de ejecución. **Resultado OK**
- **El formulario se rellena con datos válidos:** Se deberá crear la carpeta correspondiente en la biblioteca de entregas, con los document set correspondientes. **Resultado OK**
- **El formulario se rellena sin ningún usuario:** Se deberá crear la carpeta contenedora de las prácticas sin ningún document set dentro de ella. **Resultado OK**

Capítulo 8

Conclusiones y trabajo futuro

En esta sección se analizarán, atendiendo a los objetivos marcados para el proyecto, los resultados del proyecto y las conclusiones obtenidas durante la realización del mismo. De forma adicional, se tratará una sección de trabajo futuro, donde además de hablar de aquellos puntos que no se han podido llevar a cabo durante el proyecto, se comentarán algunas de las funcionalidades que se podrían añadir al desarrollo construido.

8.1. Conclusiones

8.1.1. Conclusiones a nivel de proyecto

Acorde a los objetivos marcados para el proyecto, se han obtenido las siguientes conclusiones:

- El proyecto ha servido como prueba de concepto para comprender cómo se podría crear un sistema de características similares a Campus Virtual, utilizando un entorno SharePoint. A criterio personal, se ha demostrado que la herramienta puede adaptarse, con una calidad nada despreciable, para cubrir necesidades similares a las que Moodle da solución, de una forma más sencilla a nivel de desarrollo.
- La jerarquía de objetos es uno de los aspectos fundamentales a la hora de llevar a cabo la implementación de un sistema. Durante el proyecto, se ha trabajado con distintas alternativas y se ha podido comprobar las implicaciones que esto tenía a la hora de desarrollar. A medida que la jerarquía de objetos se acercaba a la versión definitiva, los desarrollos reducían el número de operaciones necesarias y realizaban tareas cada vez más simples, para obtener un mismo resultado. Sólo después de este proceso de creación y modificación de la estructura, se ha podido llegar a comprender la verdadera importancia que tiene esta fase que, habitualmente, pasa desapercibida en comparación con la fase de desarrollo.
- El uso de buenas prácticas en el desarrollo, es algo que se pasa por alto en muchos proyectos. A medida que el desarrollo ha ido avanzando, se ha comprobado que la modificación de código para adaptarlo a los cambios en la jerarquía, era mucho más sencilla en aquellos desarrollos que utilizaban las técnicas correctas. Un ejemplo concreto es el del gestor de tareas donde, en un principio, las tareas se almacenaban como un atributo de la propia entrega en texto plano y se trataban mediante métodos de la clase String. Aunque funcionaba correctamente, posteriormente se detectó una nueva necesidad que no era posible implementar siguiendo esta metodología, por lo que finalmente se tuvo que optar por almacenar las tareas en una lista adicional, tal y como recomendaba Microsoft desde el principio. En resumen, se ha llegado a la conclusión de que, aunque hacer las cosas bien tenga un coste inicial más elevado, cualquier modificación o problema posterior siempre será mucho más sencillo.

8.1.2. Conclusiones personales

Acorde a los objetivos marcados a nivel personal, se han obtenido las siguientes conclusiones:

- El proyecto ha servido para la ampliación y consolidación de los conocimientos sobre SharePoint. Desde un punto de vista teórico, ha permitido trabajar con un espectro mucho más amplio de funcionalidades, permitiendo un conocimiento del producto mucho más en profundidad. A nivel práctico, además de consolidar los conocimientos adquiridos, ha permitido poner en práctica conceptos teóricos nuevos y comprobar las verdaderas implicaciones de los mismos en el contexto de un proyecto concreto.
- A nivel de análisis, la necesidad de enfrentarse a preguntas y tomar decisiones, acertadas y equivocadas, han permitido comprender el verdadero significado e implicaciones del modelo de objetos de SharePoint que, hasta ahora, resultaba bastante difuso.
- A nivel de desarrollo, el proyecto ha proporcionado la perfecta excusa para la investigación de patrones y buenas prácticas, y entender algunos de los motivos que generan proyectos difíciles de mantener.

