



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DO SOFTWARE

Diseño e Implementación de una aplicación Web para la gestión de consultas médicas empleando C# y .Net

Estudiante: Gorka Rodríguez Campos

Dirección: Marcos Gestal Pose

A Coruña, febreiro de 2021.

A mis padres

Agradecimientos

A mis padres y a mi familia, porque sin ellos no habria llegado hasta aquí.

Gracias a Bárbara, Olalla y Rocío por todas las videollamadas que han hecho que el trabajo fuera más ameno, a Andrea y a Pepe por invitarme tantos fin de semanas a descansar en su casa para despejarme y a Cely por ser tan buena compañera de piso.

Gracias a todo el personal médico del Centro Médico Río Sil, a mi tutor, a Fran y a Cristina que me han aconsejado y ayudado a lo largo del proyecto.

Y sobre todo gracias a Mire por haber sido la mejor compañera que uno puede tener.

Resumen

En tiempos de constante transformación donde las nuevas tecnologías están revolucionando el sistema en el que vivimos y facilitan la vida de cada uno de los que hacen uso de ella, vemos cómo la tecnología incursiona en campos antes insospechados llegando incluso al campo sanitario.

Este trabajo consiste en el diseño e implementación de un sitio web de gestión de consultas médicas orientado a clínicas. Esta aplicación web está destinada a cuatro tipos de usuarios: el administrador, los pacientes, los médicos y los recepcionistas. Su principal función es actuar de intermediaria entre la clínica y los pacientes y, además, permite su gestión interna.

El lenguaje de programación principal que se ha utilizado para el desarrollo de la aplicación es C#. Además, se ha utilizado para su implementación un conjunto de librerías y frameworks que agilizan la programación, simplifican el código y permiten un desarrollo dinámico, entre los que destaca ASP.NET.

La metodología de desarrollo empleada es una metodología ágil basada en iteraciones. En cada una de estas iteraciones, se desarrollan una o varias funcionalidades completas, de forma que al final de la iteración esté disponible una versión funcional de la aplicación.

Abstract

In times of constant development, with new technologies transforming the world we live in as well as making our lives easier, we can see how technology has made its way into many fields, including the medical field.

This research project focuses on the design and implementation of a management software for medical appointments in clinics. This application is intended for four types of users: the administrator, patients, doctors and receptionists. Its main function is works as a liaison between the clinic and patients, as well as allowing the internal management of the medical centre.

The main programming language used for this software's development is C#. Moreover, a system of frameworks and libraries have been used for its execution, allowing faster programming, simplifying the code along with a dynamic development, ASP.NET being the highlight.

In terms of the methodology of development used, an agile approach based on iterations has been followed. In each of these iterations, one or more features have been developed so that at the end of the iteration, a functional version of the application will be available.

Palabras clave:

- Aplicación web
- ASP.NET
- C#
- Clínica
- Consulta
- Diagnóstico
- Gestor
- Medico
- Paciente
- Prueba médica
- Recepcionista
- Receta médica
- Tratamiento

Keywords:

- Web application
- ASP.NET
- C#
- Clinic
- Consultation
- Diagnostic
- Manager
- Doctor
- Patient
- Medical test
- Receptionist
- Prescription
- Treatment

Índice general

1	Introducción	1
1.1	Motivación	2
1.2	Objetivos	3
1.3	Organización	3
1.3.1	Estado de la cuestión	3
1.3.2	Fundamentos tecnológicos	3
1.3.3	Metodología	3
1.3.4	Planificación y evaluación de costes	4
1.3.5	Desarrollo	4
1.3.6	Pruebas	4
1.3.7	Principales funcionalidades de la aplicación	4
1.3.8	Conclusiones	4
2	Estado de la cuestión	5
2.1	Herramientas de gestión de consultas	5
2.1.1	DriCloud	5
2.1.2	Nubimed	6
2.1.3	Ofimedic	7
2.1.4	Clinic Cloud	8
2.2	Conclusión	8
3	Fundamentos tecnológicos	11
3.1	Hardware	11
3.2	Lenguajes	11
3.2.1	C#	11
3.2.2	HTML	12
3.2.3	CSS	12
3.2.4	SQL	12

3.2.5	JavaScript	12
3.3	Frameworks y librerías	12
3.3.1	.Net	12
3.3.2	Ajax Control Toolkit	15
3.3.3	IText	15
3.4	Herramientas para el desarrollo	15
3.4.1	Visual Studio Enterprise	16
3.4.2	GIT	16
3.4.3	SQL Server Management Studio	18
4	Metodología	19
4.1	Metodologías Ágiles	19
4.2	SCRUM	20
4.3	Adaptación de las metodologías ágiles al proyecto	21
5	Planificación y evaluación de costes	23
5.1	Planificación	23
5.1.1	Sprint 0	23
5.1.2	Sprint 1	24
5.1.3	Sprint 2	24
5.1.4	Sprint 3	24
5.1.5	Sprint 4	24
5.1.6	Sprint 5	24
5.1.7	Sprint 6	25
5.1.8	Total	25
5.2	Evaluación de costes	25
6	Desarrollo	27
6.1	Sprint 0	27
6.1.1	Análisis	27
6.1.2	Diseño	30
6.1.3	Implementación	32
6.2	Sprint 1	32
6.2.1	Análisis	32
6.2.2	Diseño	33
6.2.3	Implementación	38
6.3	Sprint 2	44
6.3.1	Análisis	44

6.3.2	Diseño	45
6.3.3	Implementación	47
6.4	Sprint 3	49
6.4.1	Análisis	49
6.4.2	Diseño	50
6.4.3	Implementación	50
6.5	Sprint 4	55
6.5.1	Análisis	55
6.5.2	Diseño	55
6.5.3	Implementación	57
6.6	Sprint 5	59
6.6.1	Análisis	59
6.6.2	Diseño	60
6.6.3	Implementación	61
6.7	Sprint 6	63
6.7.1	Análisis	63
6.7.2	Diseño	63
6.7.3	Implementación	66
7	Pruebas	67
7.1	Pruebas unitarias	67
7.2	Pruebas integración	67
7.3	Pruebas funcionales	69
8	Funcionalidades más destacadas	71
8.1	Solicitud de consultas	71
8.2	Búsqueda de consultas pendientes	71
8.3	Reorganización de fecha de consultas y avisos	72
8.4	Gestión de los datos clínicos del paciente	72
8.5	Exportación de los datos de una consulta	73
8.6	Cobro de consultas	74
9	Conclusiones	75
9.1	Concordancia de objetivos	75
9.2	Lecciones aprendidas	76
9.3	Líneas futuras	77
9.3.1	Integración con un visor DICOM	77
9.3.2	Cliente SMTP	77

9.3.3	Optimización en el horario laboral de los médicos	78
9.3.4	Autocompletado	78
A	Manual de usuario	81
B	Exportación PDF de una consulta	105
	Lista de acrónimos	119
	Bibliografía	121

Índice de figuras

2.1	Solicitud de una consulta en DriCloud	5
2.2	Ficha de un paciente en Nubimed	6
2.3	Calendario de consultas en Ofimedic	7
2.4	Calendario de consultas en Clinic Cloud	8
4.1	Ciclo de vida de Scrum	21
6.1	Arquitectura de la aplicación	31
6.2	Diagrama de entidades del Sprint 1	34
6.3	Diagrama de IUserProfileDao	35
6.4	Funciones añadidas en UserService del Sprint 1	35
6.5	Mockup de autenticación	38
6.6	Página de autenticación (Authentication.aspx)	42
6.7	Configuración de Web.Config para la autenticación	44
6.8	Diagrama de entidades del Sprint 2	46
6.9	Funciones añadidas en UserService en el Sprint 2	48
6.10	Mockup de búsqueda de empleados	48
6.11	Búsqueda de empleados (SearchEmployees.aspx)	48
6.12	Diagrama de entidades del Sprint 3	51
6.13	Funciones añadidas en UserService en el Sprint 3	52
6.14	Funciones añadidas en ConsultationService en el Sprint 3	52
6.15	Mockup de solicitar consulta	52
6.16	Solicitar consulta (RequestConsultation.aspx)	53
6.17	Diagrama de entidades del Sprint 4	56
6.18	Funciones añadidas en ConsultationService en el Sprint 4	57
6.19	Mockup de ver una consulta	58
6.20	Ver consulta (ShowConsultation.aspx)	59
6.21	Funciones añadidas en ConsultationService en el Sprint 5	60

6.22	Mockup del cobro de consultas	60
6.23	Cobro de consultas (ShowConsultation.aspx)	61
6.24	Factura	62
6.25	Receta médica	62
6.26	Diagrama de entidades del Sprint 6	64
6.27	Funciones añadidas en IInformationService y ConsultationService en el Sprint 6	65
6.28	Mockup de ver los chats de un usuario	65
6.29	Chat (Chat.aspx)	66

Índice de tablas

5.1	Horas dedicadas al proyecto	25
5.2	Costes del proyecto	26
6.1	Historias de usuario	30
6.2	Historias de usuario del Sprint 1	33
6.3	Historias de usuario del Sprint 2	45
6.4	Historias de usuario del Sprint 3	49
6.5	Historias de usuario del Sprint 4	55
6.6	Historias de usuario del Sprint 5	59
6.7	Historias de usuario del Sprint 6	63

Introducción

ESTE trabajo consiste en el diseño e implementación de un sitio web de gestión de consultas médicas orientado a clínicas. Esta aplicación web actúa de intermediaria entre la clínica y los pacientes y, además, permite la gestión interna de la propia clínica.

Hay cuatro tipos de usuarios: los administradores, los médicos, los pacientes y los recepcionistas. que harán uso de sus servicios y que, dependiendo del tipo de usuario que utilice la aplicación, tendrá acceso a distintas funcionalidades.

La aplicación permite que un usuario se registre, inicie y cierre sesión, modifique los datos de su perfil y cambie su contraseña. Para el uso de la aplicación el usuario tiene que estar registrado, y posteriormente haber iniciado sesión. Los usuarios registrados como administradores podrán realizar tareas relacionadas con la gestión de empleados y la gestión de los días festivos. Los usuarios registrados como pacientes podrán solicitar cita previa como también cancelarla o modificar la fecha; además podrán ver su historial clínico y ver las próximas consultas. Los usuarios registrados como médicos podrán realizar tareas relacionadas con la realización y gestión de consultas, como también, ver los historiales clínicos de sus pacientes. Por último, los usuarios registrados como recepcionistas podrán gestionar las consultas de los pacientes como el cobro de estas mismas.

Para el desarrollo de la aplicación se ha empleado una metodología ágil, iterativa e incremental. Por lo tanto, el proyecto se ha dividido en una serie de iteraciones y se han implementado una o varias funcionalidades en cada una de estas iteraciones.

La aplicación se ha dividido en tres proyectos: Model, Test y Web siguiendo una arquitectura en capas. El backend de la aplicación está formado por una capa de acceso a datos y una capa lógica de negocio utilizando como lenguaje C#. El frontend de la aplicación está formado por una capa de acceso a servicios y una capa **Interfaz de Usuario (IU)**. En su implementación, se ha utilizado los formularios Web Forms proporcionados por ASP.NET.

1.1 Motivación

En tiempos de constante transformación donde las nuevas tecnologías están revolucionando el sistema en el que vivimos y facilitan la vida de cada uno de los que hacen uso de ella, vemos cómo la tecnología incursiona en campos antes insospechados llegando incluso al campo sanitario.

En la actualidad, es indispensable tener un ordenador o un teléfono móvil. Es por eso que las empresas han tenido que adaptarse, en mayor o menor medida, a la era digital que estamos viviendo. Además, con las tecnologías vigentes de hoy en día es posible ofrecer al usuario información en tiempo real sobre su historial clínico como también la coordinación de forma precisa entre los distintos trabajadores de una clínica. En resumen, hay muchas posibilidades para que las clínicas puedan ofrecerle una mejor experiencia al usuario. Con el fin de facilitar la adaptación de una clínica a esta era digital, se va a presentar una herramienta para la gestión de sus propias consultas.

El principal motivo por el que se ha realizado una aplicación con esta temática es debido a un interés personal por el sector sanitario y, en particular, por el de atención primaria del paciente. Además el hecho de que mi madre trabaje en este sector, me ha permitido tener más facilidades de documentarme mejor en este ámbito.

Analizando la situación se detectó que el trabajo en una clínica no solo consiste en atender a los pacientes y proveerles salud, pues la verdad es que detrás de una buena atención médica existen una serie de procedimientos administrativos que hacen esto posible. En tal sentido, es necesario proporcionar al personal herramientas necesarias para hacer de la mejor manera posible su trabajo. Al no estar digitalizada una clínica, muchas de las tareas que realizan los empleados día a día se realizarán de forma manual, lo que ocasiona una pérdida de tiempo constante que influye directamente tanto en la calidad de servicio ofrecida a los pacientes como en la gestión de la propia clínica. Para solucionar esta problemática, mi principal objetivo es que la aplicación facilite y agilice el trabajo de los empleados de la clínica permitiéndoles una mejor gestión de sus tareas, de manera que mejore la calidad de su trabajo. Esto se traducirá con una mejora de la calidad de servicio que recibirán los pacientes.

Para la elección de las tecnologías, pensé en qué me convendría aprender y tenía dos opciones: aprender algunas nuevas o emplear las ya conocidas. Considerando el tiempo que perdería aprendiendo nuevas tecnologías, decidí que me vendría mejor afianzar los conocimientos en un campo familiar. Por lo tanto, tenía dos alternativas en mi mente: utilizar las tecnologías utilizadas en Marcos de Desarrollo o las utilizadas en Programación Avanzada. Contemplando ambas opciones, me decanté por la primera, debido a que Java es el lenguaje que más he utilizado durante el grado, por lo que me interesaba más consolidar los conocimientos de C# y .Net aprendidos en Marcos de Desarrollo.

En cuanto a la metodología, he decidido seguir una metodología ágil, por sus ventajas frente a las metodologías tradicionales, como también por la gran utilización que éstas tienen en el entorno laboral.

1.2 Objetivos

Los objetivos del desarrollo de este proyecto han sido los siguientes:

- Fortalecer los conocimientos con las tecnologías empleadas. Tal y como se mencionó en el apartado anterior, las tecnologías que se han usado para realizar la aplicación ya eran conocidas; sin embargo, se buscaba un mayor conocimiento y familiaridad con las mismas.
- Realizar una planificación adecuada y seguirla durante el desarrollo, de manera que cada funcionalidad estuviese terminada en el tiempo establecido al principio de cada iteración.
- Desarrollar una aplicación funcional que cumpliera con todos los objetivos establecidos en el enunciado, siguiendo buenas prácticas tanto en el diseño como en el desarrollo de la misma.

1.3 Organización

A continuación se citan y describen brevemente los capítulos en los que se estructura el presente documento.

1.3.1 Estado de la cuestión

Se analizan las principales herramientas de software médico para la gestión de consultas determinando sus principales características.

1.3.2 Fundamentos tecnológicos

Se justifica la elección de los elementos que se han utilizado en la elaboración del proyecto, como los lenguajes, frameworks, librerías, entorno de desarrollo, etc.

1.3.3 Metodología

Se exponen las ventajas de las metodologías ágiles y se explica la metodología seguida durante el desarrollo, justificando los motivos de su elección.

1.3.4 Planificación y evaluación de costes

En este capítulo se reflejan las distintas fases de elaboración del proyecto, las distintas actividades realizadas en cada fase y la evaluación de costes.

1.3.5 Desarrollo

Se explica como se ha realizado el proyecto de manera que:

- Se definen los límites de la aplicación: qué hará la aplicación y quién la utilizará. También se explican las distintas fases en que se ha dividido la elaboración del proyecto.
- Se explica la arquitectura de la aplicación, explicando y justificando los patrones de diseño que se han seguido.
- Se explica cómo, partiendo del diseño de cada parte de la aplicación, se ha llevado a cabo la implementación de la misma, haciendo uso de distintas herramientas de apoyo.

1.3.6 Pruebas

Para validar la efectividad de la aplicación, se han realizado una serie de pruebas que se describen en este capítulo.

1.3.7 Principales funcionalidades de la aplicación

Se presentan las prestaciones más destacadas de la aplicación, de forma que quede claro qué hace la misma.

1.3.8 Conclusiones

Se revisan los objetivos planteados inicialmente y se contrastan con lo realizado. Se incluyen las lecciones aprendidas y las líneas futuras de la aplicación.

Estado de la cuestión

PARA el comienzo de este proyecto se ha realizado una búsqueda exhaustiva de información acerca de diferentes proveedores de software médico para la gestión de consultas. En este capítulo se analizan herramientas para determinar sus principales características.

2.1 Herramientas de gestión de consultas

2.1.1 DriCloud

DriCloud [1] es un software médico para la gestión de la consulta privada que unifica la totalidad de servicios necesarios: historia clínica en la Nube, gestión económica, recetas electrónicas, página web y citas online. Es un software de pago, ofreciendo diferentes precios dependiendo del número de empleados que vayan a usar la aplicación.

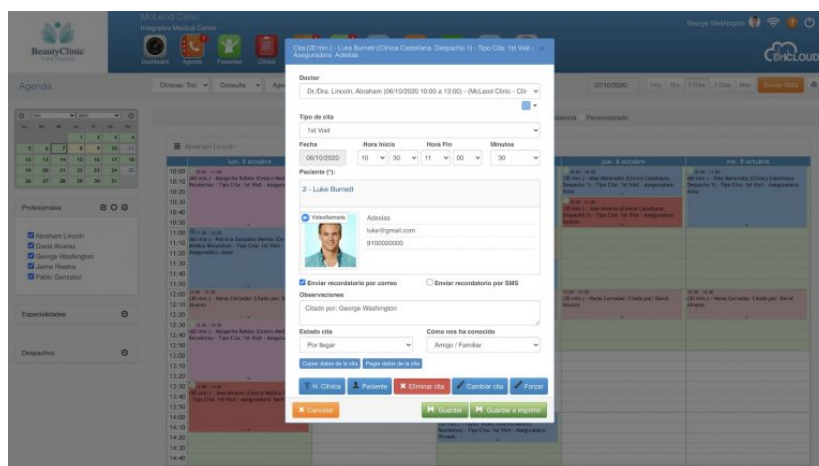


Figura 2.1: Solicitud de una consulta en DriCloud

Tras analizar la aplicación, se destacan las siguientes características:

- Totalmente personalizable que se adapta a las necesidades de cada clínica.
- Gestión de citas médicas. Podemos ver la solicitud de una consulta en la figura 2.1.
- Dashboard en tiempo real para acceder a tareas, mensajes y envío de recordatorios.
- Servicios de telemedicina con videollamadas ilimitadas encriptadas de punto a punto.
- Tiene incorporada pasarela de pago digital para poder cobrar a los pacientes por videollamada, si así se desea, portal del paciente y gestión de citas online.
- Integración con Uber para el transporte de pacientes.
- Exportación de datos.
- Subida de documentos e imágenes.
- Módulo de gestión: estadísticas, facturas, facturación, etc.

2.1.2 Nubimed

Nubimed [2] es un software de gestión de clínicas que ha sido desarrollado en colaboración con profesionales del sector médico. Es la más barata de las herramientas analizadas ya que utiliza un método de pago mensual SaaS, aumentando su precio dependiendo de los doctores que tengan agenda.

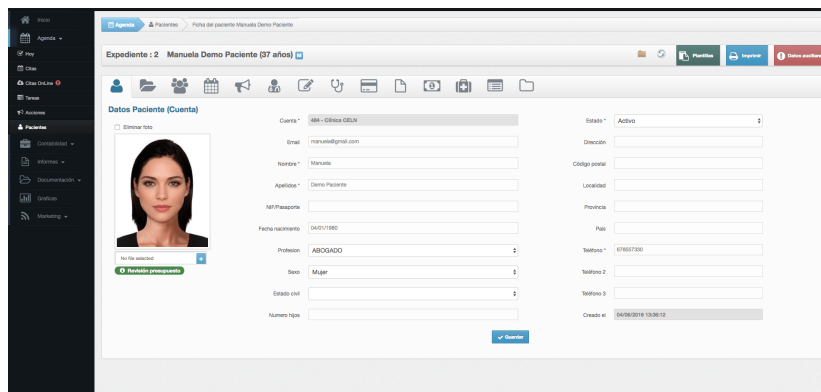


Figura 2.2: Ficha de un paciente en Nubimed

Las principales características que dispone Nubimed son las siguientes:

- Agenda de citas médicas.
- Gestión de pacientes. Se puede ver la ficha de un paciente en la figura 2.2.
- Historia clínica electrónica.

- Plantillas de documentos.
- Módulo de contabilidad y facturación.
- Informes imprimibles y exportables.

2.1.3 Ofimedic

Ofimedic [3] es un software médico diseñado para la gestión global de consultas médicas que surgió a partir del esfuerzo en común de profesionales médicos y de equipos de desarrollo de soluciones financieras por desarrollar un software para el sector médico que abarcara todas sus necesidades de gestión.

Sus principales características son:

- Reducción de errores gracias a la automatización de datos.
- Gestión de agenda y de turnos de trabajo del personal.
- Todo la información del centro se almacena en un único soporte informático y se organiza y clasifica para su consulta en tiempo real: historiales, citas, visitas, facturas, etc. Se puede ver el calendario de consultas en la figura 2.3.
- Control de cobros y pagos de clientes.
- Control de stocks y almacenes.
- Varios métodos de acceso: Ofimedic Desktop, Ofimedic Web y Ofimedic Móvil.

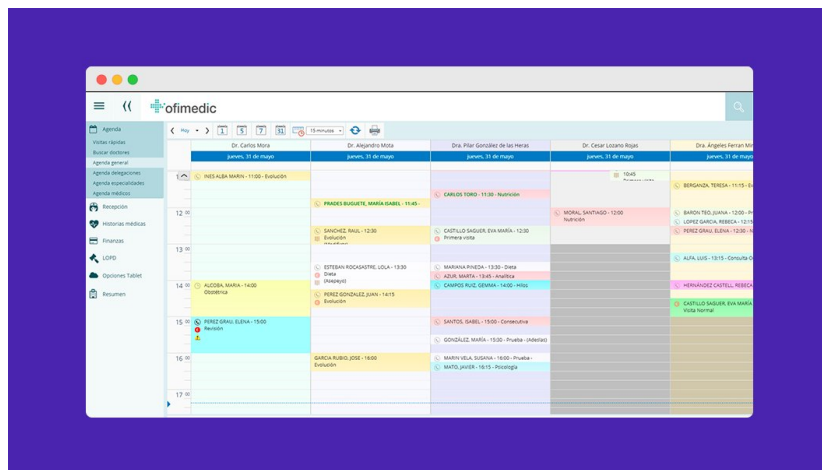


Figura 2.3: Calendario de consultas en Ofimedic

2.1.4 Clinic Cloud

Clinic Cloud [4] es un software que se presenta como una solución a la gestión y administración de clínicas. Lo interesante de esta aplicación es que proporciona un plan gratuito para probar la aplicación con un máximo de 300 pacientes y un solo especialista.

