

# Cooperating We Approve, CWA

Albert Garriga Martínez

**Resumen–** Eventos, tareas, asignaturas, documentos, foros y horarios, todos estos elementos se encuentran en el día a día de cualquier estudiante y aún así no existe ninguna herramienta que ofrezca las funciones para administrar conjuntamente todos estos elementos. Aquí nace la idea del proyecto, CWA, un proyecto que pretende ofrecer al estudiante una herramienta de soporte al estudio completa, a través de la unión de los elementos anteriormente comentados. El formato de esta herramienta debe ser portable, multiplataforma y práctica para el usuario final, debe ser una herramienta en formato de aplicación web. Este documento explica la concepción, el desarrollo y los resultados del proyecto.

**Palabras clave** Informe, aplicación, Web, SharePoint, herramienta, estudiante, soporte.

**Abstract–** Events, task, subjects, documents, forums and schedules, all of that elements are present in every day of the student life and still is there no existence of any tool that brind that functionalities in orden to manage jointly all of that elements. Here it's born the idea of the project, CWA, a project that pretends to offer at student a complete study support tool, that join all the elements previously mentioned. The format of that tool, should be portable, multiplatform and usefull to the final user, so that tool have to be in web application format. This document, tells the conception, the development and the results of the project.

**Keywords–** Report, application, Web, SharePoint, tool, student, support

## 1 INTRODUCCIÓN

El desarrollo de este trabajo de final de grado nace ante la necesidad de una herramienta de soporte al estudiante a través de la unión de módulos de gestión: gestión de horarios, de asignaturas, de eventos y de recursos sociales en el ámbito universitario. El proyecto se describe como una herramienta online, cuya tipología es de Aplicación Web, que conforma la aplicación multiplataforma perfecta para el estudiante.

Para desarrollar este proyecto, se utilizan herramientas y lenguajes de programación que permiten que una vez finalizado el proyecto, este sea de fácil mantenimiento, tolerable a modificaciones y de fácil ampliación. Éste es el resultado de la programación modular que nos ofrece el framework Sharepoint[1].

Se podría decir que el propio nombre del proyecto, es también el lema de desarrollo del mismo, la conjunción de

- E-mail de contacte: albertgarrigam@gmail.com
- Menció realizada: Enginyeria del Software
- Treball tutoritzat per: Xavier Roca Marva (Ciències de la Computació, CVC)
- Curs 2016/17

las partes es el camino del éxito. El proyecto se desarrolla en perfecta conjunción entre los sistemas de gestión que lo forman, se utiliza la unión de dichos sistemas para ofrecer un conjunto completo de operaciones que realizadas en un sistema, retroalimentan a los demás sistemas.

Para comprender la importancia de la conjunción, se entenderá el proyecto como un sistema que ofrece:

- Gestión de Asignaturas: Creación, inscripción y desinscripción
- Gestión de Horarios: Mostrar los eventos y asignaturas en un horario de planificación.
- Gestión de Eventos: Crear y marcar como realizado un evento.
- Gestión de Documentación: Subir y eliminar documentos.
- Gestión de Recursos Sociales: Crear y comentar en el foro de la asignatura.

Y finalmente, se explica la idea de la conjunción de los sistemas a través de un simple ejemplo:

1. Un *alumno A* recién dado de alta en la aplicación, se inscribe en la asignatura *Metodología*

2. El *alumno A*, en la página principal visualiza un horario semanal con las clases de la asignatura *Metodología*.  
El *alumno A* también visualiza en el horario, el evento *Examen Metodología* que creo otro alumno y marco dicho evento como general.
3. El *alumno A* navega a la página de documentos y visualiza todos los documentos de la asignatura *Metodología* que los alumnos de la asignatura han decidido compartir.
4. Finalmente el *alumno A*, accede al foro que esta enlazado con la asignatura *Metodología* y observa un extenso e interesante debate sobre porque la metodología A es presuntamente mejor que la B.

Este es el resultado de la conjunción de los sistemas y de la cooperación de los usuarios de la aplicación, Cooperating We Approve es la mentalidad que se extiende des de la base algorítmica del sistema hasta el uso del mismo.

## 2 OBJETIVOS

El propósito principal del proyecto es el desarrollo de una Aplicación Web que conforme la herramienta completa de ayuda al estudiante. Esta nueva herramienta debe integrar las funcionalidades de gestión de horarios, gestión de eventos, gestión de asignaturas, gestión de documentos y un modulo de recursos sociales, el foro.

La división del proyecto en objetivos, nos permite agrupar los requisitos del sistema según el objetivo a satisfacer.

### 2.1. Gestión de horarios

La aplicación ha de mostrar un horario en formato calendario semanal, donde el usuario podrá visualizar que asignaturas o eventos tiene en cada momento, de forma simple.

#### 2.1.1. Requisitos funcionales de la gestión de horarios

**2.1.1.1. Mostrar horario** Como hemos comentado anteriormente, la aplicación busca todas las asignaturas en las que el usuario esta inscrito y los eventos del usuario o los generales y a través de estos datos, crea un horario en tiempo real para mostrar al usuario.

**2.1.1.2. Creación de eventos** El horario, ha de ser una herramienta dinámica, debe permitir de forma intuitiva añadir eventos en él. Se debe pues, a través de un simple doble click en un cuadro de tiempo del horario, poder crear o modificar un evento.

### 2.2. Gestión de asignaturas

Uno de los aspectos principales de la aplicación, es la gestión de las asignaturas sobre las que los estudiantes podrán inscribirse.

La información de las asignaturas es el elemento de unión de los demás módulos de gestión, puesto que todos los elementos estarán enlazados a una asignatura. De tal modo, al usuario diario (conocido como alumno) se le han de conceder permisos para inscribirse en asignaturas y a los usuarios

administradores se les ha de permitir crear dichas asignaturas, el objetivo de la división de funciones por niveles, ofrece una correcta gestión de la información de las asignaturas, evitando información incorrecta y redundancia.

#### 2.2.1. Requisitos funcionales de la gestión de horarios

**2.2.1.1. Inscribirse en una asignatura** La base de datos se alimenta de la información del conjunto de usuarios, una vez ya existen asignaturas en la base de datos, cualquier usuario podrá inscribirse en una o varias asignaturas. Estas asignaturas serán las que conformen el horario.

**2.2.1.2. Crear una asignatura** Dado que la aplicación, es una aplicación dinámica, se debe permitir a los usuarios Administradores añadir o actualizar las asignaturas de cada carrera y de cada universidad.

### 2.3. Gestión de eventos

Se conoce como evento, la entrega de un trabajo, la fecha de un examen o el cumpleaños de algún conocido, dicho esto podemos diferenciar dos tipos de eventos: los eventos comunes y los eventos propios de un usuario. La aplicación ha de mostrar los eventos en el horario general del usuario, de la misma forma que ha de permitir al usuario crear sus propios eventos o los que sean comunes para todo aquel usuario que está inscrito en una asignatura.

#### 2.3.1. Requisitos funcionales de la gestión de eventos

**2.3.1.1. Creación del evento** La aplicación ha de permitir al usuario crear un evento para una fecha acordada, siendo este evento propio, solo aparecerá en el horario del usuario que ha creado el evento. En cambio, si el evento es común, por ej: la fecha de un examen, este evento estará asociado a una asignatura y se mostrará a todos los usuarios inscritos en la asignatura.

**2.3.1.2. Eliminar un evento** La aplicación ha de ofrecer la función de eliminar un evento.

**2.3.1.3. Marcar un evento como realizado** La aplicación ha de permitir al usuario marcar un evento como realizado, al marcarlo verá dicho evento de color verde en el calendario, para mostrarle que dicho evento ya ha sido realizado.

**2.3.1.4. Capacidad de crear y visualizar eventos generales** Una característica importante de la aplicación, es la capacidad de crear dos tipos de eventos, los eventos propios del usuario (que solo visualizará él) y los eventos generales de una asignatura. Un usuario, debe poder crear un evento, marcarlo como de ámbito general y al estar éste enlazado a una asignatura, se mostrará a todos los usuarios que cursen la asignatura. La diferenciación de eventos, permite crear eventos de ámbito general y propio, donde los generales podrán ofrecer a usuarios información de eventos que desconocen.