8.2. Trabajo futuro

Como trabajo futuro, se podría implementar un gran número de funcionalidades basadas en la plataforma existente y que combinasen esta con Office 365. En este apartado únicamente se tratará de exponer aquellos puntos fundamentales para conseguir un funcionamiento mínimo completo para el sistema. Para ello, los puntos pendientes son los siguientes:

- **Exámenes:** Por un lado, la creación de una solución a medida que permita registrar en una lista para cada asignatura, una serie de preguntas y sus correspondientes respuestas. Por otro lado, la creación de una solución que permita realizar exámenes a partir de dichas preguntas y automáticamente insertar las calificaciones en las listas correspondientes. Se descartó la posibilidad de realizar esta funcionalidad porque es demasiado a medida y el elevado coste de desarrollo no se veía recompensado de cara a explorar las posibilidades de SharePoint.
- **Panel de control para profesores:** La creación de un listado donde los profesores puedan visualizar los datos y estadísticas de la asignatura y de sus alumnos. Se descartó la posibilidad de implementar esta funcionalidad por razones similares al punto anterior.
- **Seminarios:** La creación de un tipo de contenido para los seminarios y una lista donde estos sean ubicados, junto con los webparts a medida que correspondan, tal y como se ha realizado para una entrega.
- **Alertas y notificaciones:** Tal y como está planteado el sistema, es posible crear alertas y notificaciones por correo electrónico para cada una de las listas o elementos de lista de SharePoint. No se ha desarrollado nada al respecto porque requiere la creación de un servidor de correo saliente.
- **Log:** La implementación de un log que registre las trazas en el propio log de SharePoint, era una tarea con una complicación excesiva y con muy poco beneficio de cara al objetivo del proyecto, por lo que se decidió dejar de lado. En su lugar, se identificó con un comentario aquellas zonas donde se debería llevar a cabo esos registros.

Capítulo 9

Bibliografía

SharePoint es una herramienta cuya documentación pública a nivel teórico, procedente de fuentes oficiales, no es tan extensa como otras tecnologías. Este hecho es debido a que Microsoft posee una amplia gama de certificaciones de pago en las que ofrece toda esa documentación a cambio de un coste. Gran parte de la información pública en este aspecto, está enfocada a cuestiones de marketing y otros fines comerciales, por lo que no es lo suficientemente precisa como para poder desarrollar un capítulo.

Para poder obtener información teórica acerca de los conceptos fundamentales de la herramienta, se ha utilizado gran parte de la información que posee la biblioteca de conocimientos de la empresa, donde se puede encontrar material de certificaciones, cursos oficiales y cursos internos. De forma adicional, se ha permitido el acceso a la documentación privada de proyectos de donde se ha extraído gran parte de la información. Debido al compromiso de confidencialidad, no es posible citar estos conocimientos dentro de la memoria.

Dadas las circunstancias tratadas en los párrafos anteriores, en este capítulo se expondrán las referencias bibliográficas ajenas a la empresa, utilizadas para el desarrollo del proyecto.

Listado de enlaces consultados a fecha 10 de julio de 2018:

- [1] *Bitacora Office 365*. URL: <https://www.bitacoraoffice365.com/tabla-periodica-de-office-365/>.
- [2] *Cifras Office 365*. URL: <https://www.muycomputerpro.com/movilidad-profesional/2017/05/10/office-365-100-millones/>.
- [3] *Cuadrante Gartner*. URL: <https://blogs.technet.microsoft.com/microsoftlatam/2017/10/19/microsoft-sharepoint-es-reconocido-como-lider-en-el-cuadrante-magico-de-gartner-para-plataforma-de-servicios-de-contenido/>.
- [4] Gustavo Vélez Duque y col. *SharePoint 2016 de principio a fin*. Krasis Consulting, S.L.; Edición: 1 (27 de junio de 2016), págs. 35-96 117-268. URL: <http://Vistapreviaen.es/scribd.com/doc/316848447/SharePoint-2016-de-Principio-a-Fin-VVAA>.
- [5] *Introducción a Sharepoint*. URL: <http://dotnetters.es/wp-content/uploads/2016/06/IntroSharePoint.pdf>.
- [6] Olga Londer y Penelope Coventry. *Microsoft SharePoint 2016 Step by Step*. Microsoft Press; 1 edition, págs. 53-74 75-85 103-144 233-313. URL: <https://ptgmedia.pearsoncmg.com/images/9780735697768/samplepages/9780735697768.pdf>.
- [7] *Microsoft Teams*. URL: <https://www.ti-m.com.mx/2017/03/14/lanzamiento-mundial-microsoft-teams/>.
- [8] *Microsoft Technet*. URL: <https://technet.microsoft.com>.
- [9] *Moodle*. URL: <https://es.wikipedia.org/wiki/Moodle>.
- [10] *MSDN*. URL: <https://msdn.microsoft.com>.
- [11] *Office365*. URL: <https://products.office.com/es-es/academic/compare-office-365-education-plans>.
- [12] Dusan Petkovic. *Microsoft SQL Server 2016: A Beginner's Guide*. McGraw-Hill Education; 6 edition (October 14, 2016).
- [13] *Powerapps*. URL: <https://powerapps.microsoft.com/es-es/>.
- [14] *Soluciones Sharepoint*. URL: <http://www.soluciones-sharepoint.com/2012/04/cual-es-la-explicacion-mas-corta-y.html>.
- [15] *Sway*. URL: <https://sway.com/>.
- [16] Catrinescu Vlad y Seward Trevor. *Deploying SharePoint 2016*. Apress, págs. 37-73 75-85.
- [17] *Wiki Sharepoint*. URL: https://es.wikipedia.org/wiki/Microsoft_SharePoint.

