



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática  
**DOCUMENTACIÓN TÉCNICA**



Presentado por Adrian Aguado  
en Universidad de Burgos — 9 de junio de 2017  
Tutor: Luis R.Izquierdo

---

# Índice general

---

Índice general	I
Índice de figuras	II
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	2
A.3. Estudio de viabilidad . . . . .	7
<b>Apéndice B Especificación de Requisitos</b>	<b>9</b>
B.1. Introducción . . . . .	9
B.2. Objetivos generales . . . . .	10
B.3. Catálogo de requisitos . . . . .	10
B.4. Especificación de requisitos . . . . .	11
<b>Apéndice C Especificación de diseño</b>	<b>16</b>
C.1. Introducción . . . . .	16
C.2. Diseño de datos . . . . .	16
<b>Apéndice D Documentación técnica de programación</b>	<b>17</b>
D.1. Introducción . . . . .	17
D.2. Requerimientos mínimos necesarios . . . . .	17
D.3. AngularCLI . . . . .	18
D.4. Manual del programador . . . . .	19
<b>Apéndice E Documentación de usuario</b>	<b>20</b>
E.1. Introducción . . . . .	20
E.2. Instalación . . . . .	20
E.3. Manual del usuario . . . . .	20

---

# Índice de figuras

---

A.1. Detalle sprint 0 . . . . .	2
A.2. Detalle sprint 1 . . . . .	3
A.3. Detalle sprint 2 . . . . .	4
A.4. Detalle sprint 3 . . . . .	4
A.5. Detalle sprint 4 . . . . .	5
A.6. Detalle sprint 5 . . . . .	5
A.7. Detalle sprint 6 . . . . .	6
A.8. Detalle sprint 7 . . . . .	6
B.1. Casos de uso. Fuente: Elaboración propia. . . . .	11

# Plan de Proyecto Software

---

## A.1. Introducción

En este capítulo se detalla la planificación del proyecto. Como gestor de tareas se comenzó utilizando *Trello+Github* pero más tarde se pasó a utilizar *Zenhub*, extensión de Google Chrome que permiten integrar los *boards* dentro del mismo repositorio de código alojado en *github*. Ya se han dado más detalles en la memoria del proyecto.

Se ha utilizado metodologías ágiles para el desarrollo del proyecto y de este modo, se ha realizado un desarrollo dividido en iteraciones. Terminada una iteración empezaba la siguiente y se agregaban a las tareas planeadas las que no habían sido completado de la iteración precedente. Las iteraciones del proyecto estaban pensadas para durar una diez días aproximadamente. No obstante, hay alguna excepción en la que la iteración duró más tiempo. También existe alguna demora entre algún sprint debido a que tenía demasiada carga de trabajo de las asignaturas, trabajaba o estaba de viaje.

La fase de planificación se puede dividir a su vez en:

- Planificación temporal.
- Estudio de viabilidad.

La primera parte me centro en la programación y desarrollo de la aplicación. Es decir elaboro un programa de tiempos con una serie de tareas a seguir para cumplimentar el proyecto.

La segunda parte se centra en el estudio de viabilidad. De la misma manera desde la segunda semana de marzo vengo realizando un plan de empresa con el programa Yuzz por lo que ello me va a facilitar el estudio de viabilidad de mi proyecto. Se desarrollará tanto la viabilidad legal como también la económica.

## A.2. Planificación temporal

Desde inicio del proyecto se planteó utilizar una metodología ágil como *Scrum* para la gestión del proyecto. Aunque no se ha seguido al 100 % la metodología al tratarse de un proyecto para la Universidad, sí que se ha aplicado en líneas generales una filosofía ágil y metódica.

A continuación se describen los diferentes *sprints* que se han realizado. Dentro de *github* cada *milestone* recibe el número del sprint asignado y dentro de cada uno de ellos existen diferentes tareas que describiré a continuación. A cada tarea le acompaña un número a la derecha el cuál es denominado *story point*, de alguna manera sirve para realizar una estimación de lo que te va llevar completar esa determinada tarea. En mi caso concreto el 1 resulta ser el más bajo lo cuál indicaría que no más de dos o tres horas con cada tarea y el 13 el más alto lo cuál significa días de trabajo.