## 2.4. Recursos sociales: Foro

La aplicación ha de ofrecer un foro sobre cada asignatura sobre la que el usuario está inscrito, dicho foro tiene el propósito de dar una herramienta de conocimiento compartido entre los usuarios, cuya finalidad principal es la de resolver dudas.

### 2.4.1. Requisitos funcionales del Foro

**2.4.1.1. Creación de un hilo** La aplicación ha de permitir al usuario crear un nuevo hilo sobre una asignatura, a fin de obtener respuestas de otros usuarios.

**2.4.1.2. Comentar en un hilo** La aplicación ha de ofrecer la capacidad a los usuarios de comentar en los hilos creados.

## 2.5. Gestión de documentación

La aplicación dispone de espacio de almacenamiento en la nube para los usuarios, de esta forma los usuarios podrán subir documentos en su espacio de almacenamiento y compartir los datos que el usuario decida compartir con otros usuarios.

### 2.5.1. Requisitos funcionales del almacenamiento de documentos

**2.5.1.1. Subir documentos** La aplicación ha de permitir al usuario subir documentos en su espacio de almacenamiento de documentos.

**2.5.1.2. Eliminar documentos** La aplicación ha de permitir al usuario eliminar documentos de su espacio de almacenamiento de documentos.

**2.5.1.3. Compartir documentos** La aplicación ha de conceder al usuario la posibilidad de compartir un documento con los demás integrantes de una asignatura, esto se realiza a través de enlazar un documento a una asignatura y marcándolo como general, de esta forma el documento será accesible para todos los integrantes de una asignatura.

## 2.6. Requisitos no funcionales

Todos los requisitos anteriores son requisitos funcionales y están organizados según el objetivo a realizar, en cambio los requisitos no funcionales de este proyecto son globales, puesto que hacen referencia al diseño de la aplicación web.

### 2.6.1. Diseño minimalista

El diseño de la aplicación ha de ser minimalista, es decir evitar el uso excesivo de colores e imágenes que resultan en un diseño sobrecargado, de dicha forma se tratará de utilizar la misma gama de colores a través de toda la aplicación.

### 2.6.2. Interfaz intuitiva

El diseño de la aplicación ha de permitir al usuario sin conocimiento previo, utilizar la aplicación sin necesidad de leer ningún manual de usuario.

### 2.6.3. Diseño responsive

El diseño de la aplicación ha de permitir al usuario utilizar la aplicación en las diferentes plataformas, siendo las principales PC y móvil, de tal forma se deberá utilizar un diseño responsive en cada aspecto de la aplicación.

## 3 ESTADO DEL ARTE

Para entender que ofrece la aplicación, a diferencia de otras existentes en el mercado, listaremos los módulos de la aplicación y los compararemos con otras aplicaciones.

1. Gestión de horarios: Horario de Clase App[8]
2. Gestión de eventos Google Calendar[6]
3. Gestión de documentos : Dropbox[5], Mega[22], Google Drive[7].
4. Recursos Sociales : Facebook[23].

Sobre la gestión de horarios, la aplicación CWA permite de modo simple crear horarios dinámicos, dependiendo de a que asignaturas esta inscrito el usuario. De tal forma, no ha de perder el tiempo diseñando un horario cada semestre, simplemente ha de inscribirse a las asignaturas que cursa a través de un simple click y el horario se autogenera.

En el ámbito de la gestión de eventos, la aplicación es novedosa gracias al uso de las asignaturas como base del evento, puesto que se puede asignar un evento a una asignatura. De la misma forma, se hace muy completa la gestión de eventos puesto que esta generalizada en la gestión de horarios, de tal modo se pueden visualizar las clases a las que el usuario ha de asistir conjuntamente con los eventos que ha de realizar, tanto si los ha creado el como si son eventos generales creados por otros usuarios.

Seguidamente, CWA utiliza la simplicidad a la hora de dar uso a los recursos sociales, sin necesidad de conocer ni contactar con los usuarios de una asignatura. A través del foro se puede preguntar y resolver dudas referentes a la asignatura, de tal forma, no es necesario conocer quien forma parte de la asignatura ni como contactar con dichas personas, simplemente se debe crear un hilo en el foro realizando la pregunta. Finalmente, la gestión de documentos permite agrupar los documentos por asignaturas y compartirlos con una simple opción con todos los usuarios de la asignatura, posiblemente interesaría compartir los documentos a personas específicas pero al hacerlo globalmente, se ofrece un sinfín de documentación a todos los integrantes de todos los integrantes, sin excluir a nadie.

Cada una de las aplicaciones mencionadas, ofrece un diverso conjunto de funcionalidades( p. ej, notificación de eventos, compartir documentos con un usuario en concreto) que la aplicación CWA no ofrece aún, pero el punto fuerte de CWA es que ofrece el conjunto completo y necesario de funciones para llevar a cabo las diferentes tareas de gestión anteriormente, en cambio las aplicaciones del mercado se especifican en una o varios módulos de gestión, no ofrecen el conjunto completo.

## 4 METODOLOGÍA

Este proyecto ha sido desarrollado a través de la metodología SCRUM[9], una metodología de desarrollo ágil,

a través de entregas de trabajo, llamadas sprints[2]. Los sprints, son las interacciones donde se desarrollan los trabajos que se asignaron a dicho sprint, por lo cual al completar cada sprint se obtiene un entregable del proyecto con las tareas del sprint desarrolladas en la entrega.

Se ha adaptado el ciclo de vida de la metodología SCRUM a las necesidades del proyecto, por lo cual el ciclo de vida de cada iteración, es decir de los sprints, es la siguiente:

- Listar las tareas a desarrollar en este sprint
- Priorizar las tareas a desarrollar y las horas estimadas de desarrollo
- Desarrollar y realizar pruebas conjuntamente
- Al acabar el sprint, valorar los problemas ocurridos y posibles soluciones.
- Si es necesario añadir nuevas tareas al listado de tareas.

El desarrollo del proyecto a través de la metodología SCRUM, nos brinda la oportunidad de cambiar los requisitos del proyecto y la planificación del mismo sin derivar en un coste elevado de re-planificación del proyecto.

Las herramientas que se utilizan para gestionar la metodología SCRUM son: Assana[3] y InstaGantt[4].

## 5 REPRESENTACIÓN DEL SISTEMA

En orden de realizar una representación entendedora de la estructura del sistema, a continuación se muestran dos diagramas.

El primer diagrama muestra la estructura modular del sistema donde se hace una breve descripción de las funciones que permite realizar cada modulo, véase la figura 1.

El segundo diagrama, en cambio muestra una representación de la estructura de la base de datos, la representación se encuentra en la figura 2.