Anexos

Anexo I. Script de creación colecciones de sitio

En este anexo se detallará el código del script de la creación de colecciones de sitios.

```
param (
    $Alias = "INF" ,
    $Name = "INF_Ingénieur_a_informática" ,
    $Url = "http://campusvirtual.uva.es"
)

Add-PSSnapin "Microsoft.Sharepoint.PowerShell" -ErrorAction
    SilentlyContinue

$siteUrl = $Url + "/sites/" + $Alias
$template = Get-SPWebTemplate "STS#0"
$owner = "TFG\Administrator"

New-SPSite -Url $siteUrl -Language 1033 -Template $template
    -Name $Name -OwnerAlias $owner
```

Anexo II. Script de creación de sitios

En este anexo se detallará el código del script de la creación de sitios.

```
param (
    $Alias = "PGPI",
    $Name = "PGPI-Planificaci n_y_gesti n_de_plataformas
        _inform ticas",
    $SiteUrl = "http://campusvirtual.uva.es/sites/INF"
)

Add-PSSnapin "Microsoft.Sharepoint.PowerShell" -ErrorAction
    SilentlyContinue

$webUrl = $SiteUrl + "/" + $Alias
$template = Get-SPWebTemplate "STS#0"
$owner = "TFG\Administrator"

New-SPWeb -Url $webUrl -Language 1033 -Template $template -
    Name $Name -UniquePermissions

$web = Get-SPWeb $webUrl

$op = New-SPClaimsPrincipal $owner -IdentityType
    WindowsSamAccountName
$administrator = $web | Get-SPUser $op
$alumnos = "$($Alias)_Alumnos"
$profesores = "$($Alias)_Profesores"

$web.SiteGroups.Add($alumnos, $administrator, $administrator
    , "Grupo_de_alumnos_de_la_asignatura")
$web.SiteGroups.Add($profesores, $administrator,
    $administrator, "Grupo_de_profesores_de_la_asignatura")

$group = $web.SiteGroups[$alumnos]
$role = $web.RoleDefinitions["Read"]
$RoleAssignment = New-Object Microsoft.SharePoint.
    SPRoleAssignment($group)
$RoleAssignment.RoleDefinitionBindings.Add($role)
$web.RoleAssignments.Add($RoleAssignment)
$web.Update()
```

```
$group = $web.SiteGroups[$profesores]
$role = $web.RoleDefinitions["Contribute"]
$RoleAssignment = New-Object Microsoft.SharePoint.
    SPRoleAssignment($group)
$RoleAssignment.RoleDefinitionBindings.Add($role)
$web.RoleAssignments.Add($RoleAssignment)
$web.Update()

$web.Dispose()
```