### Sprint 0: 18/02/2017 - 28/02/2017

Tareas principales:

- Terminar formación.
- Aprender *Sonarqube*.
- Inicio Back-End.
- Inicio Decidir base de datos a emplear.

Completed Issues and Pull Requests	Story points
Terminar formación <span>courses</span> barterAPP #1 III New Issues ↑ Sprint 0	8
Aprender uso <a href="https://www.sonarqube.org/">https://www.sonarqube.org/</a> <span>research</span> <span>various</span> barterAPP #2 III New Issues ↑ Sprint 0	1
Decidir base de datos a usar. <span>research</span> <span>various</span> barterAPP #3 III New Issues ↑ Sprint 0	1
Comenzar desarrollo Back-End. <span>web app (Back-End)</span> barterAPP #5 III New Issues ↑ Sprint 0	8
Información sobre bases de datos NoSQL <span>courses</span> <span>research</span> barterAPP #6 III New Issues ↑ Sprint 0	6

Figura A.1: Detalle sprint 0

La primera vez que hice uso de *ZenHub* ya llevaba algún tiempo formándome, de ahí el primer *Issue* del Sprint 0. Esta primera toma de contacto fue para comenzar a desarrollar el Back-End de la aplicación, además se consultaron varias fuentes para decidir que tipo de base de datos emplear.

**Sprint 1: 18/02/2017 - 28/02/2017**

Tareas principales:

- Corrección de errores en Back-End.
- Desarrollo Back-end.
- Errores en base de datos.







Completed Issues and Pull Requests	Story points
 <b>Actualizar Mendeley con el nuevo contenido consultado</b> <span>documentation</span> <span>research</span> barterAPP #8 III New Issues ↑ Sprint 1	3
 <b>Corregir errores en base de datos</b> <span>bugs</span> <span>web app (Back-End)</span> barterAPP #9 III New Issues ↑ Sprint 1	5
 <b>Añadir identificación usuarios</b> <span>web app (Back-End)</span> barterAPP #10 III New Issues ↑ Sprint 1	8
 <b>Actualizar package.json</b> <span>web app (Back-End)</span> barterAPP #11 III New Issues ↑ Sprint 1	2
 <b>Corregir errores entre versiones</b> <span>bugs</span> <span>web app (Back-End)</span> barterAPP #12 III New Issues ↑ Sprint 1	5
 <b>Errores en package.json</b> <span>bugs</span> <span>web app (Back-End)</span> barterAPP #16 III New Issues ↑ Sprint 1	5

Figura A.2: Detalle sprint 1

El Sprint 1 sirvió para continuar con el Back-end de la aplicación sin duda fue un de las partes más complicadas al pelearme con bases de datos con conceptos nuevos por lo que tuve numerosos bugs a la hora de guardar los usuarios en la base de datos.

**Sprint 2: 15/03/2017 - 31/03/2017**

Tareas principales:

- Corrección de errores en Back-End.
- Inicio Front-End.
- Bug en base de datos.
- Comienzo a leer sobre la documentación.

Completed Issues and Pull Requests	Story points
<b>Landing-page</b> <span>web app (Front-End)</span> barterAPP #13 III New Issues ↗ Sprint 2	5
<b>Estructura visual</b> <span>web app (Front-End)</span> barterAPP #15 III New Issues ↗ Sprint 2	3
<b>MVC</b> <span>enhancement</span> <span>web app (Back-End)</span> <span>web app (Front-End)</span> barterAPP #17 III New Issues ↗ Sprint 2	13
<b>Bug sobre usuarios</b> <span>bugs</span> <span>web app (Back-End)</span> barterAPP #18 III New Issues ↗ Sprint 2	5
<b>Crear componentes en angular</b> <span>web app (Front-End)</span> barterAPP #19 III New Issues ↗ Sprint 2	8
<b>ReadTheDocs</b> <span>documentation</span> <span>enhancement</span> barterAPP #20 III New Issues ↗ Sprint 2	5

Figura A.3: Detalle sprint 2

El Sprint 2 fue el momento donde una vez tenía un back-end sólido debía trasladarlo a la parte del usuario por lo que comencé a realizar la parte del front-end.

### Sprint 3: 07/04/2017 - 15/04/2017

Tareas principales:

- Subir documentación.
- Elección del calendario.
- Corrección en componentes.