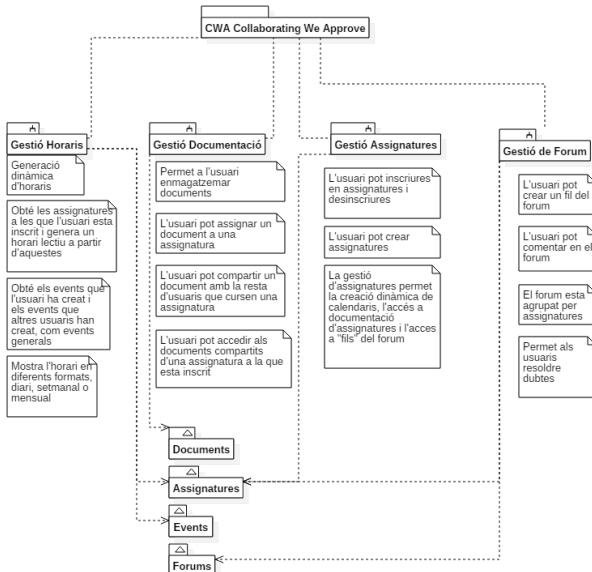


Fig. 1: Diagrama modular del sistema

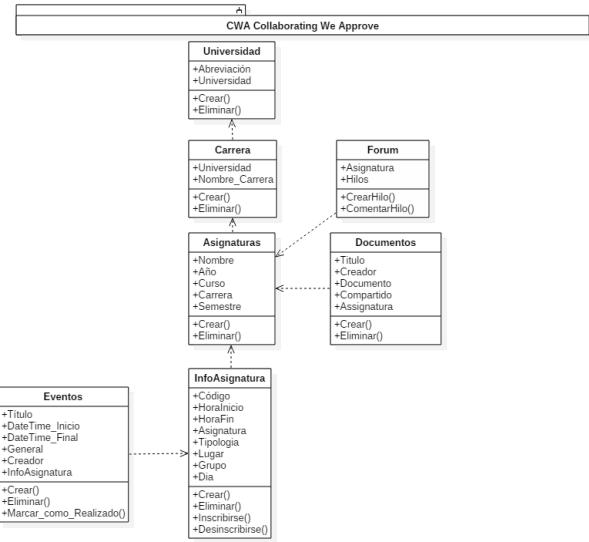


Fig. 2: Representación de la estructura de la base de datos

## 6 TECNOLOGÍAS UTILIZADAS

La principal tecnología utilizada, es Microsoft Sharepoint[1], en concreto la versión Sharepoint 2013 Server.

Sharepoint se define como una plataforma de colaboración empresarial, formada por productos y elementos software que incluyen, funciones de colaboración, módulos de administración de procesos y una plataforma de administración de documentos.

La selección de esta tecnología ha sido el resultado del principal requisito, desarrollar el mayor número de funcionalidades, en el menor tiempo posible. Dado este requisito, y mi afán de aprender nuevas tecnologías, he dado con esta tecnología, que destaca por su gran escalabilidad una vez puesta en marcha.

De tal forma, para este proyecto se decide utilizar la tecnología Sharepoint a fin de centrar los esfuerzos en los aspectos funcionales de la aplicación y no en los aspectos superfluos de esta (seguridad, integridad de los datos, acceso a la aplicación, configuración de cada módulo de la base de datos, maquetación de cada página de forma independiente, back-office, etc...) Para el desarrollo del proyecto distinguimos entre los frontales Back-End y Front-End.

### 6.1. Front-End

La maquetación de la infraestructura web se desarrolla a través de los lenguajes HTML y ASPX para la estructura y CSS para el diseño de la web. Una funcionalidad altamente práctica que ofrece SharePoint es el desarrollo de Layouts y Master Pages, que ofrecen un desarrollo de HTML y CSS en formato de plantillas, que extiende la modularidad de uso de los diseños HTML y CSS, se definen plantillas (Layouts) y páginas maestras (Master Pages) y se utilizan en las páginas que se desean, sin necesidad de repetir código. La comunicación con el servidor y la lógica de las páginas se construye a través del lenguaje Javascript.

## 6.2. Back-End

Los datos se almacenan en el servidor de SQL Server configurado para la perfecta colaboración entre los datos y la tecnología Sharepoint. De la misma forma, el tratamiento de datos por el servidor se desarrolla a través de las llamadas WebParts de Sharepoint, módulos desarrollados en lenguajes C#, que ofrecen una solución a problemas concretos.

## 7 DESARROLLO DEL PROYECTO

### 7.1. Fase 1 : Instalación y configuración de la Arquitectura: Servidor, SharePoint, SQL

Esta primera fase, corresponde a la instalación y configuración inicial del servidor.

En primer lugar, se realizó una partición del ordenador en el cual se iba a desarrollar el proyecto y se instaló Windows Server 2012 R2 en dicha partición.

Una vez instalado el entorno del sistema operativo, se procedió a instalar el programario necesario para el correcto funcionamiento del framework SharePoint. Seguidamente, se configuro el IIS[10] (Internet Information Services), herramienta a través de la cual podemos levantar la página web de la aplicación y del conjunto de granjas y servicios que esta aplicación utiliza.

El segundo programa a instalar y configurar fue, Microsoft SQL Server[11], el servidor de bases de datos que guarda la información introducida en la aplicación web y que gestiona los servicios de SharePoint, como el motor de búsqueda. Por último se procedió a instalar el framework de SharePoint y el programa que nos permite desarrollar sobre SharePoint, SharePoint Designer.

Para la instalación y configuración de SharePoint se han utilizado varias guías, la más influyente ha sido la citada a continuación, Instalación y Configuración SharePoint[12].

### 7.2. Fase 2: Configuración inicial de la Web Application

En esta fase se pretende ofrecer acceso a los usuarios y un primer esbozo de diseño en la aplicación, por lo cual en esta fase se obtiene la primera versión de la aplicación, el primer entregable marcado por el ciclo de vida de la metodología SCRUM.

Para obtener y otorgar acceso a la aplicación web utilizamos el directorio activo de Windows, por lo cual previamente es necesario configurarlo, para su configuración se ha utilizado una guia de configuración del AD,Active Directory[13].

El directorio activo de Windows ofrece la posibilidad de administrar los grupos de usuarios que tienen acceso a la aplicación y administrar eficientemente los permisos de los usuarios sobre la aplicación. Seguidamente, a través del IIS[10] se ha levantado el servicio Web que permitirá acceder a la aplicación web.

En segundo lugar, a través de la configuración general de Sharepoint creamos un Site Collection[14] de la aplicación, siguiendo los pasos marcados por la guía de creación de Site Collections[15].

### InfoAsignatura

⊕ nuevo elemento o modificar esta lista

Todos los elementos ... Buscar un elemento

Código	Horainicio	HoraFin	UIC_Asignatura	UIC_Asignatura:Curso	Tipología	Day	Lugar
103805	...	9	11 Fonaments d'Enginyeria	1	Teoría	Lunes	Aula Q2/1009
103801	...	11	12 Àlgebra	1	Problemas	Lunes	Aula Q2/1003
103801	...	12	13 Àlgebra	1	Problemas	Lunes	Aula Q2/1003
103806	...	13	15 Fonaments d'Informatàtica	1	Prácticas	Lunes	Lab.Integr.
103805	...	13	15 Fonaments d'Enginyeria	1	Prácticas	Lunes	Aula Q3/0013
102771	...	13	15 Electricitat i Electrònica	1	Prácticas	Lunes	
103805	...	13	15 Fonaments d'Informatàtica	1	Teoría	Lunes	Lab.Integr.
102771	...	9	11 Fonaments d'Enginyeria	1	Teoría	Martes	Aula Q2/1009
103806	...	11	13 Fonaments d'Informatàtica	1	Teoría	Martes	Aula Q2/1009
103805	...	13	15 Fonaments d'Enginyeria	1	Prácticas	Martes	Seminari A
103806	...	13	15 Fonaments d'Informatàtica	1	Prácticas	Martes	Lab.Integr.
101758	...	13	15 Fonaments d'Enginyeria	1	Prácticas	Martes	Aula Q3/0013
102771	...	13	15 Electricitat i Electrònica	1	Prácticas	Martes	
103806	...	13	15 Electricitat i Electrònica	1	Prácticas	Martes	Lab.Integr.
103805	...	13	15 Fonaments d'Enginyeria	1	Prácticas	Martes	Seminari A
102771	...	15	17 Electricitat i Electrònica	1	Prácticas	Martes	

Fig. 3: Esta figura muestra la lista a la que en este informe llamamos lista de elementos clase", donde los campos marcados en azul son campos relacionados con otras listas.

Finalmente, una vez configurado el IIS[10],el Active Directory y creado el Site Collection de la aplicación, se procede a crear y dar la estructura HTML y CSS a la página principal, para ello se utilizan las librerías Bootstrap[16] con el objetivo de obtener un diseño responsive[17], también se utilizan otras librerías de licencias gratuitas como Material Design Lite[18] .

### 7.3. Fase 3: Gestión de horarios.

Este es uno de los apartados más gruesos del proyecto en cuanto a tiempo de desarrollo, puesto que está formado por la creación de las listas que contienen la información de la universidad – carrera- asignatura a las que se apuntan los usuarios.