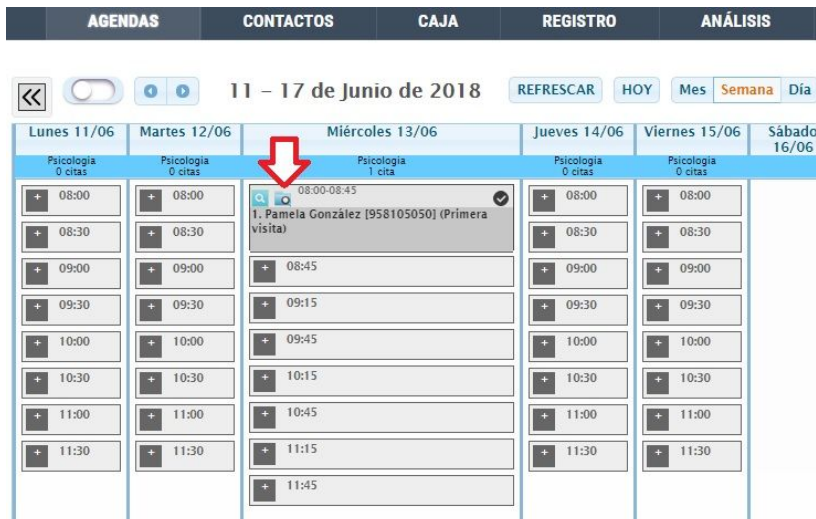


Figura 2.4: Calendario de consultas en Clinic Cloud

Entre sus características destacamos:

- Gestión de citas online que hayas recibido desde tu web y de recetas online.
- Agenda para cada usuario, facilitando el control de citas y sus pagos. Podemos ver el calendario de consultas en la figura 2.4.
- Potente software a precio asequible gracias al modelo SaaS.
- Módulo de facturación para emitir facturas.
- Exportación de datos.

2.2 Conclusión

Tras el análisis se comprobó que todas las herramientas tienen en mayor o menor medida las mismas funcionalidades, centrándose principalmente en los médicos y los pacientes, pero obviando al personal administrativo.

Desde un punto de vista objetivo, los recepcionistas son una parte fundamental en una clínica, ya que son la cara visible de esta misma y los encargados de realizar consultas, mo-

dificar sus horarios o realizar cobros. Es cierto que incluyen funcionalidades propias de este tipo de empleados, pero no lo hacen de forma óptima para facilitar su trabajo diario.

Para solucionar esto, se decidió crear una aplicación web que proporcione según el tipo de usuario diferentes funcionalidades e información que agilicen su trabajo, sin centrarse únicamente en los médicos y pacientes. Por ejemplo, un recepcionista al iniciar sesión podrá ver en su página principal todas las consultas finalizadas pendientes de cobro sin tener que buscar por usuario, en cambio, tanto los pacientes como los médicos, verán sus próximas citas.

Otro de los puntos negativos que he encontrado, es que ninguna de las aplicaciones realiza la reorganización de consultas en caso de que se modifique el horario laboral de un médico o que coja vacaciones. Por lo tanto, se implementará un algoritmo que realice la modificación de horario de las consultas afectadas. Además se intentará avisar a través de la aplicación a los perjudicados, incluyendo a los recepcionistas por si deciden avisar de manera telefónica.

Fundamentos tecnológicos

EN este capítulo se exponen y justifican los elementos que se han utilizado en el proyecto.

3.1 Hardware

Se ha utilizado un equipo personal con las siguientes características:

- Sistema operativo: Windows 10 Home
- Procesador: Intel Core i7-6700HQ 2.60GHz
- Memoria RAM: 8GB
- Tipo de sistema: 64 bits

3.2 Lenguajes

A continuación se explica brevemente qué lenguajes se han utilizado en el proyecto.

3.2.1 C#

Es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft. Está basado en objetos, con raíces en la familia de lenguajes C y creado como parte de la plataforma .NET. Es el lenguaje principal elegido tanto para el desarrollo de la lógica de negocio como para la parte del “Code-Behind” de los Webforms. El principal motivo de su elección es porque quería tener más experiencia con este lenguaje así como también su sencillez. Otro de los motivos son los frameworks y librerías que se mencionan más adelante.

3.2.2 HTML

Se trata de un sistema estandarizado para marcar la elaboración de páginas web, definiendo su estructura permitiendo así, modificar su apariencia con ayuda de [CSS](#). Es el lenguaje utilizado para las páginas de la vista de la aplicación.

3.2.3 CSS

Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy utilizado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en [HTML](#).

3.2.4 SQL

Es un lenguaje de dominio específico utilizado para administrar y recuperar información de sistemas de gestión de bases de datos relacionales. Se ha utilizado para la visualización, adición, modificación y borrado de datos en la aplicación.

3.2.5 JavaScript

JavaScript es un lenguaje interpretado orientado a objetos desarrollado por Netscape. Se ha utilizado en el lado del cliente para proporcionar ciertas mejoras en la interfaz de usuario.

3.3 Frameworks y librerías

3.3.1 .Net

Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de la plataforma de hardware y que permite un rápido desarrollo de aplicaciones. El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes:

- Proporcionar un entorno de programación orientada a objetos coherente en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
- Proporcionar un entorno de ejecución de código que:
 - Minimice los conflictos de implementación de software y control de versiones.
 - Fomente la ejecución segura de código, incluso del creado por terceros desconocidos o que no son de plena confianza.
 - Elimine los problemas de rendimiento de los entornos con scripts o interpretados.

- Ofrecer al desarrollador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en web.

.NET Framework consta de dos componentes principales: Common Language Runtime (CLR) y la biblioteca de clases de .NET Framework.

CLR

Common Language Runtime es un entorno de ejecución que puede ser usado por distintos lenguajes de programación. El CLR es el fundamento de .NET Framework y realiza diversas tareas entre las que destacamos:

- Cargar y ejecutar una aplicación .NET utilizando una compilación JIT para traducir IL a código máquina.
- Se encarga de la recolección de basura (Garbage Collector).
- Gestión de las aplicaciones, de manera que carga, verifica y ejecuta código.
- Proteger las aplicaciones unas de otras como también hacer cumplir las restricciones de seguridad.
- Proporcionar servicios de depuración e incluir soporte para versiones y desarrollo.

Biblioteca de clases

La biblioteca de clases de .NET Framework es una colección de tipos reutilizables que se integran estrechamente con Common Language Runtime. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado deriva funciones. Esto hace que los tipos de .NET Framework sean fáciles de usar y reduce el tiempo asociado con el aprendizaje de las nuevas características de .NET Framework. Además, los componentes de terceros se integran sin dificultades con las clases de .NET Framework.

ADO.NET Entity Framework

ADO.NET Entity Framework es un ORM para .NET. que permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio sin tener que preocuparse de las tablas y columnas de bases de datos subyacentes en las que se almacenan estos datos. Con Entity Framework, los desarrolladores pueden trabajar en un nivel más alto de abstracción cuando tratan con datos, y pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales.

LINQ to Entities

LINQ es una sintaxis de consulta uniforme en C# utilizada para guardar y recuperar datos de diferentes fuentes. Como está integrado en C# elimina la falta de correspondencia entre los lenguajes de programación y las bases de datos, proporcionando una interfaz de consulta única para diferentes tipos de fuentes de información. Sus principales características:

- Lenguaje familiar. Los desarrolladores no tienen que aprender un nuevo lenguaje de consulta para cada tipo de fuente de datos o formato de datos.
- Menos codificación. Reduce la cantidad de código que se debe escribir en comparación con un enfoque más tradicional.
- Código legible. LINQ hace que el código sea más legible para que otros desarrolladores puedan entenderlo y mantenerlo fácilmente.
- Forma estandarizada de consultar múltiples fuentes de datos. la misma sintaxis LINQ se puede utilizar para consultar múltiples fuentes de datos.
- Compilar la seguridad del tiempo de las consultas. Proporciona verificación de tipos de objetos en tiempo de compilación.
- Soporte IntelliSense. LINQ proporciona IntelliSense para colecciones genéricas.

Enterprise library

EL es un conjunto de componentes de software reusables (ApplicationBlocks) que resuelven problemas comunes: seguridad, acceso a datos, registro (logging), inyección de dependencias, entre otros. En nuestro caso se ha implementado la funcionalidad de log en el dll proporcionado en la asignatura de Marcos de Desarrollo.

ASP.NET

ASP.NET [5] es el framework de programación Web dentro de .NET y es la evolución de Active Server Pages (ASP). Nos ofrece tres marcos para crear aplicaciones web: formularios Web Forms, ASP.NET MVC y ASP.NET Web Pages.

Formularios Web Forms Para este proyecto se ha utilizado los formularios Web Forms [6] que son páginas que los usuarios solicitan mediante el explorador. Estas páginas se pueden escribir utilizando una combinación de HTML, script de cliente, controles de servidor y código de servidor. Cuando los usuarios solicitan una página, se compilan y ejecutan en el servidor mediante el marco de trabajo y, a continuación, el marco de trabajo genera el marcado HTML

que el explorador puede representar. El marco basado en formularios Web Forms ofrece las ventajas siguientes:

- Admite un modelo de eventos que conserva el estado sobre [HTTP](#), lo cual favorece al desarrollo de la aplicación web de línea de negocio.
- Usa un modelo de controlador de página que agrega funcionalidad a las páginas individuales.
- Usa el estado de vista o los formularios basados en servidor, lo que puede facilitar la administración de la información de estado.
- Funciona bien para los equipos pequeños de desarrolladores web y los diseñadores que deseen aprovechar el gran número de componentes disponible para el desarrollo rápido de aplicaciones.
- En general, es menos complejo para el desarrollo de aplicaciones, porque los componentes están estrechamente integrados y normalmente requieren menos código que el modelo de MVC.

3.3.2 Ajax Control Toolkit

AJAX Control Toolkit [7] es un proyecto de código abierto construido en la parte superior del marco de Microsoft ASP.NET AJAX. Nos proporciona una potente infraestructura para escribir controles extensores, así como un amplio abanico de controles que se pueden usar fuera de la caja para crear una experiencia web interactiva. Se ha utilizado por los controles “Modal Pop Up” y “File Uploader”.

3.3.3 IText

IText [8] es una biblioteca de código abierto que nos permite crear y manipular archivos PDF y [HTML](#). Fue escrita por Bruno Lowagie y Paulo Soare y está distribuida bajo Affero General Public License. Esta librería nos permite exportar los documentos en múltiples formatos, o múltiples instancias del mismo formato, en nuestro caso en PDF. Se ha utilizado para la exportación de los datos de las consultas, la generación de justificantes médicos y la generación de facturas.

3.4 Herramientas para el desarrollo

A continuación se describen brevemente otras herramientas utilizadas.

3.4.1 Visual Studio Enterprise

Visual Studio [9] es un entorno de desarrollo integrado para Windows y MacOS desarrollado por Microsoft. Ofrece las siguientes características:

- Editor de código adaptado al lenguaje en cuestión, en este caso C#. Compilación integrada.
- IntelliTrace: depuración histórica para ayudar a los equipos a depurar problemas difíciles de reproducir.
- IntelliTest: generación automática de datos de pruebas inteligentes y conjuntos de pruebas unitarias para el código.
- Integración con Git.
- Ahorro de tiempo gracias a diversas opciones como el autocompletado o los atajos de teclado.

NuGet

NuGet es una extensión de Visual Studio que hace más fácil agregar, eliminar y actualizar referencias a librerías y herramientas en proyectos de Visual Studio que utilizan .NET Framework. Ofrece las siguientes características:

- Cuando se agrega una librería o una herramienta, NuGet copia los archivos a su solución y realiza automáticamente los cambios que sean necesarios en el proyecto, tales como la adición de referencias y cambiar el archivo `app.config` o `web.config`.
- Cuando se elimina una biblioteca, NuGet elimina los archivos y revierte los cambios que hizo en su proyecto. NuGet proporciona una manera rápida y fácil de agregar características a una aplicación existente siempre que estas características están integradas en control de código fuente.

3.4.2 GIT

Git [10] es un software de control de versiones. Entre sus ventajas podemos destacar:

- Trabajo en equipo. Permite que varios desarrolladores trabajen al mismo tiempo y en paralelo en un proyecto con un acceso compartido, pero sin estar físicamente cerca, así como identificar qué usuario y cuándo ha realizado cada modificación.
- Seguimiento de errores.

- Mayor autonomía. Cada desarrollador cuenta con una copia local de todo el proyecto y de los cambios generados, lo que le permite trabajar de forma individual y a su propio ritmo, en cualquier momento y lugar.
- Permite comparar, fusionar o restaurar versiones de una aplicación y contar con una copia del código fuente para volver atrás ante cualquier imprevisto.

Git-flow

Para el desarrollo del proyecto se ha seguido el modelo de Git flow que es un flujo de trabajo aplicado a un repositorio Git ¹. Esta filosofía es ideal para proyectos que lleven una planificación de entregas iterativas. Permite la paralelización del desarrollo mediante ramas independientes para la preparación, mantenimiento y publicación de versiones del proyecto así como soporta la reparación de errores en cualquier momento. El modelo de Git-flow se divide en dos tipos de ramas: las principales y las de apoyo.

- La rama principal está formada por:
 - Rama master. Contiene cada una de las versiones estables del proyecto. Cualquier commit que subamos en esta rama debe estar preparado para que se pueda incluir en producción.
 - Rama develop. Contiene el código de desarrollo de la siguiente versión planificada del proyecto. En ella se incluirán cada una de las nuevas características que se desarrollen. Esta rama puede incorporarse tanto en una rama release (que veremos más adelante) como en la rama master, para su despliegue en producción.
- La rama de apoyo formada por:
 - Ramas feature. Estas ramas tienen que surgir de la rama develop. Cada una de estas ramas almacenan código de desarrollo con nuevas características.
 - Ramas release. Como las ramas feature, las ramas release también tienen que surgir de la rama develop. Contienen el código de la versión que se va a liberar próximamente. Es un paso previo y preparatorio para la versión definitiva de producción. En ella se incluye todo el código de develop necesario para el lanzamiento.
 - Ramas hotfix. Estas ramas surgen de la rama master. Contienen una versión de producción con un error que se desea arreglar urgentemente. Una vez arreglado el error, se incluye el contenido de esta rama en las ramas master y develop para subsanar el error.

¹ Repositorio Git del proyecto: <https://git.fic.udc.es/gorka.rodriguez/healthfic>

3.4.3 SQL Server Management Studio

Es la herramienta principal para la gestión de bases de datos SQL Server. Se basa en el shell de Visual Studio utilizado para proyectos de desarrollo de software, y brinda las siguientes características:

- Explorador de objetos. Se trata de un panel desde la cual se puede conectar a instancias de SQL Server y administrar los objetos que contiene.
- Editor de código. Puede administrar servidores de base de datos y objetos utilizando interfaces gráficas.
- Soluciones y Proyectos. Puede utilizar proyectos y soluciones para mantener los scripts relacionados, conexiones y otros documentos juntos.
- Reportes. Incluye una interfaz de informes extensible que se puede utilizar para ver información de configuración y estado detallado acerca de las instancias de SQL Server, bases de datos y otros objetos.

Metodología

LA metodología de desarrollo establece un marco de trabajo que guía todo el proceso de construcción del proyecto. Esta elección es vital, ya que de ella depende la correcta consecución de los objetivos establecidos. Por lo tanto, es necesario tomar una decisión bien fundamentada.

4.1 Metodologías Ágiles

Existen muchas metodologías desarrolladas para su aplicación en proyectos de desarrollo software. El estudio de estas metodologías es un área dentro de la ingeniería del software. El uso de las metodologías conocidas como metodologías ágiles se ha extendido mucho durante los últimos años. Estas se caracterizan por el uso de ciclos de vida iterativos que se suceden a lo largo del tiempo, para construir un producto funcional al final de cada ciclo. El objetivo de esta división en iteraciones es satisfacer los requisitos fundamentales del sistema en las primeras iteraciones, para dedicar las siguientes fases a añadir funcionalidades o refinar características. Las principales ventajas de las metodologías ágiles son:

- Permite poner el foco en las personas. Al tener empleados motivados, en un entorno adecuado y que sientan que son apoyados redundan en unos mejores resultados.
- Fomenta la responsabilidad dentro del equipo y aumenta la autonomía y la transparencia. El trabajo colaborativo, la fluidez en la comunicación de los equipos y la participación equitativa de todas las partes hacia un objetivo común, claro y conciso ayuda a que los equipos sean más eficaces y autónomos.
- Permite dimensionar mejor los proyectos minimizando los riesgos. Las revisiones continuas permiten la adaptación al cambio de una manera más eficiente, que evita sorpresas tanto para el equipo como para el cliente respecto a los tiempos de entrega y los costes.

- Facilita la priorización y toma de decisiones. Al estar todas las personas involucradas con un ritmo de trabajo constante y evitarse las jerarquías, la operativa es más rápida maximizando los resultados buscando la eficiencia y optimización.
- Mejora la experiencia del cliente. A través de feedbacks rápidos y precisos, gracias a un modelo de comunicación más colaborativo y cercano, donde el cliente es uno más del equipo, se consigue mayor nivel de satisfacción y ahorro de tiempo.
- Entrega de productos y servicios en intervalos con rápida respuesta a los cambios.
- Incrementa el valor añadido y la predictibilidad de los resultados. Este modelo de trabajo y la cercanía con el cliente permite tener mayor conocimiento del mismo, aportando valor diferencial y evitando desviaciones sobre el plan.

4.2 SCRUM

Scrum [11] es un marco de trabajo definido por Ikujiro Nonaka y Takeuchi a principios de los 80. Su objetivo es el desarrollo ágil de software mediante la aplicación de un conjunto de buenas prácticas para trabajar en equipo de forma colaborativa y obtener el mejor resultado posible. Sus principales características son:

- El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado (time to market). Motivación del equipo.
- Mayor productividad. Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- Constante comunicación entre el equipo de desarrollo y el cliente.
- Flexibilidad a cambios. Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado.

Las historias de usuario son descripciones breves y concisas sobre una funcionalidad del sistema, descritas desde la perspectiva del usuario que empleará dicha característica. Generalmente, siguen el siguiente formato:

Como [rol] quiero [funcionalidad]

Las historias de usuario forman el Product Backlog, que es una lista que contiene todo lo que se conoce que es necesario para el proyecto y es la única fuente de requisitos del producto. En ella se establecen todas las funcionalidades, cambios y mejoras que deben ser hechos en un

futuro. Es un documento que nunca está completo, evoluciona y cambia a la vez que lo hace el producto.

El sprint es un concepto fundamental en scrum. Es un evento que dura entre una y cuatro semanas. Durante cada sprint se desarrolla un incremento del producto formado por varias historias de usuario, que constituye un elemento funcional y usable y que es potencialmente entregable al cliente. El desarrollo del producto se lleva a cabo en varios sprint, que suceden a lo largo del tiempo. Es recomendable que la duración de estos sea siempre la misma y que no se introduzcan eventos no contemplados, para reducir la aparición de riesgos. En la figura 4.1 podemos ver el ciclo de vida de Scrum.

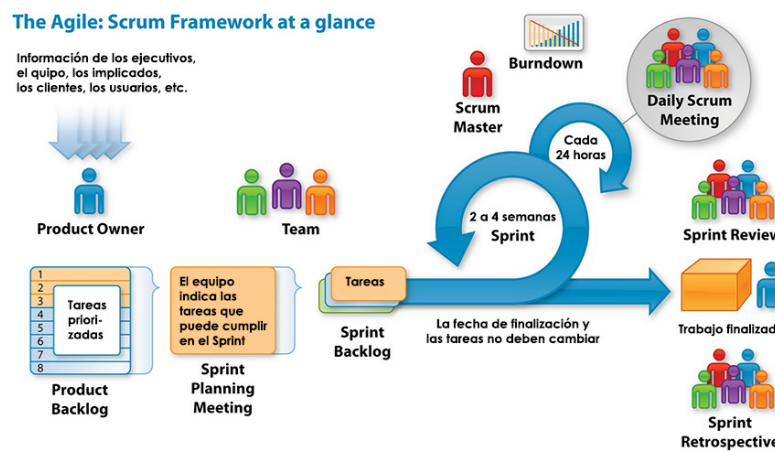


Figura 4.1: Ciclo de vida de Scrum

4.3 Adaptación de las metodologías ágiles al proyecto

Muchos de los principios de las metodologías ágiles no se pueden aplicar en este proyecto debido a los siguientes condicionantes:

- Al ser un proyecto realizado por una única persona, no se pueden aplicar los principios relacionados con la comunicación entre miembros de trabajo, el trabajo en equipo, el trabajo por pares, etc. Sin embargo, sí se ha intentado simular a un cliente, hablando con diferentes especialistas del sector médico para que dieran su opinión y ofrecieran sus consejos de mejora.
- El tiempo de dedicación al proyecto no ha sido constante a lo largo del desarrollo, debido a la realización en un largo período de tiempo con situaciones variadas en cuanto a la disponibilidad horaria.

Igualmente, se ha elegido una metodología ágil por la cantidad de ventajas que aporta sobre una metodología tradicional. No se ha elegido ninguna en concreto, sino que basándose en el conocimiento previo de varias de ellas, especialmente de Scrum, se ha intentado adoptar los principios más convenientes en cada caso:

- La simplicidad en el diseño se aplica en varias fases a lo largo del desarrollo. El diseño se ha hecho por funcionalidades similares, de forma que cada una de ellas se ha implementado con aquellas que tienen relación y se ha intentado de que todo el código sea lo más sencillo posible. Además, se han incluido comentarios en cada función para facilitar la comprensión del programa.
- Eliminación de interrupciones y gestión del flujo. Durante el desarrollo del proyecto se ha evitado abarcar la implementación de varias funcionalidades que no tuvieran relación para evitar distracciones innecesarias, poniendo foco en cada momento en cuestiones similares.
- Sprint. Igual que en el caso anterior, se podría realizar la planificación temporal del proyecto de muchas formas, debido a la gran cantidad de tiempo disponible para realizarlo. Sin embargo, por cuestiones de organización y compromiso, se han establecido sprints con una duración fija de 2 semanas y media.
- Historias de usuario. Para estimar el esfuerzo necesario para llevar a cabo una historia de usuario, se utilizan los puntos de historia. Estos puntos representan una medida abstracta del esfuerzo necesario para completar una historia de usuario, que debe contemplar todo lo que puede afectar a la correcta consecución de esta: cantidad de trabajo a realizar, complejidad del trabajo, riesgo o incertidumbre, etc.
- Enfoque desde el punto de vista del cliente . Aunque no exista un cliente propiamente dicho, en todo momento el desarrollo se ha llevado a cabo teniendo en cuenta las necesidades que tendría un usuario final de la aplicación. Para ello, como había comentado se ha ido mostrando el proyecto durante los sprints a varios especialistas del sector para saber sus opiniones y sugerencias.

Planificación y evaluación de costes

EN este capítulo se detalla la planificación temporal de cada uno de los sprints, incluyendo su duración en semanas y las horas de esfuerzo dedicadas. Partiendo de esta planificación, se estima el coste del proyecto.

5.1 Planificación

El tiempo dedicado a la realización del proyecto ha sido de 21 semanas (mediados de octubre - febrero de 2021). Siguiendo la filosofía de Scrum, la duración de los sprints ha sido constante. Como hay 7 sprints, se ha establecido que la duración de cada sprint sea de 2 semanas y media. Esto hace un total de 17 semanas y media de trabajo, por lo que queda 3 semanas y media de márgenes, y descansos. Para estimar la duración del proyecto, se ha seguido el criterio de que un punto de historia equivale a 1 hora. El proyecto está formado por 418 puntos de historia, lo que equivale a 418 horas de trabajo en total.

Para desarrollar cada historia de usuario, se han llevado a cabo actividades de análisis, diseño, implementación y pruebas. Se estima que el análisis supone un 10% del esfuerzo, el diseño un 30%, la implementación un 30% y las pruebas un 30%.