Completed Issues and Pull Requests	Story points
<b>Publicar landing page en github pages</b> <span>bugs</span> <span>enhancement</span> barterAPP #22 III New Issues ↗ Sprint 3	2
<b>Subir /docs</b> <span>documentation</span> barterAPP #23 III New Issues ↗ Sprint 3	1
<b>Login de usuarios</b> <span>web app (Back-End)</span> <span>web app (Front-End)</span> barterAPP #24 III New Issues ↗ Sprint 3	8
<b>Crear nuevos usuarios en la app.</b> <span>web app (Back-End)</span> <span>web app (Front-End)</span> barterAPP #25 III New Issues ↗ Sprint 3	6
<b>Nuevo Schema para calendarios.</b> <span>web app (Front-End)</span> barterAPP #26 III New Issues ↗ Sprint 3	3
<b>Estudiar que calendario incluir dentro de la app</b> <span>enhancement</span> <span>research</span> <span>web app (Front-End)</span> barterAPP #27 III New Issues ↗ Sprint 3	1
<b>Corregir el fallo del servicio de angular2</b> <span>bugs</span> <span>web app (Back-End)</span> barterAPP #28 III New Issues ↗ Sprint 3	3

Figura A.4: Detalle sprint 3

El Sprint 3 se centra en la parte del calendario sobre todo, además de algo de documentación y corregir los errores que he arrastrado del back-end.

**Sprint 4: 16/04/2017 - 22/04/2017**

Tareas principales:

- Actualizar a Angular CLI.
- Heroku y MLab
- Documentación





Completed Issues and Pull Requests	Story points
 <b>Actualizar a angular CLI</b> <span>enhancement</span> <span>web app (Back-End)</span> <span>web app (Front-End)</span> barterAPP #29 III New Issues ↑ Sprint 4	(5)
 <b>Crea base de datos online</b> <span>enhancement</span> <span>web app (Back-End)</span> barterAPP #30 III New Issues ↑ Sprint 4	(1)
 <b>Desplegar web en Heroku.</b> <span>enhancement</span> barterAPP #31 III New Issues ↑ Sprint 4	(2)
 <b>Mejorar documentación.</b> <span>documentation</span> <span>enhancement</span> barterAPP #32 III New Issues ↑ Sprint 4	(2)

Figura A.5: Detalle sprint 4

El Sprint 4 es más corto dado que requiere un menor tiempo en realizar las tareas.

**Sprint 5: 30/04/2017 - 07/05/2017**

Tareas principales:

- Angular CLI.
- Bugs





Completed Issues and Pull Requests	Story points
 <b>Continuar con la documentación</b> <span>documentation</span> barterAPP #33 III New Issues ↑ Sprint 5	(3)
 <b>Angular CLI vs Calendar</b> <span>bugs</span> <span>research</span> <span>web app (Front-End)</span> barterAPP #34 III New Issues ↑ Sprint 5	(8)
 <b>Error al guardar los elementos en la BD</b> <span>bugs</span> <span>web app (Front-End)</span> barterAPP #35 III New Issues ↑ Sprint 5	(8)
 <b>Angular CLI</b> <span>enhancement</span> <span>web app (Back-End)</span> <span>web app (Front-End)</span> barterAPP #36 III New Issues ↑ Sprint 5	(3)

Figura A.6: Detalle sprint 5

El Sprint 5 fue complicado debido a que cambiar a Angular CLI resulta más sencillo a la hora de desplegar en servidor pero hay que saber como funciona realmente los proyectos en Angular CLI



Sprint 6: 09/05/2017 - 16/05/2017

Tareas principales:

- Tarea1
- Tarea2

Completed Issues and Pull Requests	Story points
<div>Ⓢ Error en turnos largos <span>bugs</span> <span>web app (Back-End)</span> <span>web app (Front-End)</span></div> <div>barterAPP #37 III New Issues ↗ Sprint 6</div>	6
<div>Ⓢ Anexos de la documentación. <span>documentation</span></div> <div>barterAPP #38 III New Issues ↗ Sprint 6</div>	2
<div>Ⓢ Mejorar aspecto visual. <span>enhancement</span> <span>web app (Front-End)</span></div> <div>barterAPP #39 III New Issues ↗ Sprint 6</div>	13

Figura A.7: Detalle sprint 6

El Sprint 6

Sprint 7: 24/05/2017 - 31/05/2017

Tareas principales:

- Tarea1
- Tarea2

Completed Issues and Pull Requests	Story points
<div>Ⓢ Continuar documentación. <span>documentation</span></div> <div>barterAPP #40 III New Issues ↗ Sprint 7</div>	5
<div>Ⓢ Cambiar lógica de la app. <span>web app (Back-End)</span> <span>web app (Front-End)</span></div> <div>barterAPP #41 III New Issues ↗ Sprint 7</div>	13
<div>Ⓢ Aplicar test para ver la calidad del código. <span>tests</span></div> <div>barterAPP #42 III New Issues ↗ Sprint 7</div>	5
<div>Ⓢ Leer libro "Clean Code" <span>research</span> <span>various</span></div> <div>barterAPP #43 III New Issues ↗ Sprint 7</div>	Not estimated

Figura A.8: Detalle sprint 7

El Sprint 7

Sprint 8: 01/06/2017 - 10/06/2017

Tareas principales:

- Tarea1

- Tarea2

El Sprint 8

#### **Sprint 9: 10/06/2017 - 20/06/2017**

Tareas principales:

- Tarea1
- Tarea2

El Sprint 9

#### **Sprint 10: 20/06/2017 - 30/06/2017**

Tareas principales:

- Tarea1
- Tarea2

El Sprint 10

### **A.3. Estudio de viabilidad**

En esta sección se lleva a cabo un estudio para comprobar la viabilidad del proyecto realizado. Paralelamente al desarrollo de la aplicación, como ya se ha nombrado en la memoria anteriormente, el proyecto formó parte del programa YUZZ para jóvenes emprendedores en el que durante cinco meses realicé un plan de empresa completo. Se detalla por tanto en un documento que adjuntaré al proyecto un estudio de viabilidad exhaustivo y muy completo en el que se incluyen entre otras cosas: plan de marketing, plan de financiación, estudio de viabilidad o plan de puesta en marcha del negocio a cinco años vista.

Por lo tanto en esta sección voy a realizar un resumen del documento descrito en el párrafo anterior en el que como conclusión definitiva tendremos un boceto de lo que supondría transformar un proyecto fin de carrera y que pase a formar parte del mercado. Así mismo voy a intentar adaptarlo a las condiciones que se exigen en el proyecto dado que el plan de empresa completo es un estudio de viabilidad completo de aquí a cinco años por lo que resulta ser más extenso y detallado. Se intentará por tanto aquí realizar una estimación.

## Viabilidad económica

La viabilidad económica es la parte donde lograremos detectar si el proyecto es o no rentable económicamente hablando.

### Análisis de costes

Económica

**Coste de personal** Se considerará que el proyecto ha sido desarrollado en un periodo de cinco meses. Considerando que se ha trabajado unas 6 horas a día cada semana, y que el programador, que en este caso es una sola persona, ha percibido un sueldo de 13 e/hora, el coste del personal por lo tanto se resume en la siguiente tabla:

	Total
13 €/hora * 6 horas/día	78 €/día
78 €/día * 5 días/semana	390 €/semana
390 €/semana * 4 semanas/mes	1560 €/mes
<b>Coste total salario</b>	<b>7800 €/5 meses</b>

Tabla A.1: Tabla salarios.

**Coste de seguridad social** más información sobre el segundo item.

**Coste de software** más información sobre el segundo item.

**Coste de Hardware** más información sobre el segundo item.

## Viabilidad legal

La viabilidad legal se centra principalmente en el estudio de las licencias software utilizadas y en la licencia que se le va a ser asignada a las diferentes aplicaciones desarrolladas.

---

## Especificación de Requisitos

---

### B.1. Introducción

Este anexo recoge la especificación de requisitos que define el comportamiento del sistema desarrollado. El objetivo principal de la Especificación de Requisitos del Sistema (*ERS*) es servir como medio de comunicación entre clientes, usuarios, ingenieros de requisitos y desarrolladores.

La ERS es correcta si y sólo si todo requisito que figura en ella refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado. Se han seguido las recomendaciones del estándar IEEE 830 según la última versión del estándar IEEE 830. Las características deseables para una especificación de requisitos son:

1. No ambigua.
2. Completa
3. Verificable
4. Consistente
5. Clasificada
6. Modificable
7. Explorable
8. Utilizable durante las tareas de mantenimiento y uso

## B.2. Objetivos generales

Los objetivos generales que se perseguían con el proyecto han sido:

- Desarrollar una aplicación web que permita intercambiar turnos en función de unas restricciones.
- Desarrollar una aplicación híbrida que adapte la anterior funcionalidad a cualquier dispositivo.