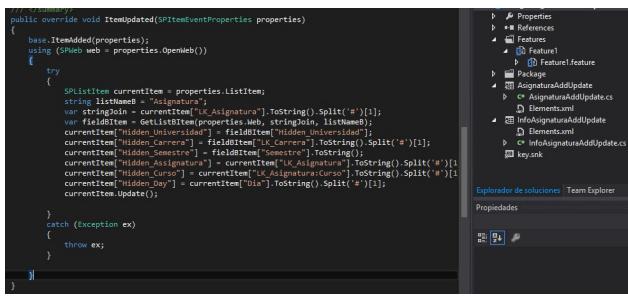
La creación de estas listas fue tarea fácil, se crearon Content Types[19] , que son estructuras de datos a los que darán forma a la estructura de una lista, por ejemplo se crea un Content Type Universidad que tiene los campos Nombre y Abreviación, ambos de tipo texto entonces se crea una lista y se le atribuye el tipo de contenido Universidad anteriormente creado, con lo cual al crear un elemento en esa lista tendrá los campos Nombre y Abreviación referentes a *Universidad*.

En esta fase se detectó un problema, y este fue que los campos que apuntan a otras listas no se pueden obtener desde el elemento enlazado final.

Por ejemplo: la *asignatura* a la que pertenece una *clase* no se puede obtener haciendo una consulta sobre el elemento *clase* que tiene una relación con el elemento *asignatura*.

Este problema era importante de resolver puesto que necesitamos obtener toda la información relacionada con un elemento *Clase*<sup>1</sup>, por ejemplo la Universidad con la cual esta relacionado el elemento clase.

<sup>1</sup>Cuando hablamos de un elemento clase, hablamos de una porción de una asignatura, por ejemplo : una asignatura A puede tener una clase de teoría, de prácticas o de problemas, de la misma forma que una asignatura puede tener diferentes grupos, con el elemento clase nos referimos concretamente a las porciones de asignatura que un usuario cursa, dónde siguiendo el ejemplo Alumna A cursa Teoría de Àlgebra en el grupo 41.



```

public override void ItemUpdated(SPItemEventProperties properties)
{
    base.ItemAdded(properties);
    using (SPSite web = properties.OpenWeb())
    {
        try
        {
            SPListItems currentitem = properties.ListItem;
            string listname = "Asignatura";
            var fieldTitle = web.Lists[listname].Asignatura;
            var fieldValue = fieldTitle.ToString();
            var listname2 = "Universidad";
            var fieldTitle2 = web.Lists[listname2].Universidad;
            currentitem["Hidden_Universidad"] = fieldValue;
            currentitem["Hidden_Carrera"] = fieldValue;
            currentitem["Hidden_Asignatura"] = fieldValue;
            currentitem["Hidden_Carrera"] = fieldValue;
            currentitem["Hidden_Asignatura"] = fieldValue;
            currentitem["Hidden_Carrera"] = fieldValue;
            currentitem["Hidden_Dia"] = fieldValue;
            currentitem.Update();
        }
        catch (Exception ex)
        {
            throw;
        }
    }
}

```

Fig. 4: Esta figura muestra una parte de la solución al problema, un event receiver de la creación de un elemento de tipo Clase.

Para dar solución a este error se han tenido que crear campos ocultos en la lista clase donde se guarda toda la información necesaria de una clase, por ejemplo ,la universidad a la que pertenece. Y para llenar la información se ha decidido crear una solución de Sharepoint en el servidor, que escucha los eventos de creación de elementos *Clase* y al crearlos rellena los campos ocultos.

El componente que realiza la escucha, se llama Event Receiver.[20] (Ejemplo de un Event Receiver.Fig 4)

### 7.3.1. Buscador personalizado de SharePoint y resultados de búsqueda

Es importante conocer la arquitectura de búsqueda de Sharepoint, Sharepoint ofrece un motor de búsqueda para diferentes contextos, en el caso del proyecto nos interesaba configurar el motor de búsqueda sobre las listas de la aplicación, para ello en la Administración Central de Sharepoint se creo un servicio de búsqueda y se le asigno el contexto de las listas de la aplicación.

En segundo lugar, es necesario crear las propiedades de las listas que se quiere que Sharepoint administre bajo las categorías que interesen para cada propiedad, por ejemplo en el caso del campo Universidad era necesario crear una propiedad administrada que fuese Refinable, a través de la cual podamos realizar agrupaciones.

Entonces, se indica a Sharepoint que realice rastreos incrementales sobre las listas y las propiedades asignadas, estos rastreos quedan guardados en documentos a través de los cuales la aplicación de búsqueda que creamos, siendo esta búsqueda sobre la lista Clase”, muestre los resultados pertinentes a la consulta añadida dentro de los documentos de rastreo.

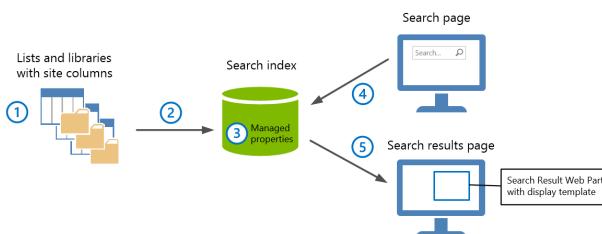


Fig. 5: Esquema de búsqueda de SharePoint

### 7.3.2. Display Templates

Una vez se obtiene toda la información necesaria en los campos, a través del buscador personalizado, donde el usuario sería capaz de visualizar todas las asignaturas existentes y inscribirse o borrarse de ellas.

Para crear el buscador, ha sido necesario crear cuatro Display Templates.

Un Display Template, es una plantilla para mostrar la información obtenida del servicio de búsqueda. Se crean en código HTML,CSS y Javascript para ofrecerles su lógica de funcionamiento.

1. Display template del cuadro de búsqueda
2. Display template del refinador Universidad
3. Display template del elemento *Clase*
4. Display template del conjunto de elementos *Clase*

Véase en la figura 6 los elementos que corresponden con los Display Templates

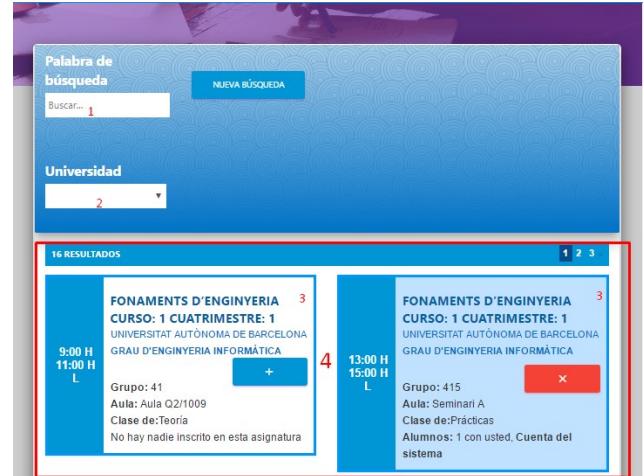


Fig. 6: Esta figura muestra un resultado de búsqueda, donde cada número de color rojo corresponde a un Display Template, enumerado anteriormente.