Anexo III. Script de creación de columnas

En este anexo se detallará el código del script de la creación de columnas.

```
param (
    $WebURL = "http://campusvirtual.uva.es/sites/INF"
)

Add-PSSnapin "Microsoft.Sharepoint.PowerShell" -ErrorAction
    SilentlyContinue

$web = (Get-SPSite $WebURL).RootWeb

$web.Fields.Add("TFG_Alumno", "User", $true)
$web.Fields.Add("TFG_Codigo", "Number", $true)
$web.Fields.Add("TFG_Calificacion", "Number", $false)
$web.Fields.Add("TFG_Tipo", "Choice", $true)
$web.Fields.Add("TFG_Fecha", "DateTime", $true)
$web.Fields.Add("TFG_Ponderacion", "Number", $true)
$web.Fields.Add("TFG_Nombre", "Text", $true)
$web.Fields.Add("TFG_Descripcion", "Text", $false)
$web.Fields.Add("TFG_Id", "Text", $false)
$web.Fields.Add("TFG_EstadoEntrega", "Choice", $true)
$web.Fields.Add("TFG_Comentarios", "Text", $false)
$web.Fields.Add("TFG_Tarea", "Text", $false)
$web.Fields.Add("TFG_EstadoTarea", "Choice", $false)

$field = $web.Fields.GetField("TFG_Alumno")
$field.Title = "Alumno"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Codigo")
$field.Title = "Codigo"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Calificacion")
$field.Title = "Calificación"
$field.Group = "TFG_Columns"
$field.Update()
```

```

$field = $web.Fields.GetField("TFG_Tipo")
$field.Choices.Add("Ex men")
$field.Choices.Add("Laboratorio")
$field.Choices.Add("Entrega")
$field.Choices.Add("Seminario")
$field.Choices.Add("Otros")
$field.Title = "Tipo"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Fecha")
$field.Title = "Fecha_entrega"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Ponderacion")
$field.Title = "Ponderaci n"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Nombre")
$field.Title = "Nombre"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Descripcion")
$field.Title = "Descripci n"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Id")
$field.Title = "Unique_ID"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_EstadoEntrega")
$field.Choices.Add("Pendiente")
$field.Choices.Add("Fuera_de_plazo")
$field.Choices.Add("Entregado")
$field.Choices.Add("Calificado")
$field.Title = "Estado_de_la_entrega"
$field.Group = "TFG_Columns"
$field.Update()

$field = $web.Fields.GetField("TFG_Comentarios")
$field.Title = "Comentarios"

```

```
$field.Group = "TFG_Columns"  
$Field.Update()
```

```
$field = $web.Fields.GetField("TFG_Tarea")  
$field.Title = "Tarea"  
$field.Group = "TFG_Columns"  
$Field.Update()
```

```
$field = $web.Fields.GetField("TFG_EstadoTarea")  
$field.Choices.Add("To-do")  
$field.Choices.Add("Doing")  
$field.Choices.Add("Blocked")  
$field.Choices.Add("Done")  
$field.Title = "Estado_de_la_tarea"  
$field.Group = "TFG_Columns"  
$Field.Update()
```

```
$web.Dispose()
```

Anexo IV. Script de creación de tipos de contenido

En este anexo se detallará el código del script de la creación de los tipos de contenido.

```
param (
    $WebURL = "http://campusvirtual.uva.es/sites/INF"
)

Add-PSSnapin "Microsoft.Sharepoint.PowerShell" -ErrorAction
    SilentlyContinue

$web = (Get-SPSite $WebURL).RootWeb

$ctypeGroup = "TFG_Content_Types"

#-----#
# CALIFICACION
#-----#

$typeName = "Calificacion"

$parent = $web.AvailableContentTypes["System"]
$ctype = New-Object Microsoft.SharePoint.SPContentType(
    $parent, $web.ContentTypes, $typeName)
$web.ContentTypes.Add($ctype)
$ctype.Group = $ctypeGroup

$field = $web.Fields.GetField("TFG_Alumno")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()

$field = $web.Fields.GetField("TFG_Codigo")
$field.EnforceUniqueValues = $true
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()
```

```

$field = $web.Fields.GetField("TFG_Calificacion")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

```

```

#-----#
# EVALUABLE
#-----#

```

```

$typeName = "Evaluable"

```

```

$parent = $web.availablecontenttypes["System"]
$type = new-object Microsoft.SharePoint.SPContentType(
    $parent, $web.ContentTypes, $typeName)
$web.ContentTypes.Add($type)
$type.Group = $typeGroup

```

```

$field = $web.Fields.GetField("TFG_Codigo")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

```

```

$field = $web.Fields.GetField("TFG_Nombre")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

```

```

$field = $web.Fields.GetField("TFG_Descripcion")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

```

```

$field = $web.Fields.GetField("TFG_Tipo")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

```

```

$field = $web.Fields.GetField("TFG_Fecha")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)

```

```

$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()

$field = $web.Fields.GetField("TFG_Ponderacion")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()

#-----#
# TAREA
#-----#

$ctypeName = "Tarea"

$parent = $web.availablecontenttypes["System"]
$ctype = New-Object Microsoft.SharePoint.SPContentType(
    $parent, $web.ContentTypes, $ctypeName)
$web.ContentTypes.Add($ctype)
$ctype.Group = $ctypeGroup

$field = $web.Fields.GetField("TFG_EstadoTarea")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()

$field = $web.Fields.GetField("TFG_Tarea")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()

$field = $web.Fields.GetField("TFG_Codigo")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$ctype.FieldLinks.Add($fieldLink)
$ctype.Update()

#-----#
# ENTREGA
#-----#

$ctypeName = "Entrega"