## B.3. Catálogo de requisitos

A continuación, se enumeran los requisitos específicos :

En esta sección se especificarán los requisitos del sistema, diferenciando los requisitos funcionales, o sea los el comportamiento del sistema, de los requisitos no funcionales, que describen características de funcionamiento.

### Requisitos funcionales

**RF-1 Gestión de usuarios:** Los usuarios son la principal baza de la aplicación se deben poder gestionar correctamente.

- **RF-1.1:** Registrar usuarios
- **RF-1.2:** Logear usuarios
- **RF-1.3:** Deslogearse

**RF-2. Gestión de calendario:** Funciona como un calendario normal.

- **RF-2.1:** Añadir turnos.
- **RF-2.2:** Eliminar turnos.
- **RF-2.3:** Editar turnos.
- **RF-2.4:** Visualizar turnos.

**RF-3. Gestión de turnos:** Parte de intercambio de turnos.

- **RF-3.1:** Enviar petición.
- **RF-3.2:** Aceptar petición.
- **RF-3.3:** Rechazar petición.
- **RF-3.4:** Aceptar intercambio.
- **RF-3.5:** Rechazar intercambio.
- **RF-3.6:** Visualizar detalles.
  - **RF-3.6.1:** Visualizar turnos.

- **RF-3.6.2:** Visualizar pendientes.
- **RF-3.6.3:** Visualizar aceptados.
- **RF-3.6.4:** Visualizar rechazados.
- **RF-3.7:** Visualizar calendario.

**RF-4. Información:** Información de como usar la aplicación

### Requisitos no funcionales

**RNF-1 Seguridad** texto.

**RNF.2 Escalabilidad:** texto

**RNF.3 Eficiencia:** texto.

## B.4. Especificación de requisitos

En esta sección se explicará el diagrama de casos de uso y se desarrollará cada uno de los requisitos en función del esquema.

### Diagrama casos de uso

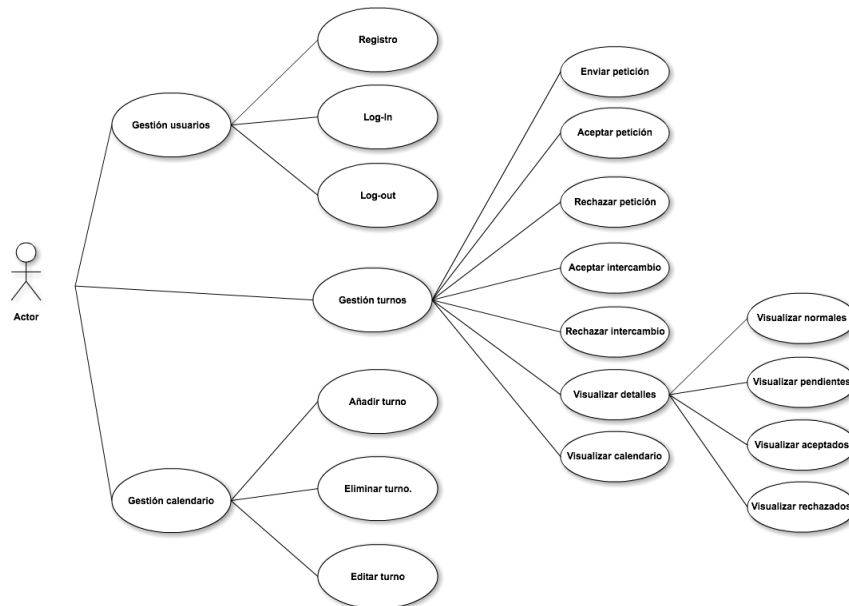


Figura B.1: Casos de uso. Fuente: Elaboración propia.