### 7.3.3. Calendario

Con la finalidad de crear el componente horarios a través del plugin dhtmlxScheduler[21], han sido necesarias muchas horas de lectura de documentación con el objetivo de sobreescribir las funciones de dicho plugin. Un ejemplo sería la eliminación de eventos del calendario, figura 8

A continuación se expone la lógica del sistema gestor de horarios. La lógica representada en la imagen es la siguiente:

1. Inicializar el plugin dhtmlxScheduler con las funciones, lógicas, estilos y idioma que hemos desarrollado para nuestro componente.
2. Realizar una consulta REST a la SharePoint API para obtener las asignaturas a las que el usuario está inscrito
3. Si hay asignaturas pasamos a 4, sino saltamos al punto 10.

```

scheduler.attachEvent("onBeforeEventDelete", function(id,e){
    //any custom logic here
    try{
        var idE = parseint(e.Id.split("-")[1]);
        var listName = "Task";
        var url_petition = _spPageContextInfo.webAbsoluteUrl;
        url_petition += "/api/web/lists/getbytitle('"+listName+"')/Items('"+idE+"')";
        if(e.userId != _spPageContextInfo.userId) throw 'P';
        jQuery.ajax({
            url: url_petition,
            type: "DELETE",
            headers: {
                "Accept": "application/json;odata=verbose",
                "X-RequestDigest": $("#__REQUESTDIGEST").val(),
                "IF-MATCH": "*"
            },
            success: function (data) {
                show_stack_bar_bottom('success',e.text,'Se ha eliminado este evento');
                return true;
            },
            error: function (data, errorCode, errorMessage) {
                show_stack_bar_bottom('error',e.text,'No tienes permisos para eliminar este evento');
                return false;
            }
        });
    }catch(ev){
        if(ev == "P"){
            show_stack_bar_bottom('error',e.text,'No tienes permisos para eliminar este evento');
            return false;
        }
        console.log("Error");
        show_stack_bar_bottom('error',e.text,'No puedes eliminar una asignatura');
        return false;
    }
    return true;
});

```

Fig. 7: Fragmento de código de uno de los eventos del plugin *dhtmlxScheduler*[21] que se ha tenido que implementar. Este fragmento de código tiene el propósito de implementar la lógica de eliminar los elementos que aparecen en el horario

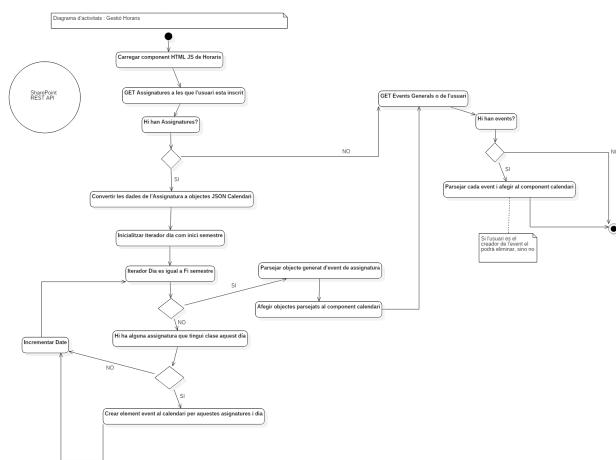


Fig. 8: Página principal, se puede observar el componente del calendario con los elementos de asignaturas en él.

4. A través de JavaScript, convertimos las asignaturas en objetos asignaturas y estos objetos asignaturas los convertimos a objetos calendario.
5. Inicializamos la fecha con el primer dia del semestre en el que esta la fecha actual.
6. Si la fecha que utilizamos como iterador no es igual a la fecha de fin de semestre pasamos al siguiente punto sino saltamos al punto 9.
7. Miramos el iterados, en que dia esta y si por ejemplo es lunes y una asignatura se hace el lunes, creamos un evento en el calendario.
8. Incrementamos fecha y vamos al punto 6.
9. Se parsean todos los eventos de asignatura creados a un objeto compatible con *dhtmlxScheduler* y finalmente se añaden dichos objetos al componente para mostrarlos.
10. Realizamos un proceso similar con los eventos.

Finalmente, se obtiene el siguiente componente de visualización de horarios.

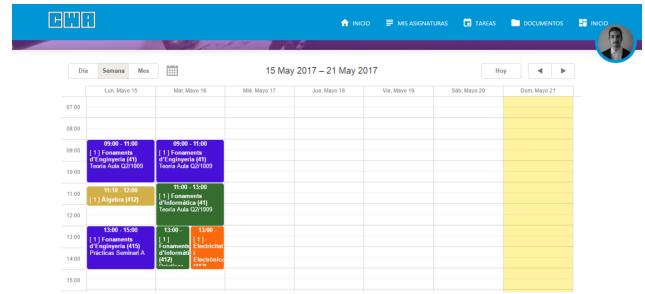


Fig. 9: En este diagrama de flujo, se puede observar una representación simplificada del funcionamiento del sistema gestor de horarios, donde se representa tanto la gestión de asignaturas como la gestión de eventos

## 7.4. Fase 4: Gestión de Tareas y Eventos.

Esta fase, anteriormente estaba contemplada como dos fases de desarrollo diferentes pero se ha decidido unir las listas y funcionalidades de tareas y eventos, obteniendo así una lista única de tareas y eventos mucho más entendedora de cara al usuario final.

### 7.4.1. Desarrollo en el servidor

Para el desarrollo de la funcionalidad de eventos, se ha creado una lista con el nombre Task que se utiliza para almacenar los eventos de cada usuario y los eventos que son comunes a todos los usuarios, eventos que denominamos Generales.

Del mismo modo que se realizó en la fase de asignaturas, en la gestión de tareas es necesario la creación de Event Receivers para poder crear la página de búsqueda de eventos, la figura 10 muestra el Event Receiver creado para la lista de Tareas.

### 7.4.2. Desarrollo en el cliente

Anteriormente, se ha explicado el desarrollo en el servidor que abarca la creación de la lista "Task" de eventos y la creación de los EventReceivers que permiten indexar los elementos de la columna Asignatura. Al indexar una columna, se pueden obtener resultados de búsqueda de esta.

#### 7.4.2.1. Display Templates

Una vez se obtiene toda la información necesaria en los campos, a través del buscador personalizado, donde el usuario sería capaz de visualizar todas las asignaturas existentes y inscribirse o borrarse de ellas.

Para crear el buscador, ha sido necesario crear cuatro Display Templates.

Un Display Template, es una plantilla para mostrar la información obtenida del servicio de búsqueda. Se crean en código HTML,CSS y Javascript para ofrecerles su lógica de funcionamiento.

1. Display template del cuadro de búsqueda

2. Display template del refinador Universidad
3. Display template del elemento "Tarea"
4. Display template del conjunto de elementos "Tareas"

#### 7.4.2.2. Tareas en el componente de Horarios

El último desarrollo sobre la fase de Eventos y tareas, es el de agrupar las asignaturas y las tareas en un único calendario, ofreciendo la funcionalidad de crear tareas desde el calendario general.

En primer lugar mostramos un ejemplo del calendario conjunto de asignaturas y tareas, desarrollado de forma similar al de Asignaturas, el componente que agrega elementos de tareas al horario, esta desarrollado en Javascript.

En segundo lugar, el desarrollo de la funcionalidad de crear tareas desde el calendario. Para ello, se han tenido que modificar las funciones del plugin dhtmlxScheduler que gestionan la creación, eliminación y actualización de tareas.

### 7.5. Fase 5: Recursos sociales "Forum".

Como se muestra en la figura 1, el modulo de foro esta relacionado con las asignaturas de cada carrera, de tal forma y con el objetivo de facilitar el uso de la aplicación, en la creación de una asignatura se hace de forma dinámica la creación del foro relacionado a esta.

Para ello, ha sido necesario un desarrollo en los receptores de eventos (Event Receiver), de las asignaturas y a su vez de la creación de la página donde se mostrarán y se permitirá crear y responder hilos del foro de una asignatura.

En la figura ?? se muestra la página de visualización de los hilos y de su creación. Seguidamente, en la figura 24 se muestra la página de un hilo del foro.