```



```

$parent = $web.availablecontenttypes["Document_Set"]
$type = New-Object Microsoft.SharePoint.SPContentType(
    $parent, $web.ContentTypes, $typeName)
$web.ContentTypes.Add($type)
$type.Group = $typeGroup

$field = $web.Fields.GetField("TFG_EstadoEntrega")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

$field = $web.Fields.GetField("TFG_Comentarios")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

$field = $web.Fields.GetField("TFG_Alumno")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

$field = $web.Fields.GetField("TFG_Codigo")
$fieldLink = New-Object Microsoft.SharePoint.SPFieldLink(
    $field)
$type.FieldLinks.Add($fieldLink)
$type.Update()

$web.Dispose()

```

Anexo V. Script de creación de listas

En este anexo se detallará el código del script de la creación de listas.

```
    param (
        $WebUrl = "http://campusvirtual.uva.es/sites/INF/PGPI"
    )
$web = Get-SPWeb $WebUrl

#-----#
# LISTA TAREAS
#-----#

$listName = "Tareas"

$listTemplate = $web.ListTemplates["Custom_List"]
$web.Lists.Add($listName, $listName, $listTemplate)

$list = $web.Lists[$listName]
$list.ContentTypesEnabled = $true
$list.Update()

$typeName = "Tarea"
$type = $web.Site.RootWeb.ContentTypes[$typeName]
$list.ContentTypes | % {$_.Delete()}
$list.ContentTypes.Add($type)
$list.Update()

$view = $list.Views["All_Items"]
$view.ViewFields.DeleteAll()
$view.Update()

$list.Fields | ? { $_.InternalName -like "TFG_*" } | % {
    $view.ViewFields.Add($_.InternalName)}
$view.Update()

#-----#
# LISTA CALIFICACIONES
#-----#
```

```

$listName = "Calificaciones"

$listTemplate = $web.ListTemplates["Custom_List"]
$web.Lists.Add($listName, $listName, $listTemplate)

$list = $web.Lists[$listName]
$list.ContentTypesEnabled = $true
$list.Update()

$typeName = "Calificacion"
$type = $web.Site.RootWeb.ContentTypes[$typeName]
$list.ContentTypes | %{$_.Delete()}
$list.ContentTypes.Add($type)
$list.Update()

$view = $list.Views["All_Items"]
$view.ViewFields.DeleteAll()
$view.Update()

$list.Fields | ? { $_.InternalName -like "TFG_*" } | % {
    $view.ViewFields.Add($_.InternalName) }
$view.Update()

#-----#
# LISTA EVALUABLES
#-----#

$listName = "Evaluables"

$listTemplate = $web.ListTemplates["Custom_List"]
$web.Lists.Add($listName, $listName, $listTemplate)

$list = $web.Lists[$listName]
$list.ContentTypesEnabled = $true
$list.Update()

$typeName = "Evaluable"
$type = $web.Site.RootWeb.ContentTypes[$typeName]
$list.ContentTypes | %{$_.Delete()}
$list.ContentTypes.Add($type)
$list.Update()

$view = $list.Views["All_Items"]
$view.ViewFields.DeleteAll()
$view.Update()

```

```
$list.Fields | ? { $_.InternalName -like "TFG_*" } | %{  
    $view.ViewFields.Add($_.InternalName)}  
$view.Update()
```

```
#-----#  
# BIBLIOTECA EVALUABLES  
#-----#
```

```
$listName = "Materiales"
```

```
$listTemplate = $web.ListTemplates["Document_Library"]  
$web.Lists.Add($listName, $listName, $listTemplate)
```

```
#-----#  
# BIBLIOTECA EVALUABLES  
#-----#
```

```
$listName = "Entregas"
```

```
$listTemplate = $web.ListTemplates["Document_Library"]  
$web.Lists.Add($listName, $listName, $listTemplate)
```

```
$list = $web.Lists[$listName]  
$list.ContentTypesEnabled = $true  
$list.Update()
```

```
$typeName = "Entrega"  
$ctype = $web.Site.RootWeb.ContentTypes[$typeName]  
$list.ContentTypes.Add($ctype)  
$list.Update()
```

```
$web.Dispose()
```

Contenido del CD

En este anexo se hace referencia al contenido del CD entregable. La estructura del dispositivo se compone de:

- **Memoria del trabajo en formato PDF.**
- **Campus.Virtual**, carpeta que contiene el código fuente de las soluciones desarrolladas.
- **Scripts**, carpeta que contiene los diferentes scripts utilizados para la creación de los contenidos necesarios en la plataforma.