CU-01	Gestión usuarios
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-1.1, RF-1.2, RF-1.3
<b>Descripción</b>	Permite registrarse, logearse al usuario o salir
<b>Precondición</b>	La página esta cargada base de datos funcionando
<b>Acciones posibles</b>	1. El usuario se registra para 2. El usuario se logea 3. El usuario interactúa 4, El usuario sale.
<b>Postcondición</b>	Se añade el usuario a la base de datos.
<b>Excepciones</b>	s
<b>Frecuencia</b>	Alta
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.1: CU-0 Gestión usuarios

## Actores

El actor es el usuario de la aplicación, tenemos que tener en cuenta que para que se de un cambio de turno deben de existir dos actores, sino sería imposible el intercambio.

## Casos de uso

Nota: El editar usuarios y eliminar usuarios no esta implementado de cara al usuario. Esta hecho pero no se encuentra disponible en pantalla para realizarse directamente por parte del usuario.

<b>CU-02</b>	<b>Registro</b>
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-1.1
<b>Descripción</b>	Permite registrarse
<b>Precondición</b>	La página esta cargada en la pestaña adecuada
<b>Acciones</b>	1. El usuario introduce los datos, a saber: A. Nombre B. Apellidos B. Nombre usuario C. E-mail D. Nombre empresa E. Tipo de turno (a elegir) F. Contraseña
<b>Postcondición</b>	Se añade el usuario a la base de datos. Redirección a log-in.
<b>Excepciones</b>	Si no se introducen todos los campos obligatorios se notifica. También si el usuario ya existe o si todo ha ido correctamente
<b>Frecuencia</b>	Alta
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.2: CU-02 Registro



CU-03	Log-in
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-1.2
<b>Descripción</b>	Permite logearse al usuario
<b>Precondición</b>	La página esta cargada en la pestaña adecuada base de datos funcionando
<b>Acciones</b>	1. El usuario se <i>logea</i> para ellos introduce: A. Correo electrónico B. Contraseña
<b>Postcondición</b>	El usuario entra en la aplicación
<b>Excepciones</b>	Si la contraseña no es correcta, aviso Si el correo no es correcto, aviso
<b>Frecuencia</b>	Alta
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.3: CU-03 Log-in

CU-04	Log-out
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-1.3
<b>Descripción</b>	Permite salir de la aplicación
<b>Precondición</b>	El usuario esta dentro de la aplicación
<b>Acciones</b>	1. Presionar el botón <i>log out</i>
<b>Postcondición</b>	Sale de la aplicación
<b>Excepciones</b>	
<b>Frecuencia</b>	Alta
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.4: CU-04 Log out

CU-05	Gestión calendario
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-2.1, RF-2.2, RF-2.3, RF-2.4
<b>Descripción</b>	Gestión del calendario
<b>Precondición</b>	El usuario esta dentro de la aplicación
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Añadir turnos</li> <li>3. Eliminar turnos</li> <li>4. Editar turnos</li> </ol>
<b>Postcondición</b>	Turno deseado
<b>Excepciones</b>	
<b>Frecuencia</b>	Media
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.5: CU-05 Gestión Calendario

CU-06	Añadir turno
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-2.1
<b>Descripción</b>	Añadir un turno determinado al calendario
<b>Precondición</b>	El usuario esta dentro de la aplicación
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. Presionar botón <i>nuevo turno</i></li> <li>2. Seleccionar título</li> <li>3. Seleccionar tipo</li> <li>4. Seleccionar fecha</li> </ol>
<b>Postcondición</b>	Turno deseado en calendario
<b>Excepciones</b>	
<b>Frecuencia</b>	Media
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.6: CU-06 Añadir turno

CU-05	Gestión calendario
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-2.1, RF-2.2, RF-2.3, RF-2.4
<b>Descripción</b>	Gestión del calendario
<b>Precondición</b>	El usuario esta dentro de la aplicación
<b>Acciones</b>	1. Añadir turnos 3. Eliminar turnos 4. Editar turnos
<b>Postcondición</b>	Turno deseado
<b>Excepciones</b>	
<b>Frecuencia</b>	Media
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.7: CU-05 Gestión Calendario

CU-05	Gestión calendario
<b>Versión</b>	1.0
<b>Autor</b>	Adrián Aguado
<b>Requisitos asociados</b>	RF-2.1, RF-2.2, RF-2.3, RF-2.4
<b>Descripción</b>	Gestión del calendario
<b>Precondición</b>	El usuario esta dentro de la aplicación
<b>Acciones</b>	1. Añadir turnos 3. Eliminar turnos 4. Editar turnos
<b>Postcondición</b>	Turno deseado
<b>Excepciones</b>	
<b>Frecuencia</b>	Media
<b>Importancia</b>	Alta
<b>Comentarios</b>	

Tabla B.8: CU-05 Gestión Calendario

## *Apéndice C*

---

# **Especificación de diseño**

---

### **C.1. Introducción**

En este apartado se incluye el diseño de la aplicación y se mostrarán también los diagramas de clases de cada una de las partes que forman el proyecto.