Para evitar la redundancia, solo se mostrarán los foros sobre las asignaturas a las que un usuario esta inscrito, vease la figura 14

```
public override void ItemAdded(SPItemEventProperties properties)
{
    base.ItemAdded(properties);
    using (SharePoint.Client.SPSite web = properties.OpenWeb())
    {
        try
        {
            SPListitem currentItem = properties.ListItem;
            string listItemB = "Infraestructuras";
            var fieldBItem = (SPFieldLookup) currentItem["TaskSubject"];
            var fieldBItem2 = GetListitem(properties.Web, stringJoin, listItemB);
            if (fieldBItem == null)
            {
                currentItem["Hidden_TaskAsignatura"] = "Otros";
                currentItem.Update();
            }
            else
            {
                currentItem["Hidden_TaskAsignatura"] = fieldBItem["k_Asignatura"].ToString().Split('#')[0] + " " + fieldBItem["Tipología"];
                currentItem["Hidden_TaskAsignatura"] = fieldBItem["k_Asignatura"].ToString().Split('#')[0] + " " + System.Convert.ToInt32(fieldBItem["Grupo"]) + " ";
                currentItem.Update();
            }
        }
        catch (Exception ex)
        {
            SPListitem currentItem = properties.ListItem;
            currentItem["Hidden_TaskAsignatura"] = "Otros";
            currentItem.Update();
            throw ex;
        }
    }
}
```

Fig. 10: Este fragmento muestra el código de la solución del servidor, en lenguaje C#, que resuelve el problema de la búsqueda sobre columnas que son de tipo búsqueda. Para ello se crea el Event Receiver, que se ejecuta cuando se crea o se actualiza un elemento de una lista de SharePoint. En este caso, al crear un elemento se lanza este código, que captura la columna Asignatura(una columna que apunta a un elemento de otra lista de SharePoint), y la copia en otra columna, Hidden\_Asignatura, donde esta última columna al no ser un campo de búsqueda sino uno de texto, se puede indexar en el motor de búsqueda de SharePoint

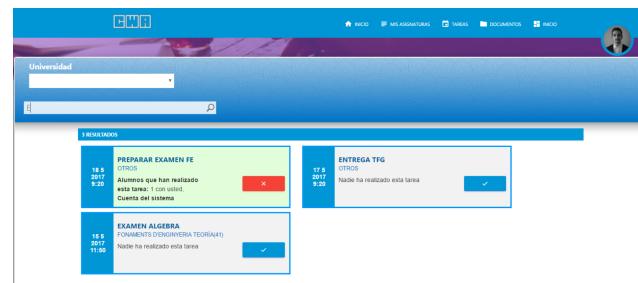


Fig. 11: Página de resultados de búsqueda de Tareas, donde el usuario visualiza las tareas que ha creado o que son generales y puede marcarlas como realizadas

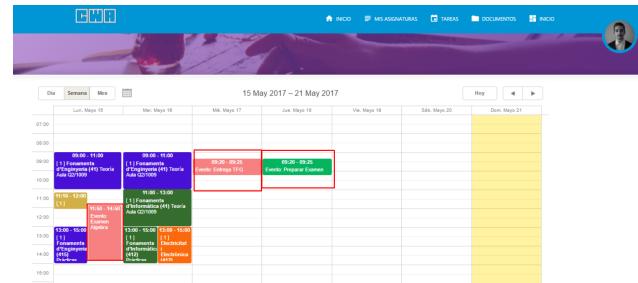


Fig. 12: Componente de tareas y asignaturas, las tareas están marcadas en un recuadro rojo. Se puede distinguir las tareas en colores rojo y verde según si han sido realizadas o no. Cabe mencionar que se muestran tanto las tareas creadas por el usuario, como las creadas por otros usuarios y marcadas como tareas generales.

### 7.6. Fase 6: Gestión de documentos

La gestión de documentos, fase que ha sido avanzada en la planificación.

El desarrollo de este componente abarca la creación de una biblioteca de documentos en el proyecto de Sharepoint y la creación de las funciones que cargan la librería en la página correspondiente.

#### 7.6.1. Librería de documentos

Se ha creado una librería de documentos en el servidor SharePoint, la librería de documentos funciona exactamente igual que una lista pero con la posibilidad de poder añadir un documento como campo, de tal forma se ha creado un Tipo de Contenido específico para la biblioteca, en el que se puede marcar un documento como general.

#### 7.6.2. Página de documentos y visualización

La visualización del conjunto de documentos ha sido desarrollada en Javascript en formato de tabla HTML, donde el usuario verá cómodamente el conjunto de documentos a los que tiene acceso. Se han añadido funciones para agregar y eliminar elementos. Solo se podrán eliminar los documentos que el usuario ha subido. Al subir un documento el usuario será capaz de asignar un documento a una asignatura y compartir con todos los integrantes de la asignatura dicho documento. De la misma forma, todos los documentos compartidos por los usuarios de una asignatura, serán accesibles por el usuario en cuestión, creando así un sistema de colaboración genérico.

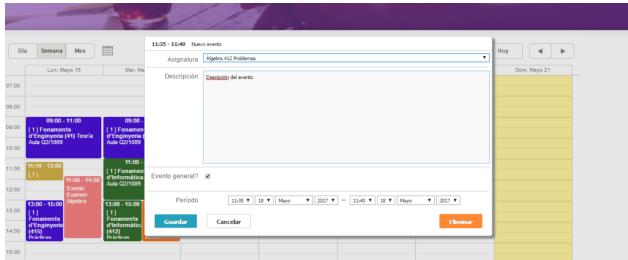


Fig. 13: Al hacer clic sobre un horario del calendario, se abre el desplegable de creación de tareas, donde se puede asignar una tarea a una asignatura y marcarla como tarea general o no.

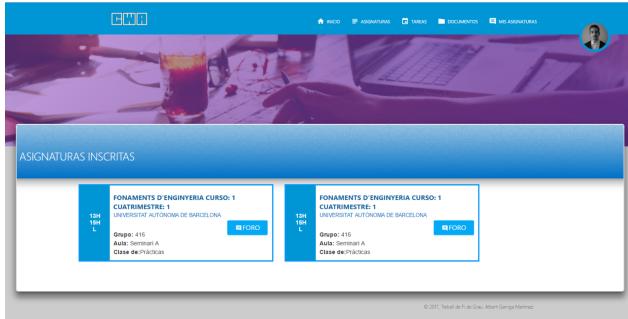


Fig. 14: Página de acceso a los foros, solo se muestran las asignaturas a las que un usuario está inscrito.

Otro aspecto interesante es la agrupación que se ha realizado de documentos, donde se agrupan por la asignatura a la que corresponden ofreciendo así un orden perfecto para el usuario.

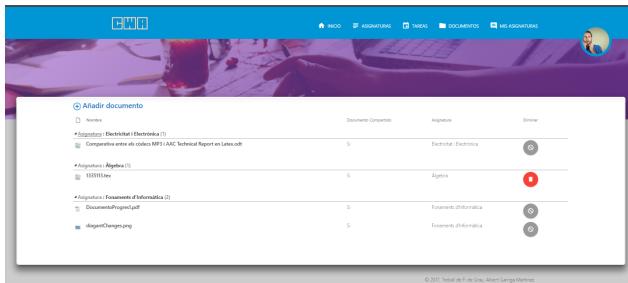


Fig. 15: Muestra de la página de documentos

## 8 RESULTADOS

El resultado del desarrollo del proyecto CWA, es una aplicación Web que se ofrece como una herramienta de soporte al estudiante, con el objetivo de dar soporte a las principales necesidades de un usuario.

Para mostrar los resultados de desarrollo se utilizarán escenarios agrupados por el módulo que los agrupa.

### 8.1. Gestión de asignaturas

Los usuarios con permisos de administración tienen la capacidad de crear Asignaturas, la creación de asignaturas se realiza de forma intuitiva como puede verse en la figura 16

Seguidamente, un usuario puede inscribirse y desinscribirse en dicha asignatura desde la página de asignaturas, véase en la figura 17 Finalmente, si el usuario lo desea en la misma página puede desinscribirse de la asignatura 18

## 8.2. Gestión de horarios

En la página principal, se muestra un calendario donde se pueden observar todas las asignaturas y todos los eventos (los creados por el usuario y los eventos generales creados por otros usuarios). En la figura 19 se puede observar el calendario de un usuario.

El calendario, también permite crear y modificar eventos de forma sencilla, como se puede ver en la figura 20

## 8.3. Gestión de eventos

Los eventos se pueden crear desde la página principal o desde la página de eventos, donde el usuario podrá marcar como realizados los eventos, véase en la figura 21

Una vez se marca el evento como realizado, pasa a visualizarse en color verde en el calendario con el objetivo de notificar visualmente al usuario que ese evento ya está realizado. En la figura 22 se ve el evento de Álgebra, anteriormente marcado como realizado.