En el sprint 0 se realiza un análisis general del proyecto, que se ha estimado que representa el 70% de todo el análisis realizado (es decir, un 7% del esfuerzo total del proyecto). En cada sprint se realizan el análisis restante (el 30% no incluido en el análisis general), el diseño, la implementación y las pruebas de las historias de usuario correspondientes.

5.1.1 Sprint 0

En el sprint 0 se realiza el análisis general del proyecto, que incluye todas las historias de usuario y supone un 7% del esfuerzo total del proyecto; es decir, 29 horas y 16 minutos.

5.1.2 Sprint 1

En el sprint 1 se implementan las historias de usuario de la tabla 6.2, que suman un total de 1 hora y 39 minutos en tareas de análisis, 16 horas y 30 minutos en tareas de diseño, 16 horas y 30 minutos en tareas de implementación y 16 horas y 30 minutos en tareas de pruebas. El tiempo total de esfuerzo en este sprint es de 51 horas y 9 minutos, repartidos a lo largo de 2 semanas y media.

5.1.3 Sprint 2

En el sprint 2 se implementan las historias de usuario de la tabla 6.3, que suman un total de 2 horas y 22 minutos en tareas de análisis, 23 horas y 42 minutos en tareas de diseño, 23 horas y 42 minutos en tareas de implementación y 23 horas y 42 minutos en tareas de pruebas. El tiempo total de esfuerzo en este sprint es de 73 horas y 28 minutos, repartidos a lo largo de 2 semanas y media.

5.1.4 Sprint 3

En el sprint 3 se implementan las historias de usuario de la tabla 6.4, que suman un total de 2 horas y 40 minutos en tareas de análisis, 26 horas y 42 minutos en tareas de diseño, 26 horas y 42 minutos en tareas de implementación y 26 horas y 42 minutos en tareas de pruebas. El tiempo total de esfuerzo en este sprint es de 82 horas y 46 minutos, repartidos a lo largo de 2 semanas y media.

5.1.5 Sprint 4

En el sprint 4 se implementan las historias de usuario de la tabla 6.5, que suman un total de 2 horas y 15 minutos en tareas de análisis, 22 horas y 30 minutos en tareas de diseño, 22 horas y 30 minutos en tareas de implementación y 22 horas y 30 minutos en tareas de pruebas. El tiempo total de esfuerzo en este sprint es de 69 horas y 45 minutos, repartidos a lo largo de 2 semanas y media.

5.1.6 Sprint 5

En el sprint 5 se implementan las historias de usuario de la tabla 6.6, que suman un total de 1 hora y 57 minutos en tareas de análisis, 19 horas y 30 minutos en tareas de diseño, 19 horas y 30 minutos en tareas de implementación y 19 horas y 30 minutos en tareas de pruebas. El tiempo total de esfuerzo en este sprint es de 60 horas y 28 minutos, repartidos a lo largo de 2 semanas y media.

5.1.7 Sprint 6

En el sprint 6 se implementan las historias de usuario de la tabla 6.7, que suman un total de 1 hora y 39 minutos en tareas de análisis, 16 horas y 30 minutos en tareas de diseño, 16 horas y 30 minutos en tareas de implementación y 16 horas y 30 minutos en tareas de pruebas. El tiempo total de esfuerzo en este sprint es de 51 horas y 9 minutos, repartidos a lo largo de 2 semanas y media.

5.1.8 Total

Sprint	Análisis (h)	Diseño (h)	Impl. (h)	Pruebas (h)	Total (h)
0	29,26	0	0	0	29,26
1	1,65	16,5	16,5	16,5	51,15
2	2,37	23,7	3,7	23,7	73,47
3	2,67	26,7	26,7	26,7	82,77
4	2,25	22,5	22,5	22,5	69,75
5	1,95	19,5	19,5	19,5	60,47
6	1,65	16,5	16,5	16,5	51,15
Total	41,8	125,4	125,4	125,4	418

Tabla 5.1: Horas dedicadas al proyecto

5.2 Evaluación de costes

Los recursos humanos dedicados al proyecto son los siguientes:

- Análisis, diseño, implementación y pruebas: Gorka Rodríguez Campos.
- Dirección: Marcos Gestal Pose.

Para calcular los costes de los recursos humanos, se ha consultado el Estudio salarial del Sector TIC en Galicia 2015-2016 [12], donde se han tomado las siguientes elecciones:

- Dirección: Jefe de proyecto Software (Provincia de A Coruña) experto, 42,00 € / hora.
- Análisis y diseño: Analista Programador Java (Provincia de A Coruña), 23,00 € / hora.
- Implementación y pruebas: Programador Java Web junior, 18,00 € / hora.

Las tareas de análisis y diseño suman un total de 167 horas y 12 minutos, lo que equivale a 3845,60 €.

Las tareas de implementación y pruebas suman un total de 250 horas y 48 minutos, lo que equivale a 4514,40 €.

Se estima que el director ha dedicado al proyecto 2 horas por cada sprint, lo que hace un total de 14 horas y equivale a 840,00 €.

Tarea	Recurso	Coste	Dedicación	Total
Análisis y diseño	Analista	23,00 €/h	67,2 h	3845,60 €
Implementación y pruebas	Programador Junior	18,00 €/h	250,8 h	4514,40 €
Dirección	Jefe de proyecto	42,00 €/h	14 h	588,00 €
				8948,00 €

Tabla 5.2: Costes del proyecto

En cuanto a los recursos materiales, se ha utilizado un ordenador personal propiedad del alumno. No se ha imputado su coste ya que habría que calcular la parte proporcional de uso respecto a la vida útil y no supondría un gran cambio con respecto al coste total del proyecto. Por lo tanto, el coste estimado total sin IVA del proyecto es de 8948,00 €. En la tabla 7.2 se recoge la descomposición de los costes del proyecto comentada.

Aplicando un IVA del 21%, el coste estimado total sería de 10.827,08 €.

Desarrollo

PARA el desarrollo de la aplicación, el proyecto se ha agrupado en 7 sprints. En cada uno de ellos, se van a explicar los siguientes apartados:

- **Análisis:** se define los límites que tendrá.
- **Diseño:** se justifican las decisiones de diseño que se han tomado, junto con las prácticas y patrones aplicados.
- **Implementación:** se explica cómo, partiendo del diseño de la aplicación, se ha llevado a cabo la implementación de la misma con el apoyo de diversos frameworks y librerías.

6.1 Sprint 0

6.1.1 Análisis

Para realizar el análisis, se han definido los actores del sistema (aquellos que usarán la aplicación) y las historias de usuario (es decir, las funcionalidades que proporcionará la aplicación a los usuarios finales).

Actores del sistema

Existen cuatro tipos de actores:

- **Administradores:** son los administradores de la clínica, cuyo objetivo principal es mantener la información de la página web acorde con el funcionamiento de la empresa, es decir, el registro de empleados y la gestión de sus vacaciones, como también, la gestión de los días festivos. Su registro se hace manualmente a través de la base de datos.

- Médicos: son el personal sanitario de la clínica, cuyo principal objetivo es la gestión de consultas médicas como también la gestión de toda la información médica del paciente. Deben estar registrados previamente por el administrador para el uso de la aplicación.
- Recepcionistas: son el personal administrativo, cuyo principal objetivo es el cobro de consultas y su gestión horaria. Además pueden realizar el registro de pacientes. Deben estar registrados previamente por el administrador para el uso de la aplicación.
- Pacientes: son los clientes de la clínica. Su principal objetivo es la solicitud de consultas. Deben estar registrados previamente para el uso de la aplicación.

Historias de usuario

Los requisitos funcionales del proyecto se expresan como historias de usuario, que forman el Product Backlog. En la tabla 6.1 se recogen las historias de usuario de la aplicación y los puntos de historia que se le han asignado a cada una.

ID	Historia de usuario	Puntos
1	Como paciente, quiero poder registrarme en la aplicación	5
2	Como administrador, quiero poder registrar a médicos y recepcionistas	10
3	Como paciente, administrador, médico o recepcionista, quiero poder iniciar sesión en la aplicación	10
4	Como administrador, quiero poder buscar empleados	12
5	Como paciente, médico, recepcionista o administrador, quiero poder ver mi perfil	10
6	Como paciente, médico, recepcionista o administrador, quiero poder modificar mi perfil	10
7	Como paciente, médico, recepcionista o administrador, quiero poder modificar mi contraseña	5
8	Como administrador, quiero poder modificar el perfil de los recepcionistas y médicos	5
9	Como recepcionista, quiero poder registrar a pacientes	5
10	Como paciente, quiero solicitar una consulta	10
11	Como recepcionista o médico. quiero solicitar una consulta para un paciente	5
12	Como paciente, médico o recepcionista, quiero poder modificar la fecha de una consulta	30
13	Como paciente, médico o recepcionista, quiero poder cancelar una consulta	5
14	Como recepcionista o médico, quiero buscar pacientes	12

15	Como médico, quiero poder añadir pruebas a una consulta	10
16	Como médico, quiero poder eliminar pruebas de una consulta	10
17	Como médico, quiero poder modificar las pruebas de una consulta	10
18	Como paciente, quiero ver mis pruebas	6
19	Como médico, quiero ver las pruebas de un paciente	1
20	Como médico, quiero poder añadir un diagnóstico a una consulta	3
21	Como médico, paciente o recepcionista, quiero poder añadir documentos a una consulta	10
22	Como médico, paciente o recepcionista, quiero poder borrar documentos a una consulta	5
23	Como médico, quiero poder añadir antecedentes a un paciente	5
24	Como médico, quiero poder eliminar antecedentes de un paciente	5
25	Como médico, quiero poder modificar los antecedentes de un paciente	5
26	Como paciente, quiero ver mis antecedentes	7
27	Como médico, quiero ver las antecedentes de un paciente	3
28	Como médico, quiero poder añadir vacunas a un paciente	5
29	Como médico, quiero poder eliminar vacunas de un paciente	5
30	Como médico, quiero poder modificar vacunas de un paciente	5
31	Como paciente, quiero ver mis vacunas	7
32	Como médico, quiero ver las vacunas de un paciente	3
33	Como médico, quiero poder añadir tratamientos a una consulta	5
34	Como médico, quiero poder eliminar tratamientos de una consulta	3
35	Como médico, quiero poder modificar los tratamientos de una consulta	5
36	Como paciente, quiero ver mis tratamientos	5
37	Como médico, quiero ver los tratamientos de un paciente	1
38	Como médico, paciente o recepcionista, quiero poder exportar a PDF los datos de una consulta	30
39	Como médico, quiero poder generar un justificante de asistencia	2
40	Como paciente o paciente, quiero ver mis consultas	8
41	Como paciente, recepcionista o médico, quiero ver sugerencias en el momento de solicitar una consulta	10
42	Como médico, quiero poder ver las consultas de un paciente	8
43	Como administrador, quiero poder añadir festivos	5
44	Como administrador, quiero poder eliminar festivos	2
45	Como médico o paciente, quiero poder enviar mensajes	10
46	Como médico o paciente, quiero poder ver mis mensajes	10

47	Como médico o paciente, quiero poder ver mis consultas pendientes en una determinada fecha	10
48	Como recepcionista, quiero poder cobrar las consultas	20
49	Como administrador, quiero poder añadir vacaciones a los empleados	5
50	Como administrador, quiero poder eliminar vacaciones de los empleados	2
51	Como paciente, médico o recepcionista, quiero poder recibir avisos sobre las consultas	13
52	Como recepcionista, quiero poder ver las consultas pendientes de pago	8
53	Como administrador, quiero poder ver gráficos de las ganancias de consultas	10
		418

Tabla 6.1: Historias de usuario

6.1.2 Diseño

Arquitectura global

La aplicación desarrollada está formada por tres proyectos (Model, Test y Web), que consta de:

- Un frontend basado en los formularios ASP.NET Web Forms y se corresponde con el proyecto Web.
- Un backend al que accede el frontend formado por una capa modelo y se corresponde con el proyecto Model.

El diseño por capas es una técnica de diseño muy utilizada. Está basado en la idea de que una capa inferior proporciona servicio a una capa superior. Este servicio se define mediante un contrato de servicio. De esta forma, es posible independizar el software de ambas capas, ya que a la capa superior no le importa cómo se implementa el servicio. Los principales beneficios son:

- Abstracción. Las capas permiten cambios que se realicen en un nivel abstracto. Se puede incrementar o disminuir el nivel de abstracción usado en cada capa de la “pila” jerárquica.
- Aislamiento. El estilo de arquitectura de capas permite aislar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema total.

- Rendimiento. Distribuir las capas entre múltiples sistemas puede incrementar la escalabilidad, la tolerancia a fallos y el rendimiento.
- Mejoras en Pruebas. La capacidad de realizar pruebas se beneficia de tener una interfaces bien definidas para cada capa así como de la habilidad para cambiar a diferentes implementaciones de las interfaces de cada capa.
- Flexibilidad. Los desarrolladores de una capa no necesitan conocer las tecnologías usadas en otras capas.
- Reutilización. Nos permite la posibilidad de reusar la misma capa modelo en distintas interfaces o distintas aplicaciones.

En cuanto a las desventajas del diseño por capas, las principales son:

- El software es más complejo.
- Tolerancia a los fallos: Si una capa falla, todas las capas superiores comienzan a fallar en cascada.
- En ciertos casos, se podría optimizar la implementación de una capa si esta conociese el funcionamiento interno de otra capa. que lo necesiten.

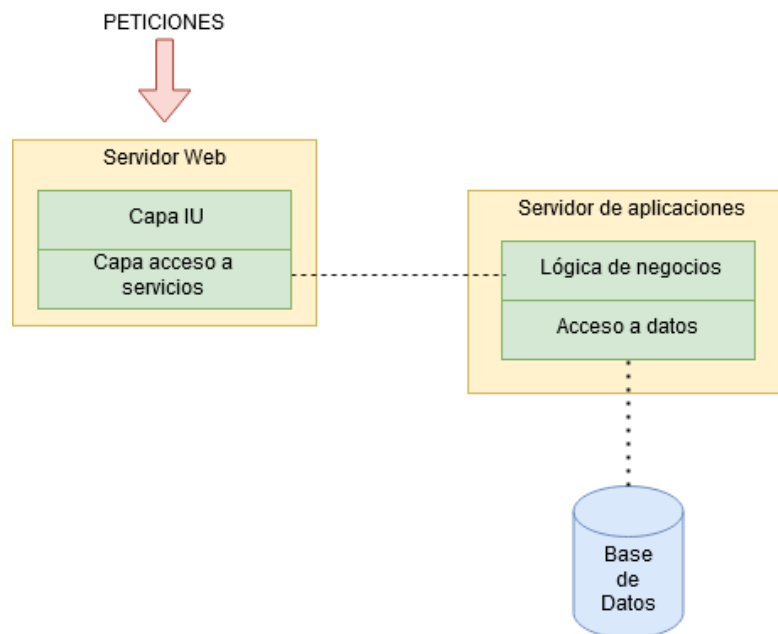


Figura 6.1: Arquitectura de la aplicación

En este caso, la aplicación se divide en las siguientes capas que podemos ver en la figura 6.1.

- Backend:
 - Capa modelo:
 - * Capa de acceso a datos. Permite el acceso a la base de datos de la aplicación. Está formada por distintas entidades y un **DAO** por cada entidad.
 - * Capa lógica de negocio. Implementa la lógica de negocio, utilizando la capa de acceso a datos para leer y escribir los datos que necesite.
- Frontend:
 - Capa de acceso a servicios. Permite al frontend acceder a las funcionalidades del backend.
 - Capa **IU**. Permite a los usuarios utilizar la funcionalidad de la capa modelo.

6.1.3 Implementación

En este sprint, se ha dedicado a la configuración del entorno de desarrollo y crear la estructura del proyecto tomando como base el proyecto proporcionado en Marcos de Desarrollo.

6.2 Sprint 1

Al ser el primer sprint, se va a detallar todos los conceptos que tendrán en común los siguientes, de manera que la lectura sea más intuitiva y fácil de leer. Para los próximos sprints se detallarán las funcionalidades, librerías y tecnologías nuevas sin tener que profundizar en lo que ya se relató previamente.

6.2.1 Análisis

En este sprint se han realizado las historias de usuario que se ven en la tabla 6.2, que suman un total de 55 puntos.

El objetivo de este sprint es el de la realización de todas las funcionalidades relacionadas con el registro y autenticación de los diferentes usuarios que usarán la aplicación. Es el sprint con menos puntos debido a que todas estas funcionalidades se habían realizado en otro proyecto previamente.

ID	Historia de usuario	Puntos
1	Como paciente, quiero poder registrarme en la aplicación	5
2	Como administrador, quiero poder registrar a médicos y recepcionistas	10
3	Como paciente, administrador, médico o recepcionista, quiero poder iniciar sesión en la aplicación	10

5	Como paciente, médico, recepcionista o administrador, quiero poder ver mi perfil	10
6	Como paciente, médico, recepcionista o administrador, quiero poder modificar mi perfil	10
7	Como paciente, médico, recepcionista o administrador, quiero poder modificar mi contraseña	5
9	Como recepcionista, quiero poder registrar a pacientes	5
		55

Tabla 6.2: Historias de usuario del Sprint 1

6.2.2 Diseño

Backend

Entidades y tablas En el diagrama de entidades (ver figura 6.2) se muestran las entidades que han surgido en esta iteración y las relaciones entre ellas. En este diagrama se reflejan todas las entidades existentes, sus atributos y las relaciones entre ellas. Las principales entidades de este sprint son:

- **UserProfile:** Es un usuario de la aplicación, que puede ser administrador, médico, paciente o recepcionista. En esta entidad incluye todos los datos que tienen en común los diferentes tipos de usuario.
- **Doctor:** Representa a un médico de la aplicación. En él se incluyen los datos adicionales que tiene el personal médico.
- **Receptionist:** Representa a un recepcionista de la aplicación. En él se incluyen los datos que tienen a mayores los recepcionistas.
- **Patient:** Representa a un paciente de la aplicación. En él se incluyen los datos adicionales que tienen los pacientes.

DAO Para la implementación de los **DAO** se ha utilizado un ensamblado proporcionado en la asignatura de Marcos de Desarrollo. Por lo tanto, todos los **DAO** extienden de **IGenericDao**, que define las operaciones **CRUD**. En cada uno de estos **DAO** se aplica el patrón de diseño Fachada, por lo que se define una interfaz y una clase que implementa las operaciones correspondientes. A modo de ejemplo, en la figura 6.3 se incluye el diagrama de **IUserProfileDao**, en el que se reflejan los métodos disponibles en **IGenericDao**.

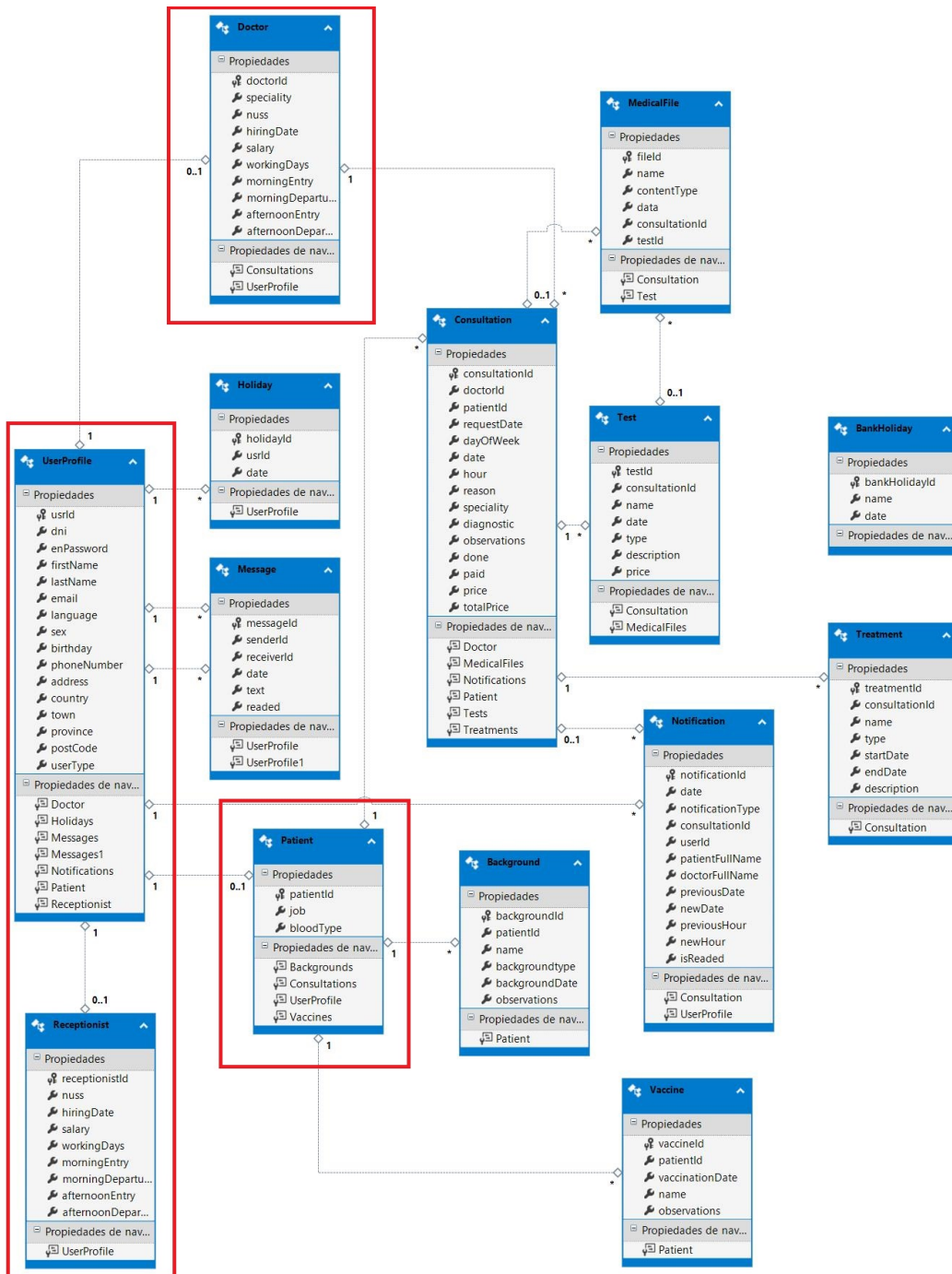


Figura 6.2: Diagrama de entidades del Sprint 1

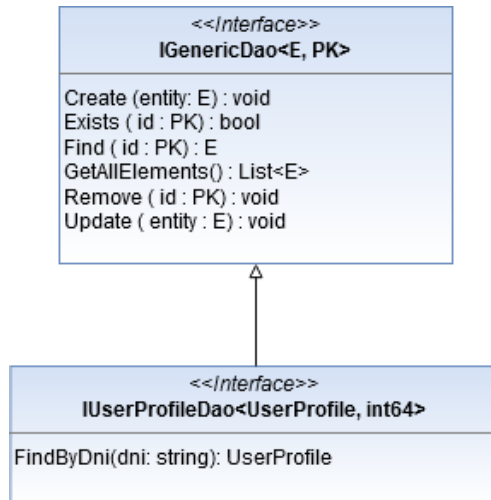


Figura 6.3: Diagrama de IUserProfileDao

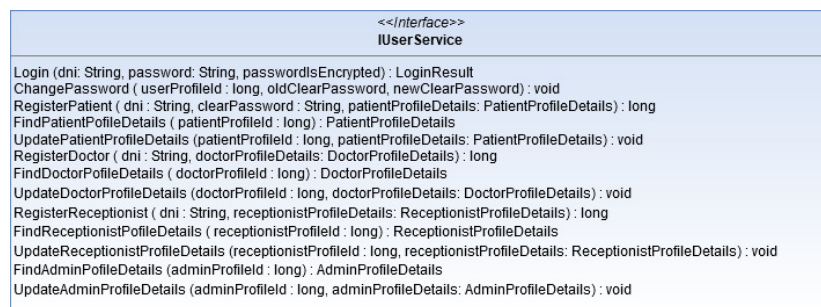


Figura 6.4: Funciones añadidas en UserService del Sprint 1

Servicios Los servicios corresponden con la capa de lógica de negocio ofreciendo una **API** que permite invocar de forma sencilla cada caso de uso y oculta los detalles de implementación. De esta manera, se independiza del resto de capas. Cada una de las funcionalidades que ofrece esta capa se corresponde, normalmente, con una funcionalidad concreta que desea ejecutar el usuario de la aplicación. En cada uno de estos servicios se aplica el patrón de diseño Fachada, por lo que se define una interfaz y una clase que implementa las funcionalidades correspondientes.