### **C.2. Diseño de datos**

En este apartado se incluye el diseño de la aplicación y se mostrarán también los diagramas de clases de cada una de las partes que forman el proyecto.

## Documentación técnica de programación

---

### D.1. Introducción

En este capítulo vamos a dividirlo en dos partes, por un lado los condiciones específicas para la aplicación y por otro vamos a aprender de manera breve y sencilla a crear una aplicación gracias a *Angular CLI*, la interfaz de línea de comandos de Angular.

El por qué he decidido incluir la parte de *Angular CLI* es por que considero que va aclarar muchas dudas acerca de cómo funciona esta tecnología y también para intentar comprender mejor como he realizado el proyecto. En diversas ocasiones para aprender a usar una tecnología tan solo hace falta tiempo pero en otras muchas el tiempo es limitado por lo que quizás con esta pequeña guía podemos comenzar a utilizar un framework cliente en muy pocos pasos y de una manera sencilla. Si bien es cierto que yo he necesitado mucho tiempo para comprender su funcionamiento y no lo he aprendido todo gracias a la interfaz de comandos, ésta te permite agilizar los trámites de creación en un tanto por ciento considerablemente alto.

Este anexo por tanto tiene como objetivo analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar, es decir, en qué condiciones ha sido desarrollado, en qué condiciones se debe usar y cuáles son los requerimientos mínimos para que un futuro programador interactúe con la aplicación.

### D.2. Requerimientos mínimos necesarios

Los requerimientos mínimos para empezar a trabajar en el proyecto son:

1. Instalar Node.js y MongoDB.

2. Instala Angular CLI

```
npm i -g @angular/cli
```

3. Desde el directorio raíz del proyecto (vía línea de comandos) instalar las dependencias necesarias

■ Stack MEAN

- Mongoose.js (MongoDB): base de datos
- Express.js: backend framework
- Angular 4: frontend framework
- Node.js: entorno tiempo de ejecución

■ Otras tecnologías que se utilizan:

- Angular CLI
- Bootstrap
- Font Awesome
- JSON WEB TOKEN

### D.3. AngularCLI

Sabiendo los requisitos mínimos que son necesarios para proceder

#### ¿Qué es typescript?

Esto se ha nombrado antes pero resulta esencial para entender el código de Angular

*TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. Any browser. Any host. Any OS. Open source.*

([?])

## **Comandos básicos**

### **Puesta en marcha**

## **D.4. Manual del programador**

En esta sección hay que tener en cuenta que el autor de este trabajo fin de grado a escogido unas herramientas, tanto para desplegar la app, como la base de datos como para desarrollar la aplicación pero que de ninguna manera resultan ser ni las únicas ni las mejores simplemente son unas herramientas que ha considerado utilizar pero existen muchas más que no son ni peores ni peores.

Para desplegar la app hemos elegido heroku, para la base de datos mlab ( que dentro tiene servidores AWS, Google Cloud o Azure)

Pasos para montar en tu propio ordenador y desarrollar tu propia API (tener en cuenta que es software desde el que se realiza es un MACBOOK PRO, por lo que pueden existir cambios respecto a otros sistemas operativos. Trataré sin embargo ajustarme y dar detalles para instalarlo en cualquier entorno.

## **Compilación, instalación y ejecución del proyecto**

Voy a tratar de explicar un desarrollo completo desde la instalación en local hasta la carga en un servidor.

### **Estructura de directorios**

### **Modo producción**

### **Estructura de directorios del proyecto**

### **Base de datos**

### **Avisos**

---

## Documentación de usuario

---

### E.1. Introducción

En este capítulo se detalla como un usuario puede comenzar a usa la aplicación deberemos diferenciar dos aspectos diferentes:

- Aplicación Web.
- Aplicación móvil.

### E.2. Instalación

Aplicación Web

Aplicación Móvil

### E.3. Manual del usuario

Aplicación Web

Aplicación Móvil