## 8.4. Gestión de documentos

Esta sección demuestra ser muy interesante, puesto que al igual que los eventos, al marcar un documento como general éste documento será accesible para todos los usuarios de la asignatura y solo lo podrá eliminar el propietario del documento.

En la figura 23 se muestra la página de documentos. Todos los documentos se agrupan por la asignatura a la que pertenecen, ofreciendo así un orden inequívoco, los documentos que aparecen con la opción de eliminar son los que el usuario ha creado y los otros son los que otros usuarios han compartido.

## 8.5. Recursos sociales: Foro

Finalmente, cada asignatura tiene un foro asociado a ella, lo que permite mostrar a los usuarios el acceso a los foros que es de su interés. En la figura 24 se muestra un ejemplo de un hilo del foro.

## 9 CONCLUSIONES

Desde mi punto de vista, se ha llegando al fin de una herramienta unificada de ayuda al estudiante que al agrupar varios módulos de gestión se obtiene una aplicación bien relacionada, muy práctica y eficiente. En cuanto al desarrollo del proyecto, creo que se ha realizado a través de una correcta planificación y una buena separación del trabajo en fases lo que ha llevado el desarrollo por el camino correcto, a pesar de haber echo algún sobreesfuerzo en algunas fases, se han cumpliendo correctamente los tiempos planificados.

En mi opinión, si esta herramienta estuviese ofrecida por alguna universidad, podría ofrecer a los usuarios una buena

herramienta de soporte al estudio y al mismo tiempo la propia herramienta daría cabida a nuevas funciones y módulos que pudiesen parecer prácticos.

## AGRADECIMIENTOS

En primer lugar me gustaría dar la gracias a la Generalitat, por brindar becas de ayuda al estudiante, sin las cuales muchos estudiantes no podrían permitirse la oportunidad de una educación universitaria.

Seguidamente, dar las gracias a todos los profesores que a lo largo de los cursos me han enseñado la gracia de la informática.

Y por último, al tutor de este trabajo, el cuál me ha brindado todo el soporte que ha sido necesario y a su vez me ha permitido la opción de desarrollar un proyecto que me pareciese motivador.

## REFERENCIAS

- [1] Qué es Sharepoint?. URL: <https://dev.office.com/sharepoint/docs/spfx/sharepoint-framework-overview>
- [9] SCRUM, Metodología Ágil, Qué es SCRUM ? URL: <https://ca.wikipedia.org/wiki/Scrum>
- [2] Metodología SCRUM. Qué es un SPRINT? URL: <http://searchsoftwarequality.techtarget.com/definition/Scrum-sprint>
- [3] Assana, Herramienta de desarrollo Ágil URL: <https://app.asana.com/>
- [4] InstaGantt, Herramienta de diagramas de Gant relacionable con Asana, gratuita URL: <https://instagantt.com/>
- [5] Dropbox, URL <https://www.dropbox.com/>
- [6] Google Calendar, URL <https://www.google.com/calendar>
- [7] Google Drive, URL <https://www.google.com/drive/>
- [8] Horario de Clase App, URL <https://play.google.com/store/apps/details?id=com.newbitmobile.handytimetable&hl=es>
- [10] IIS, Internet Information Services. <https://www.iis.net/>
- [11] MICROSOFT SQL SERVER, Qué es Microsoft SQL Server? <http://searchsqlserver.techtarget.com/definition/SQL-Server>
- [12] GUÍA SHAREPOINT, Guía de configuración e instalación de SharePoint <http://www.enterpriseconsulting.es/blog/installacion-de-sharepoint-2013-en-windows-server-2012-r2/>
- [13] GUÍA ACTIVE DIRECTORY, AD, Guía de configuración del AD <https://blogs.technet.microsoft.com/meacoex/2013/08/04/step-by-step-active-directory-import-for-sharepoint/>
- [14] SITE COLLECTION, Qué es un Site Collection ? <https://technet.microsoft.com/en-us/library/cc262410.aspx>
- [15] SITE COLLECTION,CREACIÓN, Guía de creación de un Site Collection <https://technet.microsoft.com/en-us/library/cc263094.aspx>
- [16] BOOTSTRAP, Diseño Responsive <http://getbootstrap.com/>
- [17] RESPONSIVE, Qué es un diseño responsive? <https://responsivedesign.is/examples/>
- [18] MATERIAL DESIGN LITE, Librería de estilos, basada en Material Google <https://getmdl.io/templates/text-only/index.html>
- [19] CONTENT TYPE, Qué es un Content Type? <http://sharepointmaven.com/introduction-to-sharepoint-content-types/>
- [20] EVENT RECEIVER, Qué es y como crear un Event Receiver <https://msdn.microsoft.com/en-us/library/ee231563.aspx>
- [21] DHTMLXSCHEDULER, Plugin JavaScript para mostrar planificación, en formato de horario <https://dhtmlx.com/docs/products/dhtmlxscheduler/>
- [22] MEGA, Página Web para almacenar información <https://mega.nz/>
- [23] FACEBOOK, Plataforma de redes sociales. <https://www.facebook.com>

## APÉNDICE

### 1. Apéndice referente a Resultados

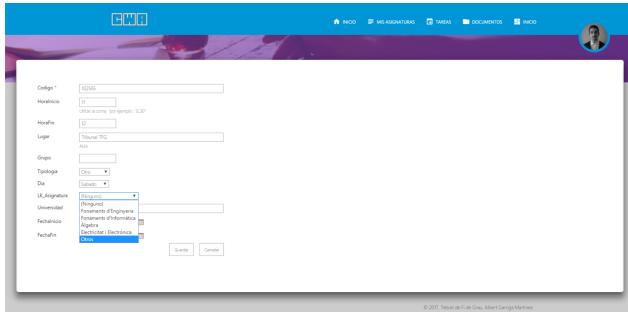


Fig. 16: Ejemplo de la creación de una Asignatura

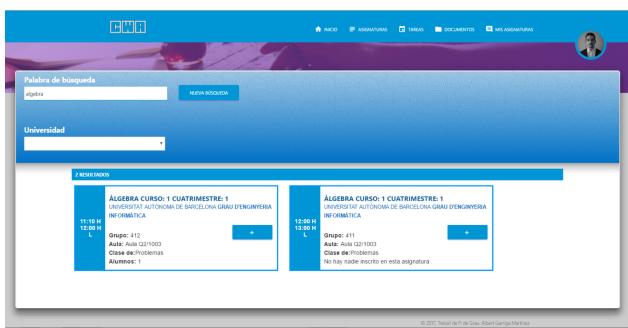


Fig. 17: Un usuario, desde la página de Asignaturas realiza una búsqueda de la asignatura *Álgebra* y con el botón + se inscribe a la asignatura.

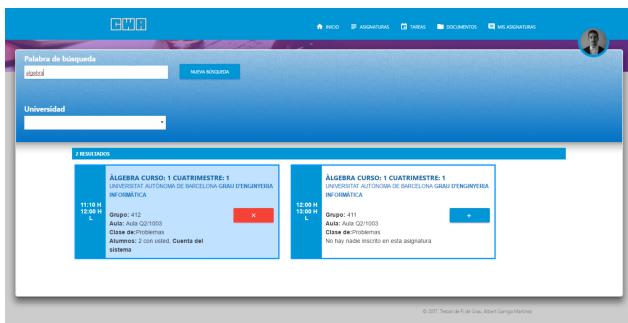


Fig. 18: Un usuario, desde la página de Asignaturas realiza una búsqueda de la asignatura *Álgebra* y con el botón X se desinscribe a la asignatura.