En este sprint ha surgido el servicio de usuarios (UserService) donde se incluyeron todas las operaciones relacionadas con el registro de usuarios, cambio de contraseña, ver perfiles, etc. Podemos ver su interfaz con las operaciones que se incluyeron en este sprint en la figura 6.4.

Excepciones A continuación se explican las excepciones que se han contemplado en la lógica de negocio de este sprint.

- DuplicateInstanceException. Se utiliza al intentar crear una entidad con una clave única repetida.
- InstanceNotFoundException. Se emplea cuando se intenta acceder a un recurso que no existe.
- DuplicateEmailException. Se emplea cuando se introduce un email ya existente en el registro o actualización de un usuario.
- DuplicateNussEception. Se utiliza cuando se introduce un número de la seguridad social ya existente en el registro o actualización de un empleado.
- IncorrectDniException. Se utiliza cuando se introduce un dni con formato incorrecto en el registro o actualización de un usuario.
- IncorrectPasswordException. Se emplea para indicar que no se ha introducido una contraseña correcta.
- IncorrectHoraryException. Se emplea al introducir un horario laboral no válido en un empleado.

Patrón DTO El patrón **DTO** tiene como finalidad de crear un objeto plano con una serie de atributos que puedan ser enviados o recuperados del servidor en una sola invocación, de tal forma que un **DTO** puede contener información de múltiples fuentes o tablas y concentrarlas en una única clase simple. Además otra de las características que tiene un **DTO** es que debe ser de solo lectura y serializable (tanto él como sus atributos).

En esta capa se han creado diversos **DTO** para la transferencia de información entre el backend y el frontend de la aplicación.

Algunas funcionalidades de la aplicación requieren que el usuario complete ciertos datos. En los casos en que estos datos se envíen en el cuerpo de la petición, se ha utilizado un **DTO** que los contenga. Por ejemplo para completar los datos de registro de un usuario.

En otros casos, se han empleado **DTO** cuando se necesita mostrar información al usuario de manera reducida. Por ejemplo, `PendingDoctorConsultationDTO` se emplea para mostrar la información de las consultas pendientes de un médico.

Frontend

El diseño del frontend se ha dividido en: la capa de acceso a servicios y la capa **IU**, que se explicarán en los siguientes apartados.

Capa de acceso a servicios La función de la capa de accesos a servicios es gestionar las peticiones que se envían en el frontend para acceder a las funcionalidades proporcionadas por el backend de la aplicación.

Esta capa está contenida en el módulo **HTTP** que está en el proyecto Web. Para mantener la misma estructura que en el backend, se han creado diferentes submódulos que corresponden a cada uno de los servicios del backend. En este sprint, existe el submódulo `Session` que incluye el fichero `SessionManager` que corresponde con las funcionalidades relacionadas con el registro y autenticación de los usuarios que se realizaron en `UserService`.

Capa IU Para diseñar la capa **IU** se han realizado previamente unos mockups para tener una visualización previa de la interfaz de usuario. (Ver figura 6.5)

Esta capa se ha dividido en diferentes módulos dentro del módulo `Pages`, de esta manera se facilita la gestión de código. Dentro de cada módulo, se han incluido las diferentes páginas que se accederán en la aplicación.

Cada una de estas páginas se corresponden a los Web Forms de ASP.Net que los usuarios solicitan mediante el explorador. Estas páginas se han utilizado como una combinación de **HTML**, script de cliente, controles de servidor y código de servidor. Cuando los usuarios solicitan una página, se compilan y ejecutan en el servidor mediante el marco de trabajo y, a continuación, el marco de trabajo genera el marcado **HTML** que el explorador puede representar.

Estas páginas se extienden de `SpecificCulturePage`, que es una clase creada específicamente para que la página se inicialice con las preferencias culturales del usuario si está logueado o en el caso de no estar autenticado, de las preferencias del navegador. Además, todas ellas,

tienen una Master Page común en la que se incluyen todos los componentes comunes: el encabezado, el menú de navegación y el pie de página.

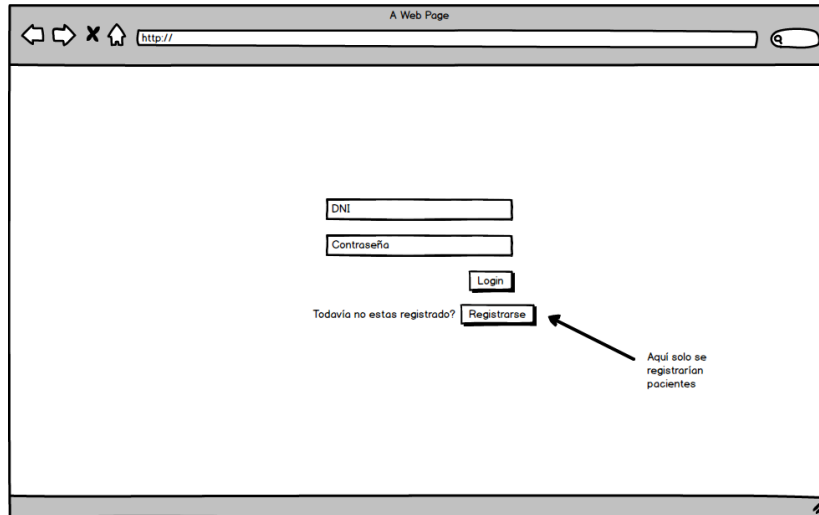


Figura 6.5: Mockup de autenticación

6.2.3 Implementación

En este apartado se explica cómo, partiendo del diseño de la aplicación, se ha llevado a cabo la implementación de la misma con el apoyo de diversos frameworks y librerías.

Backend

Entidades y relaciones Para realizar el mapping de entidades a tablas se ha realizado por medio ADO.NET Entity Framework creando un modelo a partir de una base de datos creada previamente en SQL Server Management Studio. El script para la creación de la BD se encuentra en el módulo Sql en el proyecto Model, y en el caso del proyecto Test, en el módulo Scripts.

Además para la generación automática de los métodos Equals y ToString se ha modificado el archivo ".tt" agregando código utilizando T4 Templates. El método Hashcode no se ha implementado debido a que al sobrescribirlo, producía un fallo al borrar las entidades previamente modificadas.

DAO Las interfaces de los DAO no revelan ningún detalle de implementación debido a que extienden de IGenericDao proporcionado en el ensamblado ModelUtil de Marcos de Desarrollo. Como dicho ensamblado solo incluye las operaciones CRUD básicas, se han creado diversos métodos para acceder a ciertos tipos de datos utilizando LINQ. Como algunas de

estas operaciones eran complejas, como es en el caso de la búsqueda de todas las consultas que han sido afectadas por la modificación del horario laboral de un médico, se ha utilizado PredicateBuilder proporcionado por LinqKit. Esta clase, permite crear predicados de manera clara y limpia, haciendo así, que se entienda mejor la consulta.

Otro de los motivos por los que se han creado nuevos métodos, es por la paginación, como es el caso de mostrar las consultas de un paciente.

Inyección de dependencias Este paradigma se aplica a los **DAO** y a los servicios locales, mediante el uso de anotaciones de Ninject:

- En cada uno de los servicios, todos los **DAO** necesarios estarán anotados con la anotación [Inject].

Para el caso de los Test, se ha creado una clase llamada TestManager en la que se configura el kernel de Ninject utilizando además el archivo ninjectConfiguration.xml.

Transaccionalidad Mediante una extensión, Ninject ofrece soporte a puntos de la Programación Orientada a Aspectos. Ninject Permite Interceptar la llamada a un método, analizar los contenidos de dicha llamada o bien continuar con la misma sin modificación o bien modificar comportamiento. Esto va a ser útil para la gestión de transacciones. Esta gestión nos la proporciona el ensamblado de ModelUtil en el que incluye un interceptor de transacciones, que detectará todos aquellos métodos anotados con la anotación [Transactional].

Frontend

El frontend se ha desarrollado en el proyecto Web. A continuación se describe el proceso de implementación que se ha seguido, basado en el diseño previo, y las consideraciones que se han tenido en cuenta para el desarrollo de sus dos capas: la capa **IU** y la capa de acceso a servicios. Como mencioné en el apartado de diseño, el proyecto Web se ha dividido en varios módulos.

Módulo HTTP En este módulo se incluyeron diversos submódulos que corresponden con las funcionalidades a las que se quieren acceder en el backend. Cada una de estas clases se inicializará inyectando las dependencias necesarias para acceder a su servicio correspondiente.

Módulo Pages En este módulo se incluyen todas las páginas que se accederán durante el funcionamiento de la aplicación. Estas páginas se agruparon según a los servicios a los que delegan y están basadas en los formularios Web Forms que son instancias de la clase Page que incluye propiedades como:

- Session: instancia de la clase HttpSessionState, dedicada a almacenar datos específicos del usuario que necesitan mantenerse entre solicitudes a distintas páginas.
- Application: instancia de HttpApplicationState, pensado para almacenar datos globales a la aplicación.
- Request: instancia de HttpRequest que representa valores y propiedades de la solicitud **HTTP** que causó que tu página cargara.
- Request: destacamos las siguientes propiedades:
 - Browser: permite consultar características del navegador.
 - Cookies: obtiene las colecciones de cookies enviadas con la solicitud.
 - QueryString: proporciona los parámetros que se pasaron con la cadena de consulta.
 - Url y UriReferrer: proporciona un objeto **URI** que representa la dirección actual de la página y la página de la que viene el usuario.
- Response: instancia de HttpResponse, que representa la respuesta del servidor Web a una solicitud de un cliente. Tiene las siguientes propiedades y métodos interesante:
 - Cookies: colección de cookies enviadas con la respuesta.
 - Redirect(): indica al navegador que solicite otra **URL**.

Para el diseño de los Web Forms se decidió utilizar el modelo Code-Behind que proporciona una serie de características:

- Separación de cada página en:
 - .aspx: contiene etiquetas **HTML** y controles ASP.NET.
 - .aspx.cs: que contiene el código fuente de la página.
 - .aspx.designer.cs: que contiene código generado automáticamente.
- Separación clara de la interfaz de usuario

A parte de estos formularios, existen las Master Pages que permiten crear un diseño común, que será compartido por varias páginas. En nuestro caso se ha utilizado para compartir:

- Head: donde se incluyen todas las referencias a scripts y hojas de estilo.
- Menú de navegación.
- Footer o pie de página.

Además se han incluido los controles de servidor `HTML` que son elementos `HTML` que incluyen un atributo “`runat=server`”. Los controles de servidor tienen el mismo resultado y las mismas propiedades que sus etiquetas `HTML` correspondientes. Además, los controles de servidor proporcionan administración de estado automática y eventos del servidor. Todos estos controles tienen una serie de propiedades entre las que destacamos:

- `ClientID`: Identificador único, creado por ASP.NET cuando se instala la página.
- `EnableViewState`: indica si el control debería mantenerse estado entre peticiones.
- `ID`: identificador del control. Es el nombre a través del cual se puede acceder al control desde código.
- `Visible`: indica si el control debe renderizarse.
- `DataBind()`: enlaza el control con un `DataSource`.

Además existen los controles de validación que se usan para validar los valores que se especifican en otros controles de la página. Los controles de validación realizan la validación del lado cliente, la validación del servidor o ambas, en función de las funciones del explorador en el que se muestra la página. Entre estos controles los que he utilizado son:

- `RequiredFieldValidator`: comprueba que el control que tiene que validar no está vacío en el momento de enviar el formulario.
- `CompareValidator`: valida contra un valor constante o contra otro control (mayor que, menor que, etc).
- `RegularExpressionValidator`: valida contra un patrón o expresión regular.

En este sprint se creó el módulo `User`, donde se incluyeron las siguientes páginas:

- `Authentication.aspx`. Aquí se realizará la autenticación para acceder a la aplicación. (Ver figura 6.6)
- `ChangePassword.aspx`. En esta página se cambiará la contraseña del usuario.
- `Logout.aspx`- Página para cerrar sesión.
- `RegisterEmployee.aspx`. Página que incluye el formulario de registro de los empleados.
- `RegisterPatient.aspx`. Página que incluye el formulario de registro de los pacientes.
- `RegisterPatientByReceptionist.aspx`. Página que incluye el formulario de registro de los pacientes a través de un recepcionista.

- ShowAdminProfile.aspx. Página que muestra el perfil de un administrador.
- ShowEmployeeProfile.aspx. Página que muestra el perfil de un empleado.
- ShowPatientProfile.aspx. Página que muestra el perfil de un paciente
- UpdateAdminProfile.aspx. Página que incluye el formulario de la actualización del perfil de un administrador.
- UpdateEmployeeProfile.aspx. Página que incluye el formulario de la actualización del perfil de un empleado.
- UpdatePatientProfile.aspx. Página que incluye el formulario de la actualización del perfil de un paciente.

The image shows a screenshot of a web application's authentication page. At the top, there is a dark navigation bar with the text 'HealthFic Home Autenticación'. The main content area features a white login form with a blue header 'Autenticación'. The form contains two input fields: 'Usuario' and 'Contraseña'. Below the password field is a checkbox labeled 'Recordar mi contraseña (las cookies deben estar habilitadas)'. A blue 'Aceptar' button is positioned below the checkbox. At the bottom of the form, there is a blue link that says 'Registrarme (usuario no registrado)'. The page is framed by a dark footer containing the text 'Copyright © Facultad de Informática - Universidade da Coruña'.

Figura 6.6: Página de autenticación (Authentication.aspx)

Otras cuestiones de internacionalización

Internacionalización En el frontend se trata la internacionalización de los mensajes y algunos datos en inglés, gallego y castellano. Para ello ASP.NET nos ofrece los archivos de recursos, donde se almacenan los diferentes datos que se utilizarán en la aplicación según las preferencias de idioma. Estos recursos tienen las siguientes características:

- Se almacenan en las carpetas App_LocalResources en el caso de páginas individuales y para aquellos recursos presentes en varias páginas en App_GlobalResources.
- Formato: <NombreWebForm>.aspx.<codigoCultura>.resx

Para su acceso hay dos tipos de localización:

- Localización implícita: a través del atributo “resource_key”.
- Localización explícita: a través de la expresión “<%%\$ Resources: Class, Resource_ID%>”.

Además para la internacionalización de los elementos de los desplegados y algunos datos fijos en algunos atributos de las entidades (como en el caso de los tipos de pruebas), se ha creado la carpeta ApplicationObjects en el módulo View donde se han creado las siguientes clases que se encargarán de traducir sus respectivos datos:

- BackgroundTypes.cs : Tipos de antecedentes.
- BloodTypes.cs : Tipos de sangre.
- Countries.cs : Tipos de países.
- Genders.cs : Tipos de género
- Languages.cs : Tipos de idiomas.
- Months.cs : Meses del año.
- Specialities.cs : Tipos de especialidades.
- Tests.cs : Tipos de pruebas.
- Treatments.cs : Tipos de tratamientos.
- UserTypes.cs : Tipos de usuarios.
- NotificationTypes.cs : Tipos de notificaciones.

Autenticación y autorización ASP.Net define diferentes formas de autenticación entre la que escogí la autenticación basada en formularios, ya que es una solución flexible que se integra fácilmente con la aplicación Web. Para ello hay que hacer una serie de configuraciones:

- En IIS: Establecer el método de autenticación como “acceso anónimo”
- En el archivo de configuración añadir una serie de parámetros. (Ver figura 6.7)

Posteriormente, llamaremos al método Login, generándose un ticket de autenticación en forma de cookie y la añade al response.

Para la autorización, se define el control de acceso para recurso específico, mediante una sección <location> en el archivo de configuración. Además tenemos que tener claro las dos políticas existentes:

- Política restrictiva:
 - Primero se deniega el acceso a todos los usuarios anónimos: <deny users = "?" />
 - Luego se da acceso a los recursos que no necesitan autenticación.
- Política permisiva:
 - Primero se permite el acceso a todos los recursos: <allow users = "*" />
 - Luego se indican aquellos recursos que necesitan autenticación.

```

<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms
        name=".ASPXAUTH" | nombre de la cookie de autenticación
        loginUrl="/Authentication.aspx" | URL a la que debe redirigirse la
        | solicitud de inicio de sesión, si no se
        | encuentra ninguna cookie de
        | autenticación válida
        timeout="30" | minutos de vigencia de la cookie
        path="/"
        defaultUrl="/MainPage.aspx"
        cookieless="AutoDetect" />
      </authentication>
    </system.web>
  </configuration>

```

Figura 6.7: Configuración de Web.Config para la autenticación

Web Responsive Se ha decidido que la aplicación sea capaz de adaptarse a cualquier dispositivo, debido a que al ser una página web, algunas consultas pueden que se lleguen a solicitar desde dispositivos móviles, de esta manera se mejora la experiencia del usuario. Para hacer que la aplicación sea responsiva se han utilizado clases proporcionadas por Bootstrap.

Encriptación de la base de datos [13] Para dar cumplimiento a las normativas que aplica el Reglamento General de Protección de Datos de la Unión Europea se ha decidido encriptar la base de datos a través de un cifrado transparente. Para ello se ha creado un script para realizar la encriptación que se ha incluido en la carpeta "Sql" del proyecto Model.

6.3 Sprint 2

6.3.1 Análisis

En este sprint se han realizado las historias de usuario que se ven en la tabla 6.3, que suman un total de 79 puntos.

El objetivo de este sprint es el de la búsqueda tanto de empleados como de pacientes. Además se permite al personal sanitario la gestión de los datos médicos de los pacientes, es decir, de las vacunas y los antecedentes médicos.

ID	Historia de usuario	Puntos
4	Como administrador, quiero poder buscar empleados	12
8	Como administrador, quiero poder modificar el perfil de los recepcionistas y médicos	5
14	Como recepcionista o médico, quiero buscar pacientes	12
23	Como médico, quiero poder añadir antecedentes a un paciente	5
24	Como médico, quiero poder eliminar antecedentes de un paciente	5
25	Como médico, quiero poder modificar los antecedentes de un paciente	5
26	Como paciente, quiero ver mis antecedentes	7
27	Como médico, quiero ver las antecedentes de un paciente	3
28	Como médico, quiero poder añadir vacunas a un paciente	5
29	Como médico, quiero poder eliminar vacunas de un paciente	5
30	Como médico, quiero poder modificar vacunas de un paciente	5
31	Como paciente, quiero ver mis vacunas	7
32	Como médico, quiero ver las vacunas de un paciente	3
		79

Tabla 6.3: Historias de usuario del Sprint 2

6.3.2 Diseño

Backend

Entidades y tablas En el diagrama de entidades (ver figura 6.8) se muestran las entidades que han surgido en esta iteración y las relaciones entre ellas. Las principales entidades de este sprint son:

- Vaccine: Representa las vacunas suministradas de un paciente.
- Background: Representa los antecedentes médicos de un paciente.

Servicios En este sprint se incluyeron en UserService todas las operaciones relacionadas con la búsqueda de usuarios y las operaciones **CRUD** de las vacunas y antecedentes médicos de un paciente. Podemos ver su interfaz con las operaciones que se incluyeron en este sprint en la figura 6.9.

Paginación Se ha empleado paginación para mostrar los datos de las vacunas y antecedentes de un paciente y tanto para la búsqueda de empleados como de pacientes. Para ello se ha creado la clase Block donde almacena una parte de la lista de los elementos resultado, según el tamaño previamente indicado, y contiene un campo que indica si existen aún más elementos.

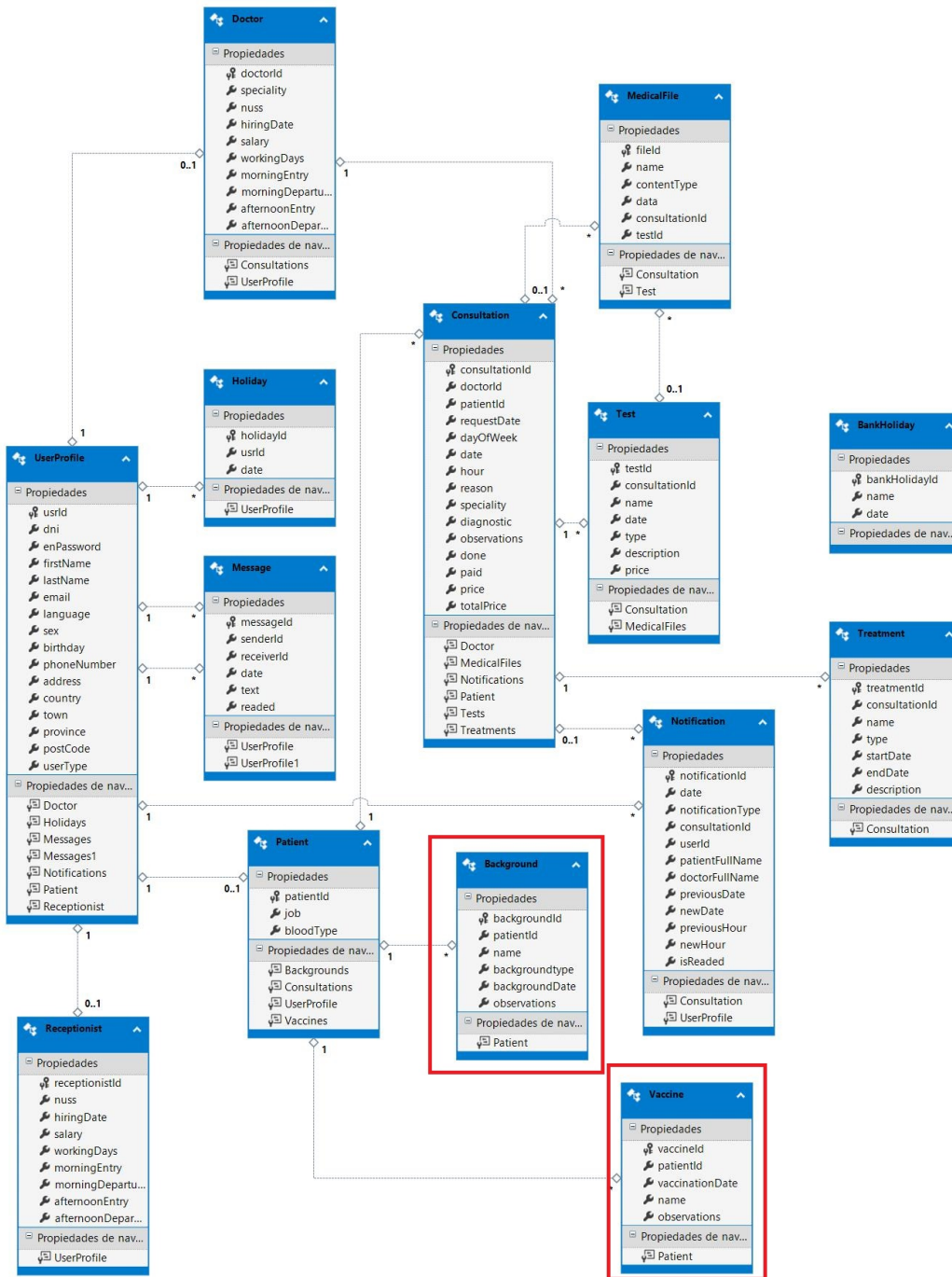


Figura 6.8: Diagrama de entidades del Sprint 2

Frontend

Al igual que en el sprint 1, se han realizado los mockups relacionados con las funcionalidades que se van a implementar en este sprint. En la figura 6.10 podemos ver el mockup de búsqueda de empleados.

6.3.3 Implementación

En este apartado se explica cómo, partiendo del diseño de la aplicación, se ha llevado a cabo la implementación de la misma con el apoyo de diversos frameworks y librerías.

Backend

En este apartado no se han realizado implementaciones a destacar a mayores de las ya explicadas en el sprint 1.

Frontend

Módulo Pages En este sprint, se incluyeron en el módulo User las siguientes páginas.

- SearchBackground.aspx. Página que incluye las operaciones **CRUD** de los antecedentes médicos de un paciente.
- SearchEmployees.aspx. Página dedicada a la búsqueda de empleados. (Ver figura 6.11)
- SearchPatients.aspx. Página para la búsqueda de pacientes.
- SearchVaccines.aspx. Página que incluye las operaciones **CRUD** de las vacunas de un paciente.
- UpdateEmployeeProfileByAdmin.aspx. Página que incluye el formulario de la actualización del perfil de un empleado realizado por un administrador.

Ajax Control Toolkit Para añadir más funcionalidades, se incluyó los controles proporcionados por Ajax Control Toolkit. Para poder utilizar dichos controles, se ha incluido en el proyecto el ensamblado y se ha agregado en cada página en la que se quería utilizar la siguiente etiqueta:

```
<% Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="cc1"
```

Entre estos controles los que he utilizado son:

- ConfirmButtonExtender: control que muestra mensaje de confirmación.
- ModalPopupExtender: control utilizado para mostrar pop-up.


```

<<Interface>>
IUserService

FindEmployeesByKeywordsAndUserType(keywords : String, userType : String, startIndex : int, count : int) : Block<UserDTO>
FindPatientsByKeywords(keywords : String, startIndex : int, count : int) : Block<UserDTO>
CreateVaccine(patientId : long, name : String, observations : String, vaccinationDate : DateTime) : long
UpdateVaccineDetails(vaccineId : long, name : String, observations : String, vaccinationDate : DateTime) : void
FindVaccineDetails(vaccineId : long) : VaccineDetails
RemoveVaccine(vaccineId : long) : void
FindVaccinesOfPatientByKeywords(patientId : long, keywords : String, startIndex : int, count : int) : Block<VaccineDTO>
CreateBackground(patientId : long, name : String, backgroundType : String, observations : String, backgroundDate : DateTime) : long
UpdateBackgroundDetails(backgroundId : long, name : String, backgroundType : String, observations : String, backgroundDate : DateTime) : void
FindBackgroundDetails(backgroundId : long) : BackgroundDetails
RemoveBackground(long backgroundId) : void
FindBackgroundsOfPatientByKeywordsAndBackgroundType(patientId : long, keywords : String, backgroundType : String, startIndex : int, count : int) : Block<BackgroundDTO>
    
```

Figura 6.9: Funciones añadidas en UserService en el Sprint 2

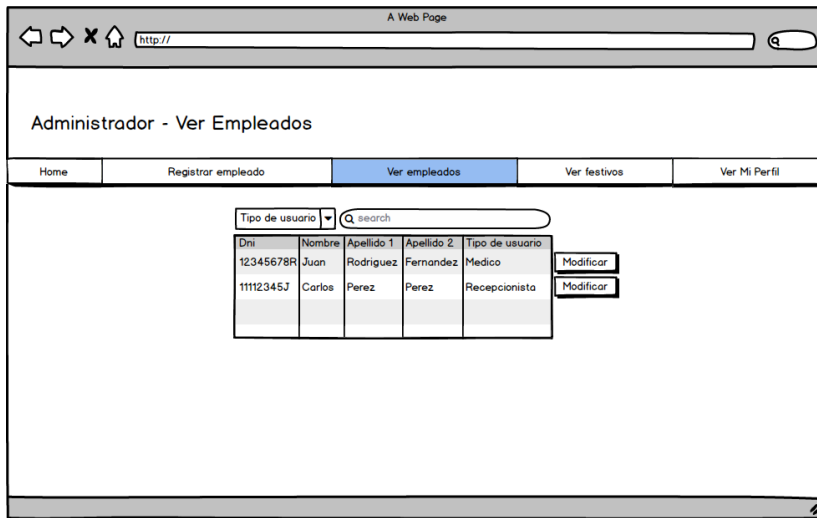


Figura 6.10: Mockup de búsqueda de empleados

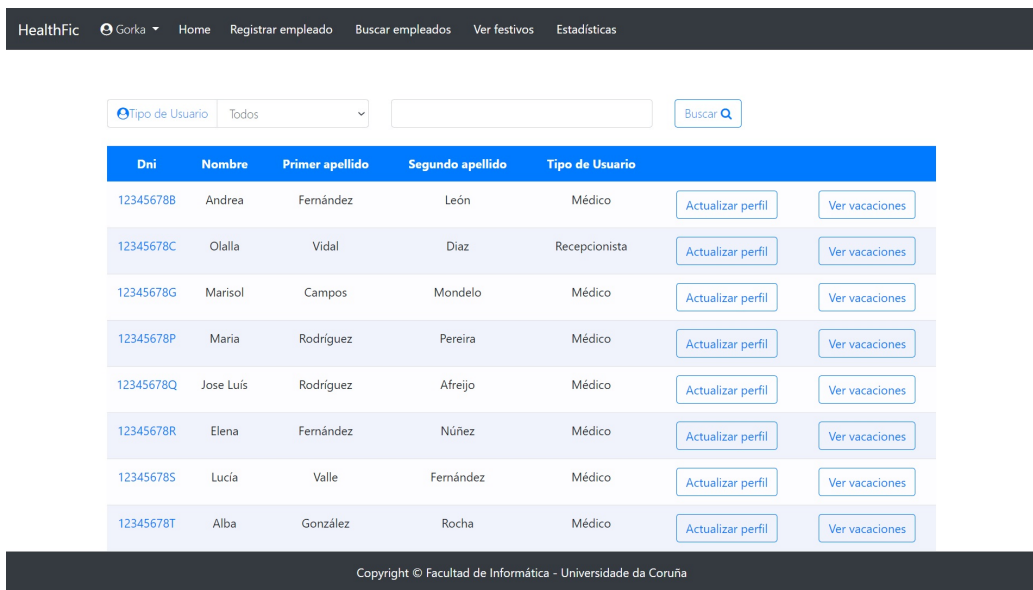


Figura 6.11: Búsqueda de empleados (SearchEmployees.aspx)

6.4 Sprint 3

6.4.1 Análisis

ID	Historia de usuario	Puntos
10	Como paciente, quiero solicitar una consulta	10
11	Como recepcionista o médico. quiero solicitar una consulta para un paciente	5
12	Como paciente, médico o recepcionista, quiero poder modificar la fecha de una consulta	30
13	Como paciente, médico o recepcionista, quiero poder cancelar una consulta	5
21	Como médico, paciente o recepcionista, quiero poder añadir documentos a una consulta	10
22	Como médico, paciente o recepcionista, quiero poder borrar documentos a una consulta	5
43	Como administrador, quiero poder añadir festivos	5
44	Como administrador, quiero poder eliminar festivos	2
47	Como médico o paciente, quiero poder ver mis consultas pendientes en una determinada fecha	10
49	Como administrador, quiero poder añadir vacaciones a los empleados	5
50	Como administrador, quiero poder eliminar vacaciones de los empleados	2
		89

Tabla 6.4: Historias de usuario del Sprint 3

En este sprint se han realizado las historias de usuario que se ven en la tabla 6.4 , que suman un total de 89 puntos.

El objetivo principal de este sprint es el de permitir a los usuarios solicitar consultas. Además se incluye la búsqueda de consultas pendientes de los médicos y pacientes, y también se permite a los administradores la gestión de los festivos de la clínica como también la gestión de las vacaciones de los empleados.

Este sprint es el más largo, debido a que también se reorganizan las consultas en el caso de que estas se vean afectadas al modificar el horario laboral de un médico, añadir vacaciones a este último o añadir algún festivo. Es por esto que la historia de usuario nº 12 tiene tantos puntos de historia.

6.4.2 Diseño

Backend

Entidades y tablas En el diagrama de entidades (ver figura 6.12) se muestran las entidades que han surgido en esta iteración y las relaciones entre ellas. Las principales entidades de este sprint son:

- Consultation: Representa una consulta.
- Holiday: Representa un día de vacaciones de los empleados.
- BankHoliday: Representa un festivo.
- MedicalFile: Representa un archivo que se adjunta a una consulta o a una prueba médica.

Servicios En este sprint se incluyeron en UserService todas las operaciones relacionadas con. Además ha surgido el servicio de las consultas (ConsultationService) donde se incluyeron las operaciones relacionadas con la creación, cancelación y modificación de fecha de consultas y la descarga de ficheros. Podemos ver la interfaz de ambos servicios con las operaciones que se incluyeron en este sprint en las figuras 6.13 y 6.14.

Excepciones A continuación se explican las excepciones que se han contemplado en la lógica de negocio de este sprint.

- ItsNotAnEmployeeException. Se utiliza para controlar que no se añaden vacaciones a un paciente.
- HourBusyException. Se utiliza para controlar que al crear o modificar la hora de una consulta, el horario no esté ocupado por otra.

Frontend

Al igual que en los otros sprints, se han realizado los mockups relacionados con las funcionalidades a implementar. En la figura 6.15 podemos ver el mockup de solicitar consulta.

6.4.3 Implementación

En este apartado se explica cómo, partiendo del diseño de la aplicación, se ha llevado a cabo la implementación de la misma con el apoyo de diversos frameworks y librerías.

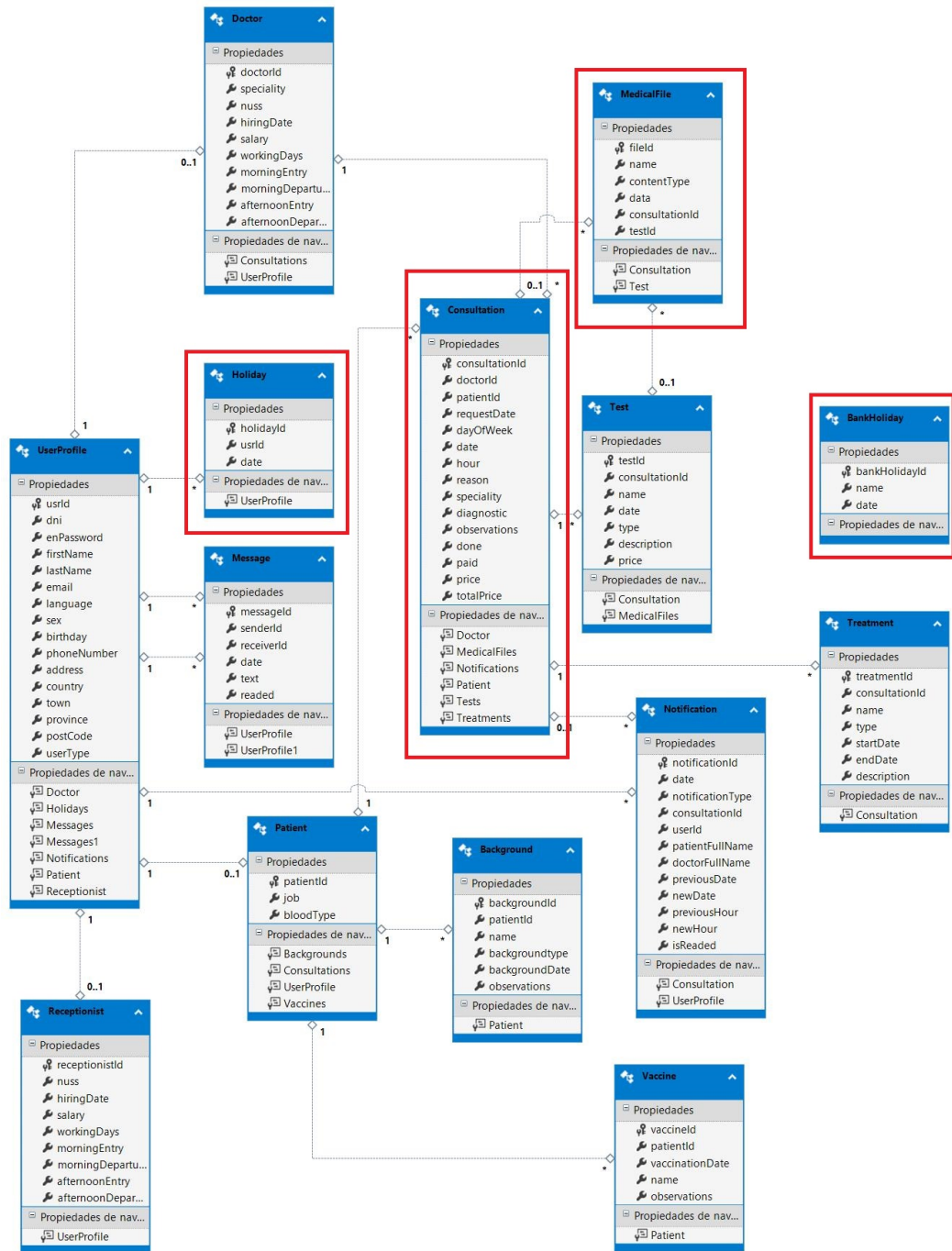


Figura 6.12: Diagrama de entidades del Sprint 3

```

<<Interface>>
UserService

CreateBankHoliday(name : String, bankHolidayDate : DateTime) : long
UpdateBankHolidayDetails(bankHolidayId : long, name : String, bankHolidayDate : DateTime) : void
RemoveBankHoliday(bankHolidayId : long) : void
FindBankHolidaysByMonthAndYear(month : int, year : int, startIndex : int, count : int) : Block<BankHoliday>
FindBankHolidayDetails(bankHolidayId : long) : BankHoliday
CreateHolidaysForEmployee(employeeId : long, holidayDateStart : DateTime, holidayDateEnd : DateTime) : List<long>
RemoveHoliday(holidayId : long) : void
FindHolidaysOfEmployeeByMonthAndYear(employeeId : long, int month, int year, startIndex : int, count : int) : Block<Holiday>

```

Figura 6.13: Funciones añadidas en UserService en el Sprint 3

```

<<Interface>>
ConsultationService

FindSpecialties() : List<string>
FindDoctorsBySpecialty(specialty : String) : List<UserProfile>
GetFreeHours(doctorId : long, date : DateTime) : List<TimeSpan>
CreateConsultation(doctorId : long, patientId : long, date : DateTime, hour : TimeSpan, reason : String) : long
AddMedicalFileToConsultation(consultationId : long, medicalFile : MedicalFile) : long
FindPendingPatientConsultationsByKeywordsAndDateAndSpecialty(patientId : long, keywords : String, date : DateTime?, specialty : String, startIndex : int, count : int) : Block<PendingPatientConsultationDTO>
FindPendingDoctorConsultationsByKeywordsAndDate(doctorId : long, keywords : String, date : DateTime?, startIndex : int, count : int) : Block<PendingDoctorConsultationDTO>
List<string> FindAllSpecialtiesOfPendingPatientConsultations(patientId : long) :
UpdateDateAndHourOfConsultation(consultationId : long, date : DateTime, hour : TimeSpan) : void
RemoveConsultation(long consultationId) : void
RecognizeConsultsOfDoctorByDate(doctorId : long, date : DateTime, notificationType : String) : void
RecognizeConsultsOfDoctorWhenUpdateHourlyWorkingDays(doctorId : long, daysAffected : String, morningEntry : TimeSpan?, morningEntryNew : TimeSpan?, morningDeparture : TimeSpan?,
morningDepartureNew : TimeSpan?, afternoonEntry : TimeSpan?, afternoonEntryNew : TimeSpan?, afternoonDeparture : TimeSpan?, afternoonDepartureNew : TimeSpan?) : void
FindMedicalFileOfConsultation(consultationId : long) : List<MedicalFileDTO>
FindConsultation(consultationId : long) : Consultation
FindMedicalFile(medicalFileId : long) : MedicalFile
RemoveMedicalFile(medicalFileId : long) : void

```

Figura 6.14: Funciones añadidas en ConsultationService en el Sprint 3

Paciente - Solicitar Consulta

Home Ver Mi Perfil Realizar una Consulta Ver Mensajes

Especialidad

Médico

12/12/2021

Hora	Selección
10:00	<input type="radio"/>
10:30	<input type="radio"/>

Motivo

Aceptar

Se muestran las horas disponibles para ese médico (es decir, aquellos que no incluyen festivos, vacaciones u horarios ocupados por otras consultas. En el caso de que no haya hora para seleccionar en la fecha indicada se mostrarán sugerencias

Figura 6.15: Mockup de solicitar consulta

Backend

En este apartado no se han realizado implementaciones a destacar a mayores de las ya explicadas en el sprint 1.

Frontend

Módulo Pages En este sprint, se incluyeron en el módulo User las siguientes páginas.

- SearchBankHolidays.aspx. Página que incluye las operaciones **CRUD** de los festivos.
- SearchHolidaysOfEmployee.aspx. Página que incluye las operaciones **CRUD** de las vacaciones de un empleado.

Además se creó el módulo Consultations donde se han incluido las páginas relacionadas con las funcionalidades proporcionadas por el ConsultationService. En este sprint se crearon las siguientes páginas.

- DownloadFile.aspx. Página para la descarga de archivos.
- RequestConsultation.aspx. Página que incluye un formulario para la solicitud de una consulta. (Ver figura 6.16)
- ShowConsultation.aspx. Página que muestra una consulta y todos los datos relacionados con esta misma. Incluye formularios para añadir ficheros, pruebas y tratamientos.

HealthFic Bárbara Home Solicitar consulta Mis mensajes Notificaciones

Back Solicitar consulta

Especialidad Alergología

Médico Maria Rodriguez Pereira

Fecha 19/02/2021

Hora 10:00

Motivo

Solicitar consulta

Copyright © Facultad de Informática - Universidade da Coruña

Figura 6.16: Solicitar consulta (RequestConsultation.aspx)

Subida y descarga de archivos Para la subida de archivos Ajax Control Toolkit nos proporciona AjaxFileUpload, un control que permite la subida de múltiples ficheros, además incluye la funcionalidad de arrastrar y soltar. Para su utilización se incluyeron una serie de parámetros en el archivo de configuración.

```

1 <system.web>
2   <httpHandlers>
3     <add verb="*" path="AjaxFileUploadHandler.axd"
4       type="AjaxControlToolkit.AjaxFileUploadHandler,
5       AjaxControlToolkit" />
6     <add path="ChartImg.axd" verb="GET,HEAD,POST"
7       type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler,
8       System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
9       PublicKeyToken=31BF3856AD364E35" validate="false"/>
10    </httpHandlers>
11  </system.web>
12
13 <system.webServer>
14   <validation validateIntegratedModeConfiguration="false"/>
15   <handlers>
16
17     <add name="ChartImageHandler" preCondition="integratedMode"
18       verb="GET,HEAD,POST" path="ChartImg.axd"
19       type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler,
20       System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
21       PublicKeyToken=31bf3856ad364e35"/>
22     <add name="AjaxFileUploadHandler" verb="*"
23       path="AjaxFileUploadHandler.axd"
24       type="AjaxControlToolkit.AjaxFileUploadHandler,
25       AjaxControlToolkit" />
26   </handlers>
27 </system.webServer>

```

Además la aplicación permite la descarga de archivos de las consultas y de las pruebas que están almacenados en la base de datos. Para su descarga se han utilizado los métodos proporcionados por la clase Response. Entre ellos destacamos:

- `OutputStream.Write`: método encargado de escribir el archivo pasándole como parámetros el array de bytes, la posición desde donde comienza a leer y la posición donde termina.
- `Response.Flush`: método que envía toda la información almacenada en el buffer al cliente.

6.5 Sprint 4

6.5.1 Análisis

ID	Historia de usuario	Puntos
15	Como médico, quiero poder añadir pruebas a una consulta	10
16	Como médico, quiero poder eliminar pruebas de una consulta	10
17	Como médico, quiero poder modificar las pruebas de una consulta	10
18	Como paciente, quiero ver mis pruebas	6
19	Como médico, quiero ver las pruebas de un paciente	1
20	Como médico, quiero poder añadir un diagnóstico a una consulta	3
33	Como médico, quiero poder añadir tratamientos a una consulta	5
34	Como médico, quiero poder eliminar tratamientos de una consulta	3
35	Como médico, quiero poder modificar los tratamientos de una consulta	5
36	Como paciente, quiero ver mis tratamientos	5
37	Como médico, quiero ver los tratamientos de un paciente	1
40	Como paciente o paciente, quiero ver mis consultas	8
42	Como médico, quiero poder ver las consultas de un paciente	8
		75

Tabla 6.5: Historias de usuario del Sprint 4

En este sprint se han realizado las historias de usuario que se ven en la tabla 6.5, que suman un total de 75 puntos.

El objetivo principal de este sprint es el de poder completar toda la información que tendrán las consultas (pruebas, tratamientos y diagnóstico). Además se completa la búsqueda de todos los datos clínicos que tiene un paciente, como también la búsqueda de todas las consultas de un médico.

6.5.2 Diseño

Backend

Entidades y tablas En el diagrama de entidades (ver figura 6.17) se muestran las entidades que han surgido en esta iteración y las relaciones entre ellas. Las principales entidades de este sprint son:

- Test: Representa una prueba médica que se le realiza a un paciente durante una consulta.
- Treatment: Representa un tratamiento que se le adjunta a una consulta.