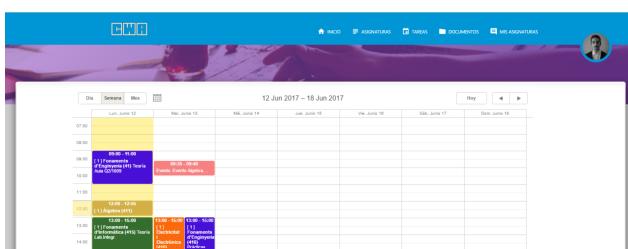


Fig. 19: Un usuario, desde la página principal visualiza las asignaturas y eventos en un formato semanal

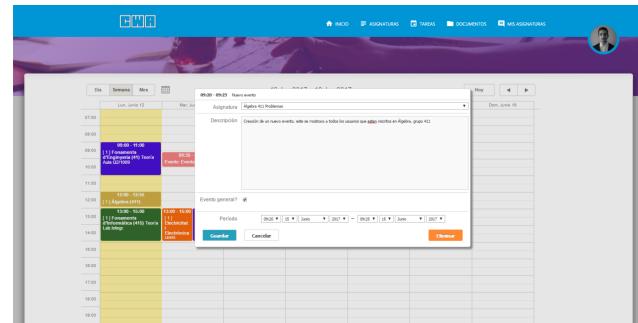


Fig. 20: Un usuario, hace click en un cuadrado del calendario y se abre un desplegable que le permite crear o modificar un evento. En este ejemplo el usuario crea un evento general, al ser general, el evento se mostrará a todos los usuarios de la asignatura.

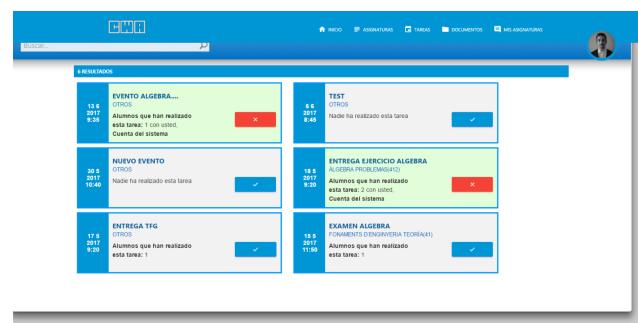


Fig. 21: Página de eventos, donde el usuario podrá crear y marcar como realizados los eventos

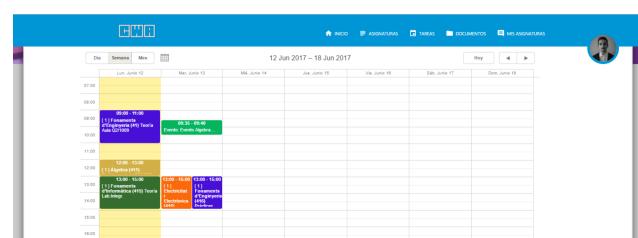


Fig. 22: Una vez marcado un evento como realizado, este pasa a visualizarse en color verde.

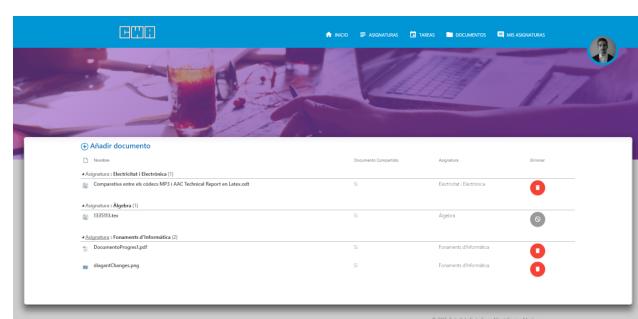


Fig. 23: Página de documentos

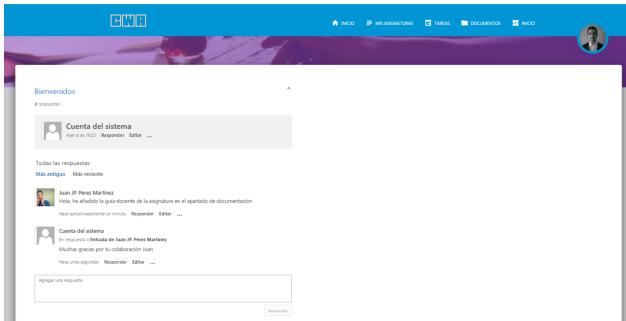


Fig. 24: Muestra de la página de un hilo del foro

## 2. Diagrama de planificación

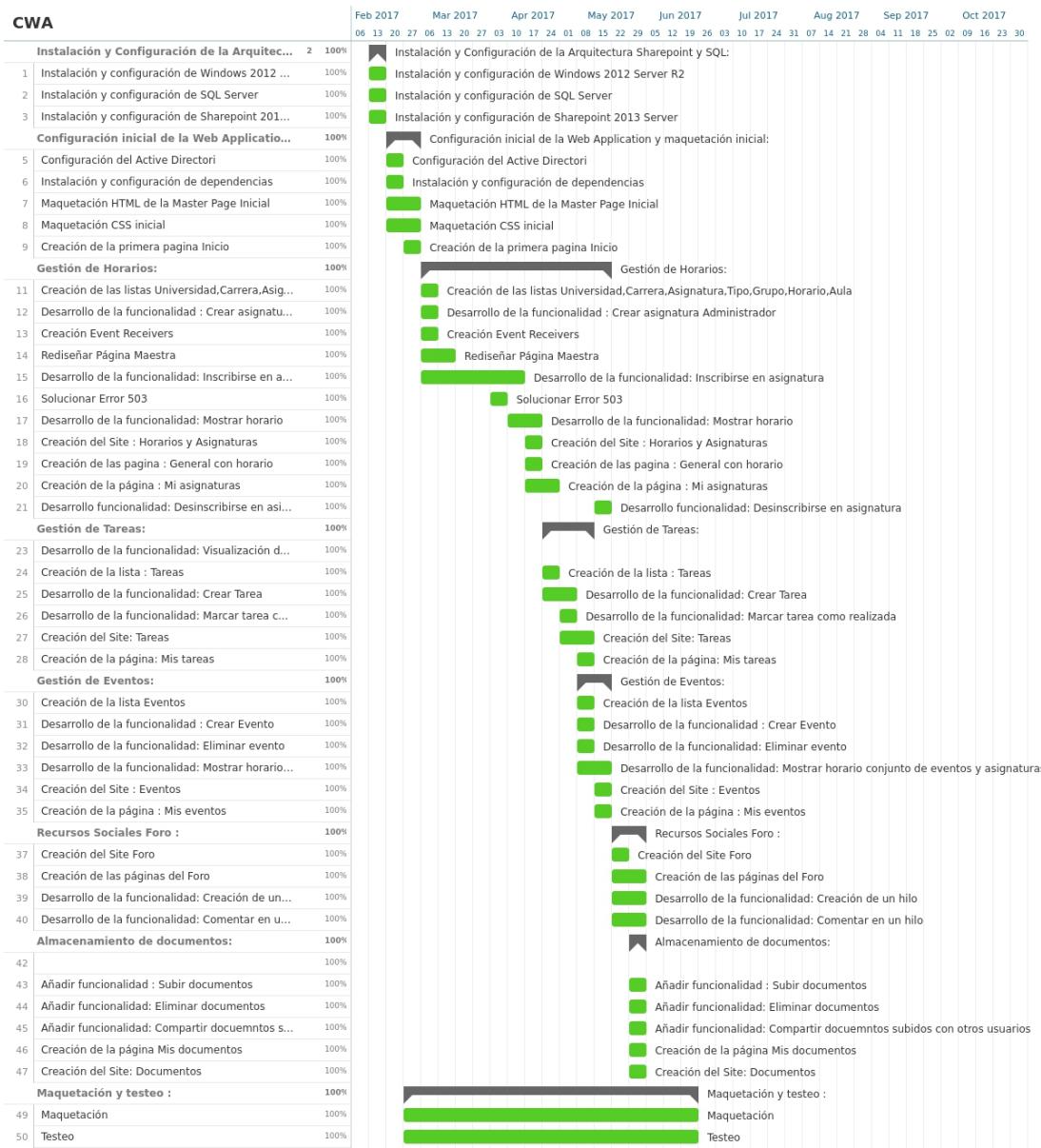


Fig. 25: Diagrama de Gantt de la planificación de las tareas de desarrollo del proyecto