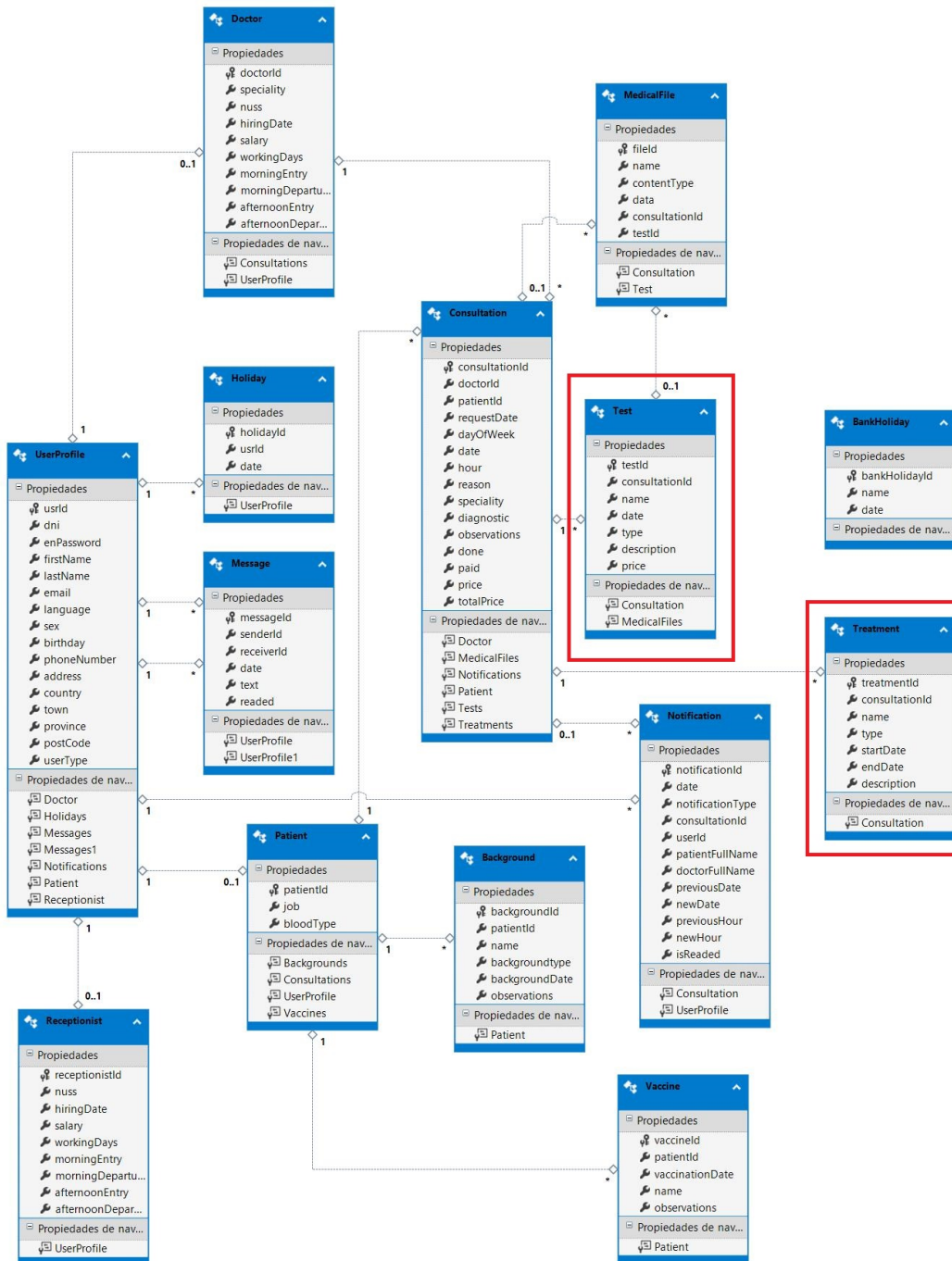


Figura 6.17: Diagrama de entidades del Sprint 4

Servicios En este sprint se incluyeron en ConsultationService las operaciones relacionadas con la búsqueda de consultas de un paciente o un médico y las operaciones **CRUD** de las pruebas y tratamientos médicos de un paciente. Podemos ver su interfaz con las operaciones que se incluyeron en este sprint en la figura 6.18.

```

<<interface>>
IConsultationService

CreateTest(consultationId : long, type : String, name : String, description : String) : long
FindTestsOfConsultation(consultationId : long) : List<TestDTO>
FindTestsOfPatientByTypeMonthAndYear(patientId : long, type : String, int month, int year, startIndex : int, count : int) : Block<TestDTO>
FindTest(testId : long) : Test
UpdateTest(testId : long, type : String, name : String, description : String) : void
RemoveTest(testId : long) : void
AddMedicalFileToTest(testId : long, medicalFile : MedicalFile) : long
FindMedicalFileOfTest(testId : long) : List<MedicalFileDTO>
FindTestTypesOfPatient(patientId : long) : List<String>
CreateTreatment(consultationId : long, type : String, name : String, startDate : DateTime, endDate : DateTime, description : String) : long
FindTreatmentsOfConsultation(consultationId : long) : List<TreatmentDTO>
FindTreatmentsOfPatientByTypeMonthAndYear(patientId : long, type : String, month : int, year : int, startIndex : int, count : int) : Block<TreatmentDTO>
FindActiveTreatmentsOfPatientByType(patientId : long, type : String, startIndex : int, count : int) : Block<TreatmentDTO>
FindTreatment(treatmentId : long) : Treatment
UpdateTreatment(treatmentId : long, type : String, name : String, startDate : DateTime, endDate : DateTime, description : String) : void
RemoveTreatment(treatmentId : long) : void
FindTreatmentTypesOfPatient(long patientId) : List<String>
AddDiagnosticToConsultation(consultationId : long, String diagnostic, String observations) : void
EndConsultation(long consultationId) : void
FindPatientConsultationsByKeywordsMonthYearAndSpecialty(patientId : long, keywords : String, String specialty : String, month : int, year : int, done : int, startIndex : int, count : int) : Block<PatientConsultationDTO>
FindDoctorConsultationsByKeywordsMonthYearAndDone(long doctorId, keywords : String, month : int, year : int, done : int, startIndex : int, count : int) : Block<DoctorConsultationDTO>
    
```

Figura 6.18: Funciones añadidas en ConsultationService en el Sprint 4

Paginación Se ha descartado la paginación en las listas de archivos médicos, pruebas y tratamientos en una consulta debido a que se ha considerado que nunca serán demasiadas al ser solo de una misma consulta.

Frontend

Al igual que los anteriores sprints, se han realizado los mockups relacionados con las funcionalidades a implementar. En la figura 6.19 podemos ver el mockup de ver la información de una consulta.

6.5.3 Implementación

En este apartado se explica cómo, partiendo del diseño de la aplicación, se ha llevado a cabo la implementación de la misma con el apoyo de diversos frameworks y librerías.

Backend

En este apartado no se han realizado implementaciones a destacar a mayores de las ya explicadas en el sprint 1.

Frontend

Módulo Pages En este sprint, se añadieron las funcionalidades relacionadas con las pruebas y tratamientos en la página ShowConsultation.aspx (ver figura 6.20) y se incluyeron en el módulo Consultations las siguientes páginas:

A Web Page

http://

Médico - Ver Consulta

Home Ver Mi Perfil Ver Consultas Buscar Pacientes Ver Mensajes

Lara Perez Díaz Tipo de Sangre: A+ Ver Antecedentes Ver Vacunas Ver Consultas
Sexo: Mujer DNI: 12341234A

Médico: Juan Rodríguez Fernández Pediatría

Consulta: Pendiente Pagada: No 25/02/2021 - 11:00 Cambiar fecha Se abrirá un formulario para modificar la fecha
Cancelar Consulta

Motivo de la consulta: Dolor Garganta

Documentos Adjuntos: Añadir Documento
[parte de accidente.pdf](#)

Pruebas: Añadir prueba Ver pruebas paciente

Se mostrara la lista de las pruebas realizadas en la consulta

Tipo	Nombre	Fecha	Ver información
Radiografía	Radiografía Garganta	24/02/2021	Mostrará un pop up con los datos de la prueba.

Tratamiento: Añadir tratamiento Ver tratamientos paciente

Se mostrara la lista de los tratamientos añadidos

Tipo	Nombre	Fecha	Ver información
Medicamento	Ibuprofeno	24/02/2021	Mostrará un pop up con los datos del tratamiento

Diagnóstico:

Diagnóstico

Observaciones

Generará un PDF con los datos del paciente, del médico, el motivo de la consulta, diagnóstico y el tratamiento.

Generar PDF Ver factura Generar Justificante de asistencia

Mostrará un pop up con los precios de las consultas y pruebas

Finalizar

Figura 6.19: Mockup de ver una consulta

- FindDoctorConsultation.aspx. Página que muestra las consultas de un doctor.
- FindPatientConsultations.aspx. Página que muestra las consultas de un paciente.
- SearchTestOfPatient.aspx. Página que muestra las pruebas de un paciente
- SearchTreatmentsOfPatient.aspx. Página que muestra los tratamientos de un paciente.

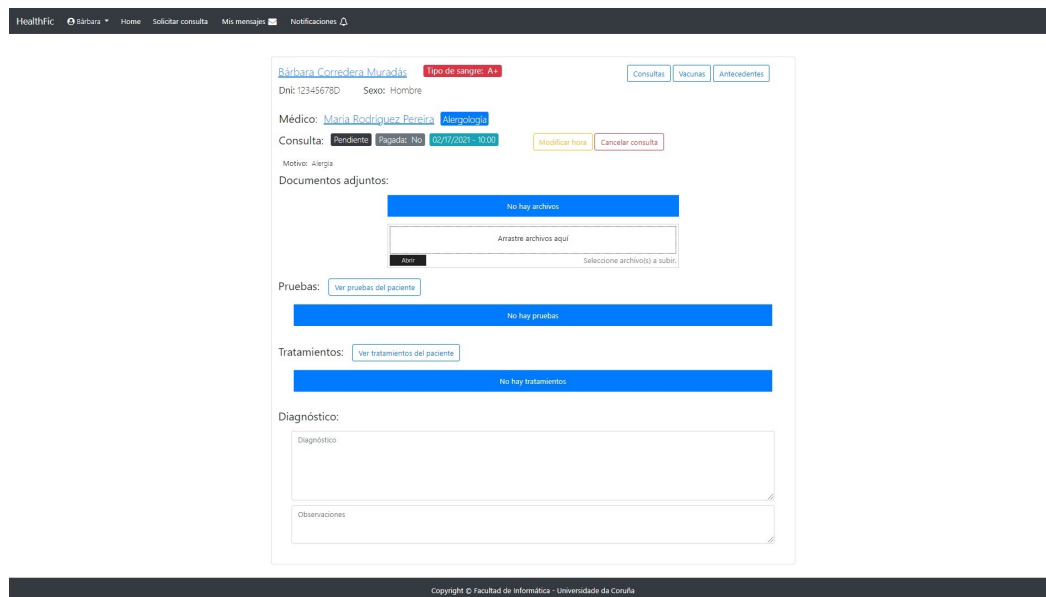


Figura 6.20: Ver consulta (ShowConsultation.aspx)

6.6 Sprint 5

6.6.1 Análisis

ID	Historia de usuario	Puntos
38	Como médico, paciente o recepcionista, quiero poder exportar a PDF los datos de una consulta	30
39	Como médico, quiero poder generar un justificante de asistencia	2
48	Como recepcionista, quiero poder cobrar las consultas	20
52	Como recepcionista, quiero poder ver las consultas pendientes de pago	13
		65

Tabla 6.6: Historias de usuario del Sprint 5

En este sprint se han realizado las historias de usuario que se ven en la tabla 6.6, que suman un total de 65 puntos.

El objetivo principal de este sprint es el cobro de consultas y la exportación a PDF de los datos de estas mismas.

6.6.2 Diseño

Backend

Entidades y tablas En este sprint no se crea ninguna entidad nueva.

Servicios

En este sprint se incluyeron en ConsultationService las operaciones relacionadas con el cobro de consultas y la búsqueda de consultas pendientes de pago. Podemos ver su interfaz con las operaciones que se incluyeron en este sprint en la figura 6.21.

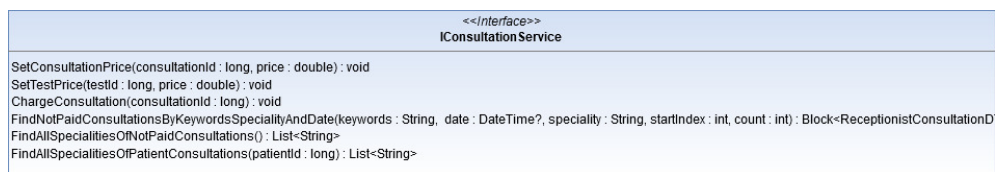


Figura 6.21: Funciones añadidas en ConsultationService en el Sprint 5

Frontend

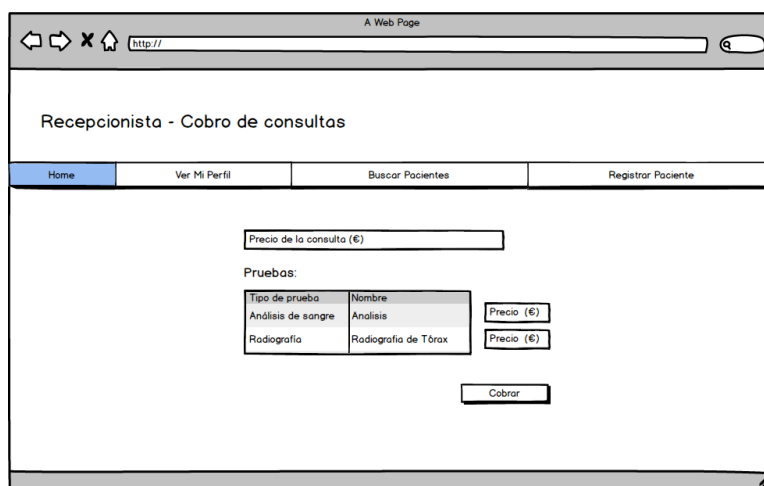


Figura 6.22: Mockup del cobro de consultas

Al igual que los anteriores sprints, se han realizado los mockups relacionados con las funcionalidades a implementar. En la figura 6.22 podemos ver el mockup de menú principal de los recepcionistas.

6.6.3 Implementación

En este apartado se explica cómo, partiendo del diseño de la aplicación, se ha llevado a cabo la implementación de la misma con el apoyo de diversos frameworks y librerías.

Backend

En este apartado no se han realizado implementaciones a destacar a mayores de las ya explicadas en el sprint 1.

Frontend

Módulo Pages No se ha añadido ninguna página nueva, para implementar las funcionalidades de este sprint se han utilizado los formularios ya existentes. Podemos ver en la figura 6.23 el cobro de consultas.

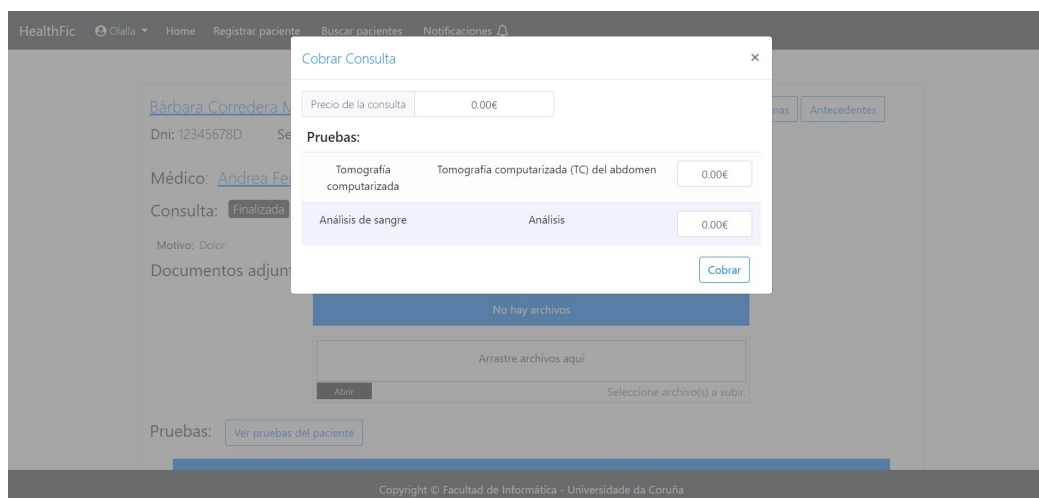


Figura 6.23: Cobro de consultas (ShowConsultation.aspx)

Exportación a PDF Para la exportación de los datos a PDF se ha utilizado la librería IText7 que nos proporciona diferentes clases para su creación. Entre ellas destacamos:

- PdfWriter: clase utilizada para la creación de un flujo de salida.
- PdfDocument: clase que se encarga de abrir un archivo PDF en modo escritura.

- Document: clase utilizada para crear un documento a partir de un PDF especificando un tamaño de página, en nuestro caso A4.
- PdfMerger: clase encargada de unir PDF. Se ha utilizado para unir los archivos adjuntos en las consultas y pruebas.
- ImageDataFactory: clase encargada de crear una instancia de una imagen a partir de un array de datos.

FACTURA

Nº Factura	Fecha factura	Cliente
3	31/01/2021	Mireia Marti Vilas Nº Teléfono: 988123456 Plaza Pontevedra N°1 5°B A Coruña, A Coruña, España Código Postal: 15001
DNI Cliente	Especialidad médica	
12345678L	Medicina familiar	

#	Tipo	Descripción	Fecha	Importe
1	Medicina familiar	Consulta médica	01/02/2021	50 €
2	Tomografía computerizada	Tomografía computerizada (TC) del abdomen y pelvis con contraste	30/01/2021	149,99 €
3	Análisis de orina	Análisis de orina	30/01/2021	30 €
Precio Total				229,99 €

Figura 6.24: Factura



COLEGIO OFICIAL DE
Médicos de Navarra



ORGANIZACIÓN
MÉDICA COLEGIAL
DE ESPAÑA



CONSEJO GENERAL
DE COLEGIOS OFICIALES
DE MÉDICOS

Nº de Receta: 31-552317

RECETA MÉDICA PARA ASISTENCIA SANITARIA PRIVADA			
RECETA MÉDICA PRIVADA	PRESCRIPCIÓN Consignar el medicamento: DCI o marca, forma farmacéutica, vía de administración, dosis por unidad y unidades por envase. (Datos correspondientes en caso de producto sanitario). Núm. envases / unidades: <input type="checkbox"/>	Duración del tratamiento Posología Unidades Pauta <input type="checkbox"/>	PACIENTE (Nombre, apellidos, año de nacimiento y nº de DNI / NIE / pasaporte) PRESCRIPTOR (datos de identificación y firma) Dr. _____ Núm. Colegiado: 31-_____ Medicina general Email: _____ Tfno/Fax: +34948-_____ Fecha de la prescripción * ____/____/____
	Información al Farmacéutico, en su caso	Motivo de la sustitución <input type="checkbox"/> Urgencia <input type="checkbox"/> Desabastecimiento	FARMACIA (NIF/CIF, datos de identificación y fecha de dispensación)
	<small>* La validez de esta receta expira a los 10 días naturales de la fecha prevista para la dispensación, o en su defecto de la fecha de prescripción. La medicación prescrita no superará los 3 meses de tratamiento. La receta es válida para una única dispensación en la farmacia. Esta receta podrá ser dispensada en cualquier oficina de farmacia del territorio nacional.</small>	 ZwCdDcMPvPRVuAVp6U/b5FTb658=	<small>Es cumplimiento de lo establecido en el artículo 9 de la Ley Orgánica 13/1985, de 27 de junio, de los Colegios Profesionales de España, así como en un fichero de la farmacia dispensadora para dar cumplimiento a la obligación establecida en el RD 1718/2010. Se le informa que podrá ejercer sus derechos de acceso, rectificación, cancelación y oposición en la dirección del prescriptor por la farmacia dispensadora.</small>

Figura 6.25: Receta médica

Además se ha utilizado la clase `MemoryStream`, en la que se crea un flujo de datos que usa a la memoria RAM como almacenamiento de apoyo en vez del disco duro o red. La clase

MemoryStream encapsula la información almacenada como un array de bytes sin signo que se inicializa después de la creación del objeto MemoryStream. La información encapsulada se accede directamente en la memoria RAM. Estos streams de memoria pueden reducir la necesidad del uso de Buffers temporales y archivos en la aplicación.

En las figuras 6.24 y 6.25 se pueden ver una factura y una receta médica creada por la aplicación. Se han incluido el apéndice B la exportación de los datos de una consulta.

6.7 Sprint 6

6.7.1 Análisis

ID	Historia de usuario	Puntos
41	Como paciente, recepcionista o médico, quiero ver sugerencias en el momento de solicitar una consulta	10
45	Como médico o paciente, quiero poder enviar mensajes	10
46	Como médico o paciente, quiero poder ver mis mensajes	10
51	Como paciente, médico o recepcionista, quiero poder recibir avisos sobre las consultas	15
53	Como administrador, quiero poder ver gráficos de las ganancias de consultas	10
		55

Tabla 6.7: Historias de usuario del Sprint 6

En este sprint se han realizado las historias de usuario que se ven en la tabla 6.7, que suman un total de 55 puntos.

El objetivo principal de este sprint es gestión de mensajería de la aplicación, tanto mensajes entre usuarios como avisos sobre la modificación del estado de una consulta (creación, cancelación o modificación de hora). Además se incluirán diversos gráficos sobre las ganancias de la clínica que podrán ver los administradores. Este sprint es de los más cortos por lo explicado anteriormente, ya que en este se dedicará más tiempo a posibles correcciones y finalización de la memoria.

6.7.2 Diseño

Backend

Entidades y tablas En el diagrama de entidades (ver figura 6.26) se muestran las entidades que se han creado y sus relaciones. Las principales entidades de este sprint son:

- Message: Representa un mensaje.
- Notification: Representa los avisos que recibirán los usuarios.

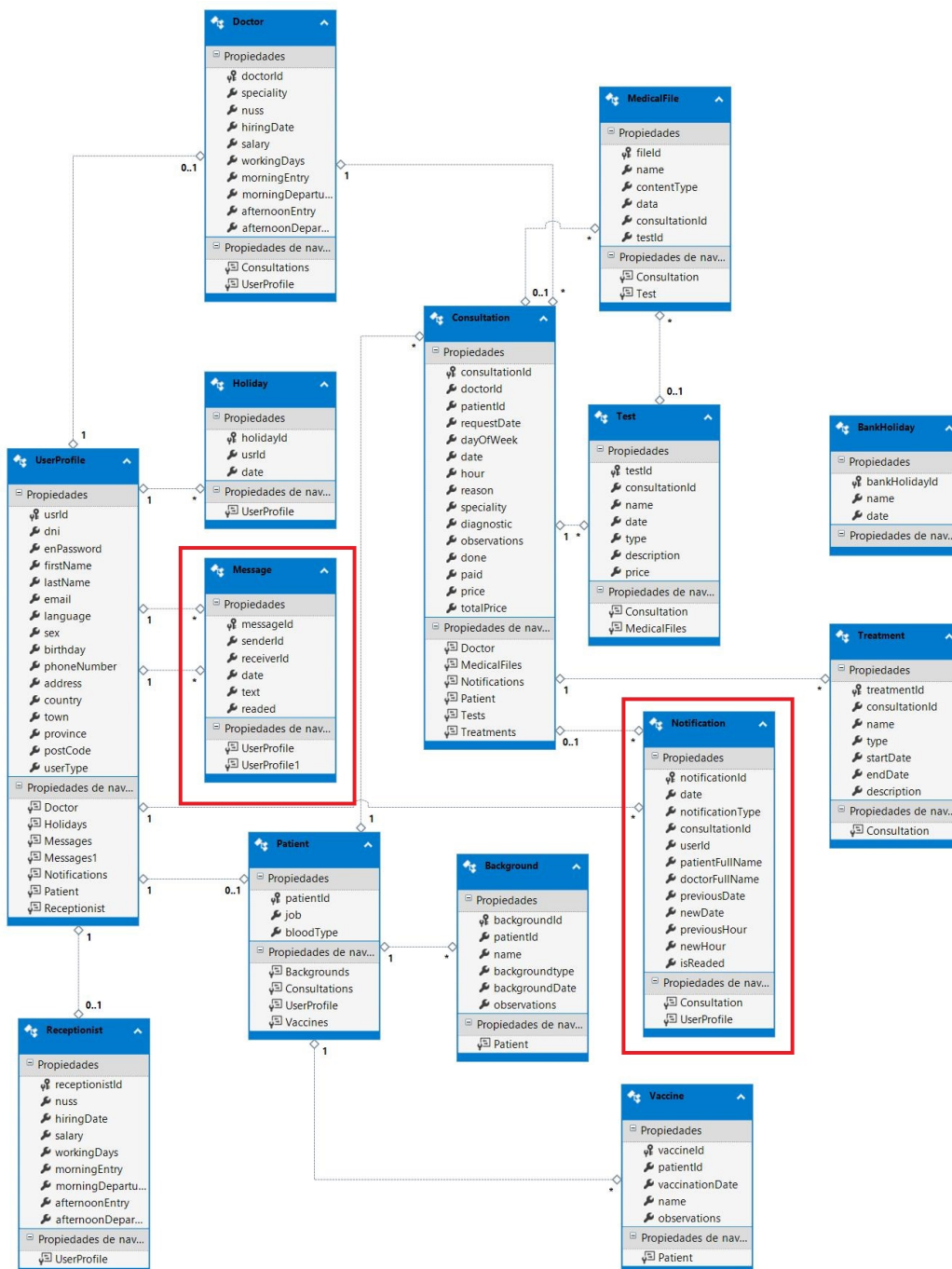


Figura 6.26: Diagrama de entidades del Sprint 6

Servicios En este sprint se incluyeron las operaciones relacionadas con el envío de mensajes, con los avisos y la muestra de estadísticas, para ello se creó un nuevo servicio llamado `InformationService`. Además en `IConsultationService` se incluyó las operaciones relacionadas con las recomendaciones de consultas. Podemos ver su interfaz con las operaciones que se incluyeron en este sprint en la figura 6.27.

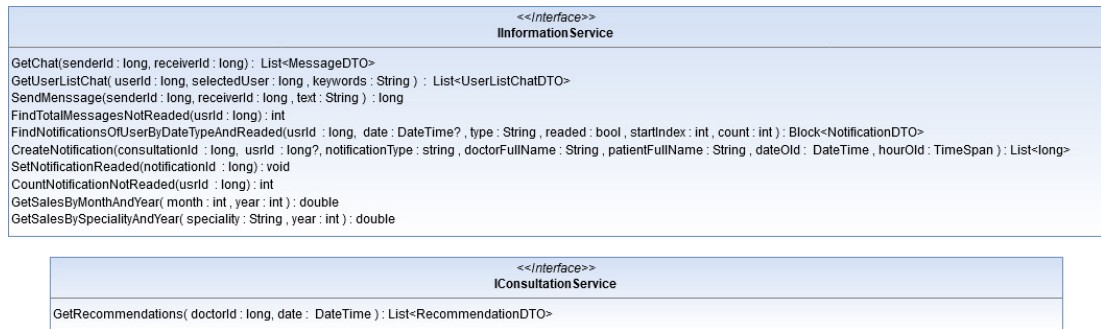


Figura 6.27: Funciones añadidas en `InformationService` y `ConsultationService` en el Sprint 6

Frontend

Al igual que los anteriores sprints, se han realizado los mockups relacionados con las funcionalidades a implementar. En la figura 6.28 podemos ver el mockup de ver los chats de un usuario.

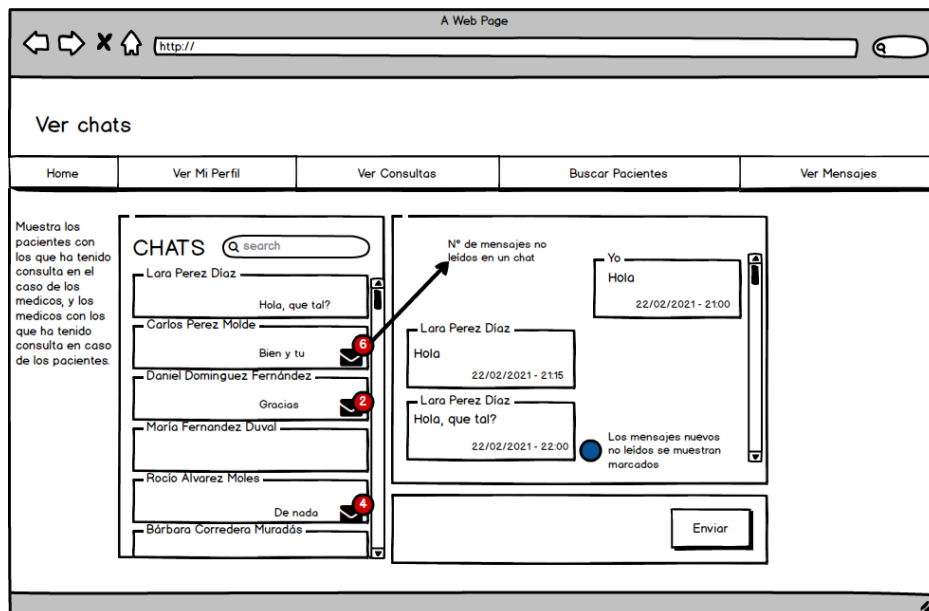


Figura 6.28: Mockup de ver los chats de un usuario

6.7.3 Implementación

Backend

En este apartado no se han realizado implementaciones a destacar a mayores de las ya explicadas en el sprint 1.

Frontend

En este sprint, se creó el módulo Information donde se incluyeron las siguientes páginas.

- Chat.aspx. Página que incluye los chats de los médicos o pacientes. (Ver figura 6.29)
- ShowNotifications.aspx. Página donde se verán las notificaciones de los usuarios.
- Statistics.aspx. Página donde se mostrarán gráficos de las ventas de la clínica.

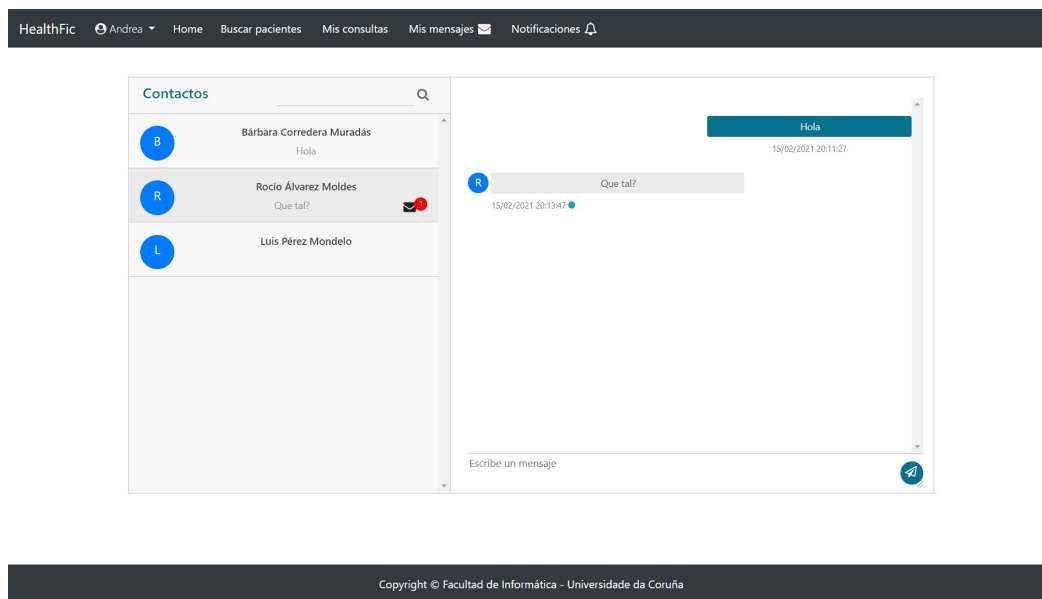


Figura 6.29: Chat (Chat.aspx)

Pruebas

PARA probar el correcto funcionamiento de las distintas partes de la aplicación y del conjunto de la misma, se ha creado el proyecto Test donde se han realizado distintos tipos de pruebas: unitarias, de integración y funcionales, que se describen a continuación.

7.1 Pruebas unitarias

Se han realizado pruebas unitarias para los métodos de lógica de negocio con los que cuentan las entidades, mencionados anteriormente. No se han realizado pruebas unitarias a los **DAO** que acceden a la **BD**, ya que se trata de código que se prueba de forma indirecta en las pruebas de integración.

7.2 Pruebas integración

Se ha utilizado TestTools proporcionado por Visual Studio para implementar las pruebas de integración de cada uno de los servicios. Como se ha mencionado, también prueban de forma indirecta el funcionamiento de los **DAO** y las entidades.

Por cada servicio, se ha creado una clase de prueba anotada con [TestClass], en la que se incluirán diferentes métodos anotados con [TestMethod] para probar cada funcionalidad. Estos test siguen un ciclo de vida:

- Al inicio de la ejecución de pruebas.
 - Creación de instancia. Unit test runner.
 - El test runner crea una sesión de test y obtiene los atributos de test.
 - Se crea o recupera un test context, que existe durante la totalidad de la sesión de test.

- Se llaman a los métodos de inicialización del assembly y de la clase: `AssemblyInitialize` y a `ClassInitialize` donde se inyectarán las dependencias necesarias para ejecutar los tests.
- Por cada método de test.
 - Se recupera una instancia de la sesión de test.
 - Se llama al método de inicialización, aquel con el atributo `TestInitialize`.
 - Se invoca el método que se desea probar.
 - Se llama al método `testCleanup`, aquel con el atributo `TestCleanup`.
- Una vez ejecutados todos los tests.
 - Se llama al método `classCleanup`, aquel con el atributo `ClassCleanup`.
 - Se llama al método `assemblyCleanup`, aquel con el atributo `AssemblyCleanup`.

Además se ha utilizado la clase `TransactionScope` para crear bloques de código transaccional. De esta manera, cada método acabará con un `rollback`; es decir, las modificaciones que haya hecho ese caso de prueba en la base de datos se deshacen automáticamente. De esta forma:

- Cada caso de prueba se ejecuta en una transacción.
- Se garantiza que los casos de prueba sean independientes entre sí.
- No es necesario eliminar los datos creados.

```

1      [TestMethod]
2      [ExpectedException(typeof(IncorrectPasswordException))]
3      public void ChangePasswordWithIncorrectPasswordTest()
4      {
5          using (var scope = new TransactionScope())
6          {
7              // Register user
8              var userId = userService.RegisterPatient(dni,
9              clearPassword, new PatientProfileDetails(dni, firstName,
10             lastName, email, language, sex, birthday, address,
11             phoneNumber, country, town, province,
12             postCode, job, bloodType));
13             // Change password
14             var newClearPassword = clearPassword + "X";
15             userService.ChangePassword(userId, clearPassword +
16             "Y", newClearPassword);
17         }
18     }

```

7.3 Pruebas funcionales

El objetivo de este tipo de pruebas es verificar el correcto funcionamiento de la aplicación en su conjunto, utilizando directamente la interfaz como haría el usuario final. Normalmente, cada caso de prueba se corresponde con una secuencia de navegación del usuario, que es una funcionalidad concreta de la aplicación.

Estas pruebas se han realizado de forma manual. Para ello, se han probado todos los casos de uso de la aplicación y sus correspondientes excepciones. En la medida de lo posible, se ha intentado:

- Probar varias veces un mismo caso de uso, siguiendo la misma secuencia o distinta secuencia.
- Probar varios casos de uso en órdenes distintas.

Además he tenido la posibilidad de que personal médico pruebe la aplicación, de esta manera he podido comprobar y tener una mejor idea de cómo se utilizará la aplicación día a día.

Funcionalidades más destacadas

EN este capítulo se explican las principales funcionalidades de la aplicación desde el punto de vista del usuario.

8.1 Solicitud de consultas

El usuario elige una especialidad, un médico y una fecha determinada. En el caso de que no haya horarios disponibles en ese determinado día, el desplegable de la hora no se habilitará y se mostrarán diferentes sugerencias:

- En el caso de que haya más médicos en la especialidad seleccionada, nos mostrará por cada uno de los médicos, una sugerencia en el caso que tengan disponible para el mismo día, otra para un día anterior y otra para el día siguiente.
- En el caso de que no haya más médicos mostrará el próximo y el anterior día con un horario disponible.

Una vez que el usuario seleccione un horario disponible y confirme la consulta, se mostrará la opción de poder adjuntar ficheros en dicha consulta.

La solicitud de consultas está habilitada para pacientes, recepcionistas y médicos.

8.2 Búsqueda de consultas pendientes

El usuario autenticado como médico o paciente, en su página principal se mostrará una tabla con las consultas pendientes de ese día, teniendo la opción de buscar las consultas para otras fechas. Además para mayor comodidad, se incluyen otra serie de filtros:

- En el caso de pacientes, por especialidad y keywords contenidos en el motivo de la consulta o en el dni o nombre del médico.

- En el caso de los médicos, por keywords contenidos en el motivo de la consulta o en el dni o nombre del paciente.

8.3 Reorganización de fecha de consultas y avisos

Cuando se añade un festivo, se modifica el horario laboral de un médico o incluso si se añaden vacaciones a este mismo, es posible que algunas consultas queden afectadas. Por eso, por mayor comodidad se creó un algoritmo que modifica la fecha de dichas consultas a otra válida. Dependiendo del caso se reorganiza de una manera u otra:

- Para el caso de los festivos y vacaciones, la consulta pasará a la fecha posterior válida más cercana a la fecha que tenía.
- Para el otro caso, la fecha de la consulta será modificada a una nueva válida más cercana al día de hoy.

Además cada vez que se crea una consulta, se modifica su fecha de una consulta o se cancele la aplicación generará una serie de avisos que se verán en la página web avisando de dichos cambios.

8.4 Gestión de los datos clínicos del paciente

El usuario selecciona el perfil de un paciente (su perfil en el caso de un paciente), donde verá una serie de botones donde están sus datos médicos. Cada uno de ellos redirigirá a una nueva página con los datos correspondiente:

- Antecedentes médicos: Se listan los antecedentes médicos con los siguientes filtros:
 - Tipo de antecedente.
 - Palabras clave contenidas en el nombre.
- Vacunas: Se listan los antecedentes médicos con los siguientes filtros:
 - Palabras clave contenidas en el nombre.
- Consultas: Se listan los antecedentes médicos con los siguientes filtros:
 - Especialidad.
 - Año y mes.
 - Palabras clave contenidas en el nombre o dni del médico, o en el motivo.
 - Estado, si está pendiente o finalizada.

- Tratamientos: Se listan los antecedentes médicos con los siguientes filtros:
 - Tipo de tratamiento.
 - Mes y año.
 - Estado, si está activo o no.
- Pruebas: Se listan los antecedentes médicos con los siguientes filtros:
 - Tipo de prueba.
 - Mes y año.

Además de eso el médico podrá modificar o eliminar dichos datos. Los otros usuarios solo tendrán permiso de visualización de todos los datos de estos mismos. En el caso de las consultas se redirigirá a una página con todos sus datos, en los otros se mostrará un popup donde se verá su información.

8.5 Exportación de los datos de una consulta

Para mayor portabilidad de los datos el usuario tendrá la posibilidad de exportación a PDF. Para ello, el usuario selecciona una consulta, y podrá ver diferentes botones para exportar los datos de esta misma:

- Exportación de datos de toda la consulta. Incluye:
 - Datos del paciente y sus antecedentes médicos.
 - Datos del médico.
 - Datos de la consulta.
 - Archivos adjuntos de la consulta.
 - Pruebas y sus archivos adjuntos.
 - Tratamientos.
 - Diagnóstico.
- Receta médica para un medicamento.
- Factura.

8.6 Cobro de consultas

El recepcionista tendrá en su página principal aquellas consultas que están pendientes de cobro. Al elegir la consulta, seleccionará el botón de cobrar y se le abrirá un formulario con la lista en la que incluirá la consulta y cada una de sus pruebas con la opción de establecer un precio a cada uno de estos elementos.

Finalmente, una vez que cubra el formulario, se generará una factura con los datos introducidos.

Conclusiones

TRAS la realización del proyecto, se ha realizado un pequeño análisis en el que se contrastan los objetivos planteados inicialmente con lo realizado en el proyecto. Se incluyen también las lecciones aprendidas durante el desarrollo y las posibles líneas futuras de la aplicación.

9.1 Concordancia de objetivos

A continuación se citan los objetivos iniciales de la aplicación, y se comenta en qué grado y por qué motivos se considera que se ha alcanzado cada uno de ellos.

- Afianzar los conocimientos con las tecnologías empleadas. Utilizando de nuevo tecnologías ya empleadas previamente, se ha conseguido una mayor comprensión del funcionamiento de las mismas. A lo largo del proyecto se ha adquirido práctica y familiaridad con las mismas, proporcionando una mayor facilidad, flexibilidad y rapidez a la hora de programar. Además, al adquirir un conocimiento en mayor profundidad, nos permitirá en un futuro aprender de forma más rápida tecnologías distintas.
- Realizar una buena planificación y seguirla durante el desarrollo. Este aspecto ha sido fundamental, debido a que al ser un proyecto de larga duración en comparación con los realizados hasta el momento, tener fechas de entrega intermedias ha contribuido en gran medida a llevar un ritmo de desarrollo más o menos constante.
- Desarrollar una aplicación funcional. Se ha conseguido implementar todas las funcionalidades principales planteadas en el enunciado. Sin embargo, al ser un ejemplo académico, se podría mejorar algunos de sus aspectos técnicos para optimizar su utilización en un contexto real. Esto se debe, principalmente, a que no hay un cliente especializado en el contexto, aunque se ha intentado corregir hablando con diferentes personas que

trabajan en el sector, puede que no sea totalmente suficiente para cumplir todos los aspectos técnicos.

- Hacer un buen diseño de la aplicación. A lo largo de la carrera se han estudiado diferentes patrones de diseño que han sido aplicados durante el desarrollo de la aplicación. Sin embargo, durante la implementación se han detectado algunos aspectos del diseño que se podían mejorar y que se han corregido. Al emplear una metodología ágil, estas correcciones no han tenido gran repercusión en la planificación, pero sí han servido para adquirir más conocimientos, especialmente orientado a la práctica, sobre las técnicas de diseño empleadas.

9.2 Lecciones aprendidas

A continuación se explican las principales lecciones aprendidas durante el desarrollo del proyecto.

- Importancia de la planificación. La elección de la metodología se consideraba inicialmente un aspecto poco relevante, pero ciertamente condiciona el desarrollo del proyecto. La división del proyecto en múltiples entregas ha permitido que el desarrollo sea constante y cumplir el plazo de finalización establecido.
- Como el proyecto se ha realizado de forma individual, a diferencia de la gran mayoría de los realizados hasta el momento, me ha obligado a conocer cada pequeño detalle. Esto se ha traducido en una mayor comprensión del mismo.
- Durante la carrera nos han insistido en gran cantidad de ocasiones en que el cliente no tiene claro lo que quiere. Aunque he realizado muchos trabajos, siempre había un enunciado al que ceñirse, o una persona a la que preguntarle en caso de duda. En esta ocasión, yo he puesto el enunciado y en gran cantidad de ocasiones no tenía claro lo que quería. Por lo tanto, he experimentado lo que significa ser el cliente y no saber lo que quiere.
- Como he mencionado, siempre he tenido interés en el sector sanitario y al realizar un proyecto relacionado con este ámbito me ha permitido tener un mayor conocimiento del mismo.

9.3 Líneas futuras

9.3.1 Integración con un visor DICOM

DICOM es un protocolo estándar de comunicación ideado por el Colegio Americano de Radiología para el intercambio de las imágenes médicas obtenidas en los estudios radiológicos. Un archivo **DICOM** consiste en un encabezado y conjuntos de datos de imagen empaquetados en un solo archivo, donde las informaciones del paciente se organizan en una serie constante y estandarizada de encabezados. Desafortunadamente, a diferencia de otros formatos de archivo de imagen, como **JPEG**, Windows y Mac no reconocen los archivos **DICOM** como archivos de imagen. Para ver estas imágenes en su computadora, deberá usar un visor **DICOM**, que interpretará las informaciones del archivo y lo mostrará como una imagen. Debido a la necesidad de utilizar un visor para ver estas imágenes, y por simplicidad, la aplicación no da soporte a este tipo de archivos.

Por lo tanto, existe el protocolo para poder compartir esas imágenes médicas (**DICOM**) y, por otro lado, existe el software que posibilita su almacenamiento, impresión y manipulación: el visor **DICOM**.

La utilización tanto de este protocolo, como de los visores proporciona las siguientes características:

- El intercambio de las imágenes.
- Proporciona un almacenamiento más sencillo.
- Ahorro de costes en todos los procesos.
- Ayuda en el diagnóstico
- Mejora del seguimiento de diferentes enfermedades o dolencias (cáncer, problemas cardiovasculares...).

Debido a todas estas ventajas y que los especialistas trabajan con volúmenes amplios de información requiriendo cada vez más de software de lectura que se ajuste a sus necesidades específicas, es interesante la integración y soporte de un visor **DICOM** en la aplicación web.

9.3.2 Cliente SMTP

La aplicación envía una serie de avisos cuando se realiza una consulta, cuando se modifica su horario o cuando se cancela. Estos avisos solo se ven dentro de la aplicación, por lo que en el caso de alguna modificación habría que estar atento y entrar en la página web frecuentemente. Para solucionar esto, sería interesante la inclusión de un cliente **SMTP** en la aplicación que se encargue de mandar correos electrónicos avisando de estas acciones a todos los involucrados.

9.3.3 Optimización en el horario laboral de los médicos

La aplicación permite establecer el horario laboral de un médico de manera que se puede escoger tanto los días que trabaja como su horario de mañana y de tarde, pero en una clínica existe la posibilidad que alguno de los especialistas solo trabajen en periodos específicos, es decir, que vengan un día cada dos semanas, cada mes, etc.

Para simular estos casos en la aplicación, se puede añadir vacaciones del empleado durante los periodos en los que no trabaja, pero esto hace una carga de trabajo importante para el administrador. Por lo tanto, para mejorar la experiencia de usuario, sería interesante poder establecer un horario laboral basado en periodos de tiempo, evitando así la pérdida de tiempo que supone la anterior solución.

9.3.4 Autocompletado

Para mejorar la experiencia del usuario y la velocidad con la que se realizan ciertas acciones, es interesante implementar el autocompletado en campos de algunos formularios, como sería el caso de los nombres de los tratamientos, pruebas, etc.

Apéndices

Apéndice A

Manual de usuario

1. Introducción	3
2. Entrada al sistema	3
2.1 Registro de pacientes	3
2.2 Registro de empleados	3
2.3 Autenticación	4
2.4 Modificación de perfil	5
2.4 Modificación de la contraseña	5
2.5 Cerrar sesión	5
3. Funcionalidades	5
3.1 Barras de navegación	5
3.1.1 Paciente	6
3.1.2 Médico	6
3.1.3 Recepcionista	6
3.1.4 Administrador	6
3.2 Menú principal	6
3.2.1 Paciente	6
3.2.2 Médico	7
3.2.3 Recepcionista	7
3.3 Perfiles	8
3.3.1 Paciente	8
3.3.2 Empleados	8
3.4 Buscar pacientes	9
3.5 Buscar empleados	9
3.5 Solicitud de consulta	10
3.6 Ver consulta	11
3.6.1 Añadir prueba	12
3.6.2 Añadir tratamiento	12
3.6.3 Cobro de consultas	12
3.6.4 Otras operaciones	13
3.7 Ver tratamientos del paciente	14
3.8 Ver pruebas del paciente	14
3.9 Ver consultas del paciente	15
3.10 Ver consultas del médico	15
3.11 Antecedentes	16
3.11.1 Ver antecedentes de un paciente	16
3.11.2 Añadir antecedente	16

3.11.3 Otras operaciones	17
3.12 Vacunas	17
3.12.1 Ver vacunas de un paciente	17
3.12.2 Añadir vacuna	17
3.12.3 Otras operaciones	18
3.13 Notificaciones	18
3.14 Mis mensajes	18
3.15 Vacaciones	19
3.15.1 Ver vacaciones de un empleado	19
3.15.2 Añadir vacaciones	19
3.16.3 Otras operaciones	20
3.17 Festivos	20
3.17.1 Ver festivos	20
3.17.2 Añadir Festivo	21
3.17.3 Otras operaciones	21
3.18 Estadísticas	22

1.Introducción

La aplicación de gestión de consultas HealthFic, está pensada y diseñada para facilitar y mejorar la experiencia de los usuarios de una clínica. Actualmente, existen cuatro roles para acceder a la aplicación: Paciente, Médico, Recepcionista y Administrador.

2. Entrada al sistema

2.1 Registro de pacientes

A continuación se muestra el formulario para que un usuario no autenticado se registre en la aplicación como paciente.

The screenshot shows the 'Registro de pacientes' (Patient Registration) form. At the top, there is a navigation bar with 'HealthFic', 'Home', and 'Autenticación'. The form itself has a blue header with a 'Volver' button and the title 'Registro de pacientes'. The form fields are as follows:

- Documento de identidad (text input)
- Contraseña (text input) and Repita la contraseña (text input)
- Nombre (text input), Primer apellido (text input), Segundo apellido (text input)
- Sexo (dropdown menu, currently 'Hombre'), Cumpleaños (calendar icon, text input 'dd/mm/aaaa'), Idioma (dropdown menu, currently 'Español')
- Email (text input), Número de teléfono (text input)
- Pais (dropdown menu, currently 'España'), Provincia (text input), Localidad (text input)
- Dirección (text input), Código postal (text input)
- Trabajo (text input), Tipo de sangre (dropdown menu, currently 'A+')

At the bottom of the form is an 'Aceptar' button. Below the form, there is a footer with the text 'Copyright © Facultad de Informática - Universidade da Coruña'.

Importante:

- Los recepcionistas pueden registrar pacientes, en su caso no aparecería las entradas de contraseña, ya que esta sería igual al dni del paciente.

2.2 Registro de empleados

Para el registro de empleados, únicamente podrá realizarlo el administrador.

Importante:

- Para habilitar el desplegable de especialidad, hay que seleccionar como tipo de usuario: Médico.
- Para realizar el registro correctamente, hay que introducir un horario laboral válido, es decir, el horario de mañana tiene que acabar antes que el de tarde; y los horarios de salida acabar más tarde que los de entrada.

HealthFic Gorka Home Registrar empleado Buscar empleados Ver festivos Estadísticas

Registro de empleados

Documento de identidad

Nombre Primer apellido Segundo apellido

Sexo Hombre Cumpleaños dd/mm/aaaa Idioma Español

Email Número de teléfono

Pais España Provincia Localidad

Dirección Código postal

Tipo de Usuario Recepcionista Especialidad Alergología

Nuss Salario

Horario laboral

Días de trabajo: Lunes Martes Miércoles Jueves
 Viernes Sábado Domingo

Entrada por la mañana --:-- Salida por la mañana --:--

Entrada por la tarde --:-- Salida por la tarde --:--

Aceptar

Copyright © Facultad de Informática - Universidade da Coruña

2.3 Autenticación

HealthFic Home Autenticación

Autenticación

Usuario

Contraseña

Recordar mi contraseña (las cookies deben estar habilitadas)

Aceptar

[Regístrame \(usuario no registrado\)](#)

Copyright © Facultad de Informática - Universidade da Coruña

Para acceder a la aplicación, el usuario debe de hacer uso de sus credenciales de acceso (dni y contraseña).

Importante:

- En el caso de que el paciente no esté registrado, tendrá que acceder al formulario de registro (apartado 2.1) o solicitar al recepcionista que lo registre, en este caso, inicialmente su contraseña será igual a su dni.
- En el caso de que un empleado no esté registrado, tendrá que ponerse en contacto con el administrador para que haga su registro (apartado 2.2).

2.4 Modificación de perfil

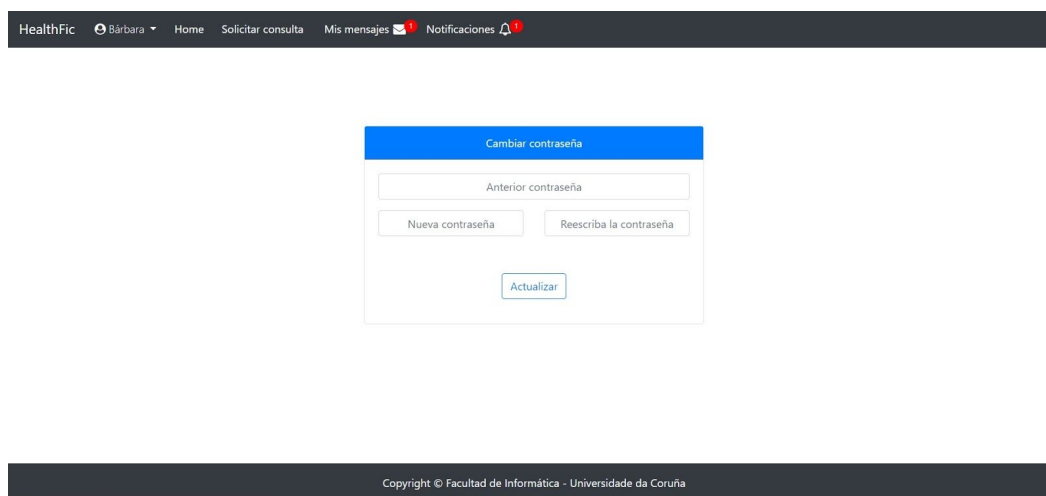
Los usuarios podrán modificar sus perfiles una vez autenticados, a través de un formulario.

Importante:

- En el caso de que se modifique el horario laboral del médico, todas las consultas pendientes que se vean afectadas, es decir, que se realicen en horario en el que el médico no trabaje, se actualizarán su fecha a una que tenga libre el médico.

2.4 Modificación de la contraseña

Los usuarios autenticados podrán actualizar su contraseña por una nueva a través de un formulario.



The screenshot shows a web application interface. At the top, a dark navigation bar contains the text 'HealthFic', a user profile icon for 'Bárbara', and menu items: 'Home', 'Solicitar consulta', 'Mis mensajes' (with a notification icon), and 'Notificaciones' (with a notification icon). Below this, a white modal window titled 'Cambiar contraseña' is centered. It contains three input fields: 'Anterior contraseña', 'Nueva contraseña', and 'Reescriba la contraseña'. Below these fields is an 'Actualizar' button. At the bottom of the page, a dark footer bar contains the text 'Copyright © Facultad de Informática - Universidade da Coruña'.

2.5 Cerrar sesión

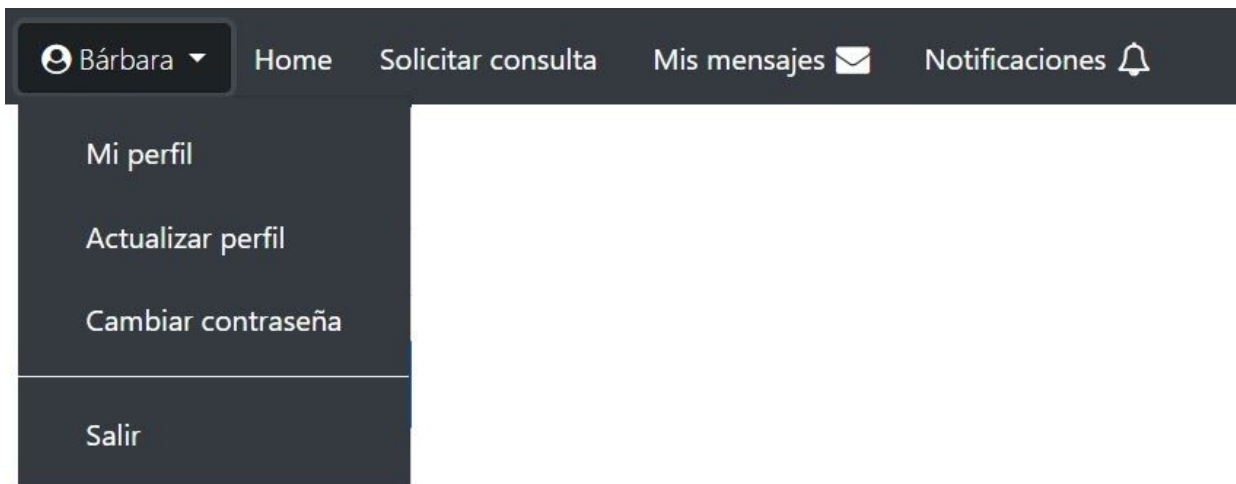
Los usuarios autenticados podrán cerrar sesión en el desplegable de su nombre en su barra de navegación.

3. Funcionalidades

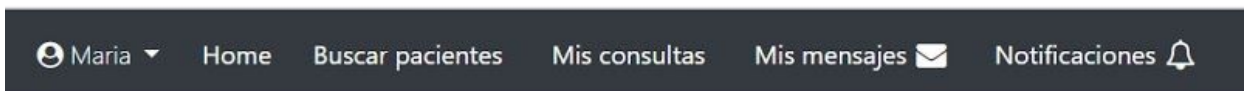
3.1 Barras de navegación

El usuario autenticado, tendrá su propia barra de navegación dependiendo de su rol. Al seleccionar su nombre, saldrá un desplegable con operaciones relacionadas con su perfil (se puede ver en el apartado 3.1.1).

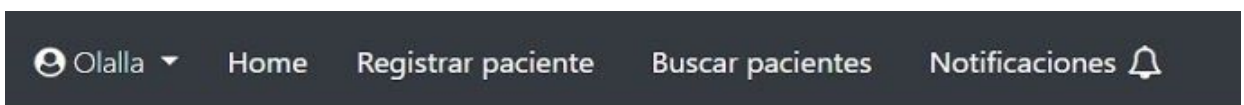
3.1.1 Paciente



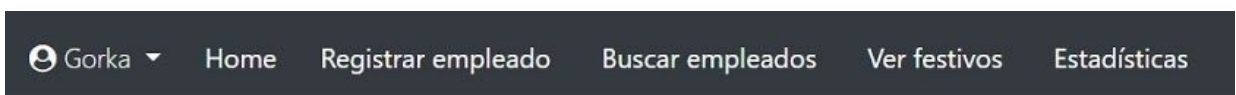
3.1.2 Médico



3.1.3 Recepcionista



3.1.4 Administrador



3.2 Menú principal

3.2.1 Paciente

Una vez autenticado, el paciente será redirigido a su página principal donde se mostrarán sus consultas pendientes y podrá filtrarlas por especialidad, fecha y palabras clave.

HealthFic Barbara Home Solicitar consulta Mis mensajes 2 Notificaciones 3

Especialidad ALL Fecha 17/02/2021 Buscar

Especialidad	Médico	Motivo	Fecha	Hora	Pagada	
Alergología	Maria Rodriguez Pereira	Alergia	17/02/2021	10:00		<input type="button" value="Ver consulta"/> <input type="button" value="Modificar hora"/>

0

Copyright © Facultad de Informática - Universidade da Coruña

3.2.2 Médico

Una vez autenticado, el médico será redirigido a su página principal donde se mostrarán sus consultas pendientes y podrá filtrarlas por fecha y palabras clave.

HealthFic María Home Buscar pacientes Mis consultas Mis mensajes 2 Notificaciones 3

Fecha 17/02/2021 Buscar

Dni	Paciente	Motivo	Fecha	Hora	Pagada	
12345678D	Bárbara Corredera Muradas	Alergia	17/02/2021	10:00		<input type="button" value="Ver consulta"/> <input type="button" value="Modificar hora"/>

0

Copyright © Facultad de Informática - Universidade da Coruña

3.2.3 Recepcionista

Una vez autenticado, el recepcionista será redirigido a su página principal donde se mostrarán las consultas pendientes cobro y podrá filtrarlas por especialidad, fecha y palabras clave.

HealthFic [Olalla](#) Home [Registrar paciente](#) [Buscar pacientes](#) [Notificaciones](#)

Especialidad ALL Fecha 17/02/2021 Buscar

Especialidad	Dni	Paciente	Médico	Motivo	Fecha	Hora	Realizada	
Alergología	12345678D	Bárbara Corredera Muradás	Maria Rodríguez Pereira	Alergia	17/02/2021	10:00	<input checked="" type="checkbox"/>	Ver consulta

0

Copyright © Facultad de Informática - Universidade da Coruña

3.3 Perfiles

3.3.1 Paciente

HealthFic [Bárbara](#) Home [Solicitar consulta](#) [Mis mensajes](#) [Notificaciones](#)

Bárbara Corredera Muradás Paciente **A+**

Documento de identidad: 12345678D Trabajo: Estudiante
 Sexo: Hombre Idioma: Español Cumpleaños: 01/01/1990
 Email: barbicm@udc.es Número de teléfono: 988123456
 País: España Provincia: A Coruña Localidad: A Coruña
 Dirección: Plaza Pontevedra Nº1 5ºB
 Código postal: 15001

[Ver vacunas](#) [Ver antecedentes médicos](#)
[Consultas](#) [Pruebas](#) [Tratamientos](#)

Copyright © Facultad de Informática - Universidade da Coruña

3.3.2 Empleados

HealthFic [Maria](#) Home [Buscar pacientes](#) [Mis consultas](#) [Mis mensajes](#) [Notificaciones](#)

Maria Rodríguez Pereira Médico **Alergología**

Documento de identidad: 12345678P Nuss: 1234123412341233
 Sexo: Mujer Idioma: Español Cumpleaños: 29/07/1975
 Email: mrodriguez@udc.es Número de teléfono: 988123456
 País: España Provincia: A Coruña Localidad: A Coruña
 Dirección: Plaza Pontevedra Nº1 5ºB
 Código postal: 15001

Días de trabajo: Lunes Martes Miércoles Jueves
 Viernes Sábado Domingo

Horario:
 Mañana: 10:00 - 14:00
 Tarde: 16:00 - 20:00

[Vacaciones](#)

Copyright © Facultad de Informática - Universidade da Coruña

3.4 Buscar pacientes

Los usuarios autenticados como médicos o recepcionistas, podrán buscar los pacientes que están registrados, con la opción de filtrarlos a través de su nombre o dni.

HealthFic Olalla Home Registrar paciente Buscar pacientes Notificaciones

Buscar

Dni	Nombre	Primer apellido	Segundo apellido	Tipo de Usuario	
12345678D	Bárbara	Corredera	Muradás	Paciente	Solicitar consulta
12345678E	Rocío	Álvarez	Moldes	Paciente	Solicitar consulta
12345678F	Luis	Pérez	Mondelo	Paciente	Solicitar consulta
12345678H	Luisa	Rodríguez	Campos	Paciente	Solicitar consulta
12345678I	Carlos	Moral	Campos	Paciente	Solicitar consulta
12345678J	Fernando	Vidal	Campos	Paciente	Solicitar consulta
12345678K	Carla	Puente	Matamoros	Paciente	Solicitar consulta
12345678L	Mireia	Marti	Vilas	Paciente	Solicitar consulta
12345678M	Demora	Amoedo	Fernández	Paciente	Solicitar consulta

Copyright © Facultad de Informática - Universidade da Coruña

3.5 Buscar empleados

HealthFic Gorka Home Registrar empleado Buscar empleados Ver festivos Estadísticas

Tipo de Usuario: Todos

Buscar

Dni	Nombre	Primer apellido	Segundo apellido	Tipo de Usuario		
12345678B	Andrea	Fernández	León	Médico	Actualizar perfil	Ver vacaciones
12345678C	Olalla	Vidal	Diaz	Recepcionista	Actualizar perfil	Ver vacaciones
12345678G	Marisol	Campos	Mondelo	Médico	Actualizar perfil	Ver vacaciones
12345678P	Maria	Rodríguez	Pereira	Médico	Actualizar perfil	Ver vacaciones
12345678Q	Jose Luís	Rodríguez	Afreijo	Médico	Actualizar perfil	Ver vacaciones
12345678R	Elena	Fernández	Núñez	Médico	Actualizar perfil	Ver vacaciones
12345678S	Lucía	Valle	Fernández	Médico	Actualizar perfil	Ver vacaciones
12345678T	Alba	González	Rocha	Médico	Actualizar perfil	Ver vacaciones

Copyright © Facultad de Informática - Universidade da Coruña

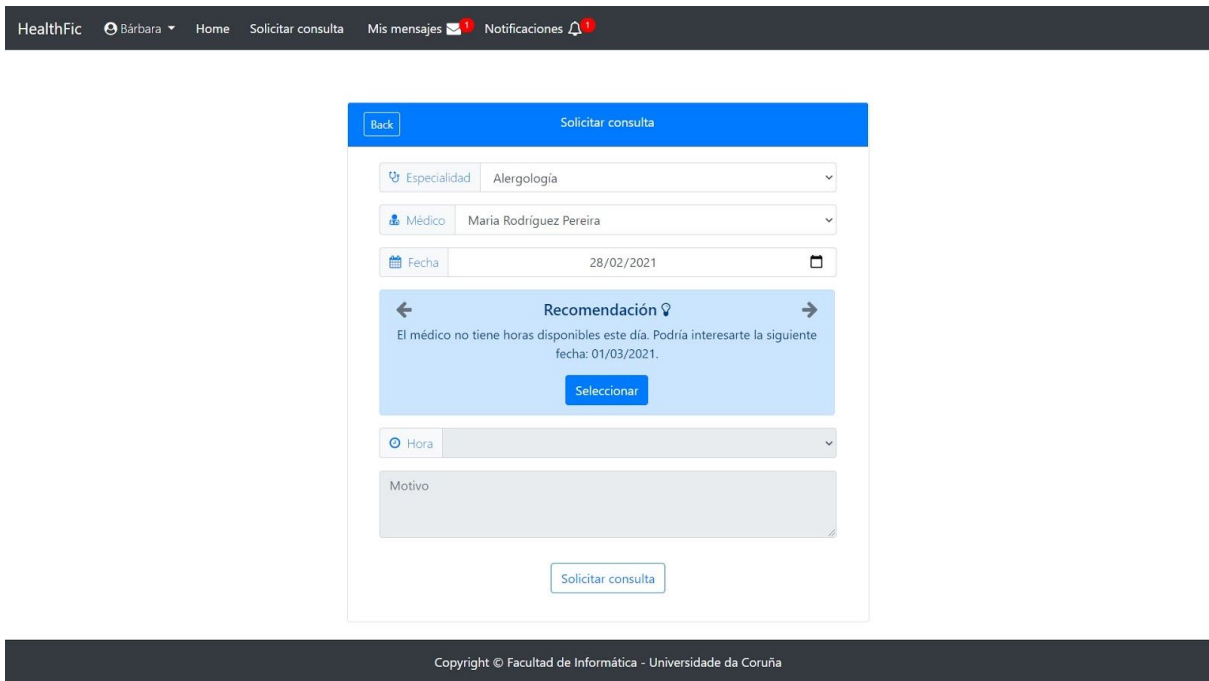
El usuario autenticado como administrador, podrá buscar los empleados que están registrados, con la opción de filtrarlos a través de tipo de usuario y su nombre o dni.

3.5 Solicitud de consulta

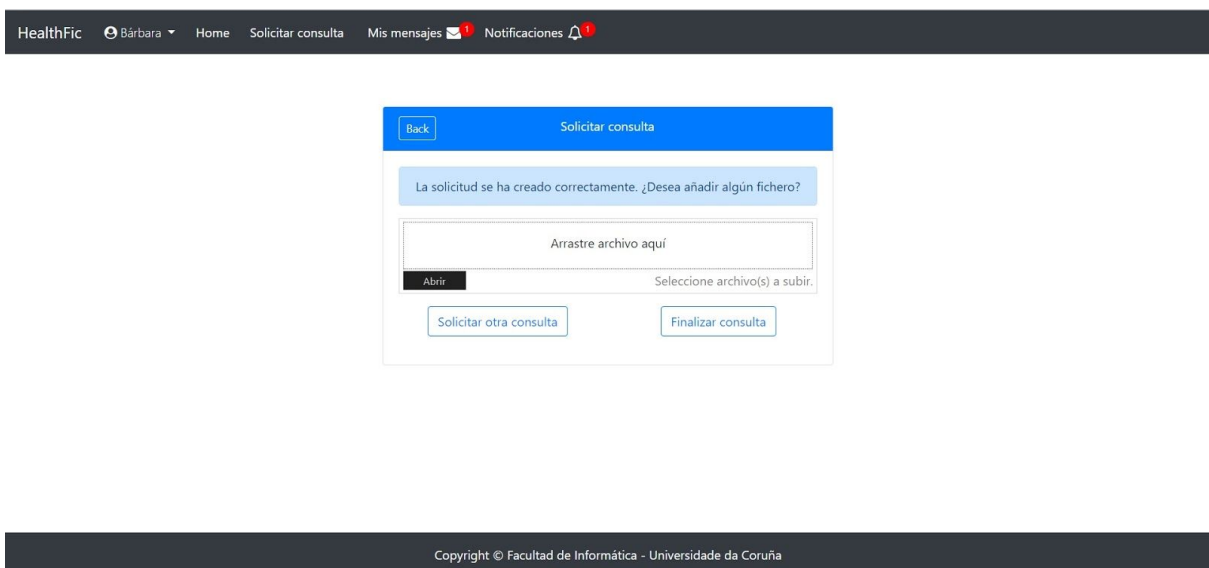
Para solicitar consulta, el usuario tendrá que ir seleccionando la especialidad, después el médico, posteriormente la fecha y finalmente la hora.

Importante:

- En el caso de que el médico no tenga horas libres el día seleccionado, se mostrarán una serie de recomendaciones de días que esté disponible, y en el caso de que haya más médicos en la especialidad seleccionada, se mostrarán recomendaciones para dichos médicos.



Una vez cubiertos todos los campos se procederá a solicitar la consulta y se abrirá la siguiente pantalla donde el usuario podrá adjuntar documentos a la consulta.



3.6 Ver consulta

HealthFic María Home Buscar pacientes Mis consultas Mis mensajes Notificaciones

Bárbara Corredera Muradás Tipo de sangre: A+ Consultas Vacunas Antecedentes


Dni: 12345678D Sexo: Hombre

Médico: **Maria Rodríguez Pereira** Alergología

Consulta: Pendiente Pagada: No 02/17/2021 - 10:00 Modificar hora Cancelar consulta

Motivo: Alergia


Documentos adjuntos:

Tipo de archivo	Nombre
 image/jpg	Tarjeta Sanitaria.jpg


Arrastre archivos aquí

Abrir Seleccione archivo(s) a subir.

Pruebas: Añadir prueba Ver pruebas del paciente

Tipo	Nombre	Fecha	
 Análisis de orina	Análisis	17/02/2021	Ver detalles Actualizar

Tratamientos: Añadir tratamiento Ver tratamientos del paciente

Tipo	Nombre	Comienza	Finaliza	
 Medicamento	Paracetamol	17/02/2021	24/02/2021	Ver detalles Actualizar

Diagnóstico: Guardar

Diagnóstico

Observaciones

Generar justificante médico Finalizar consulta

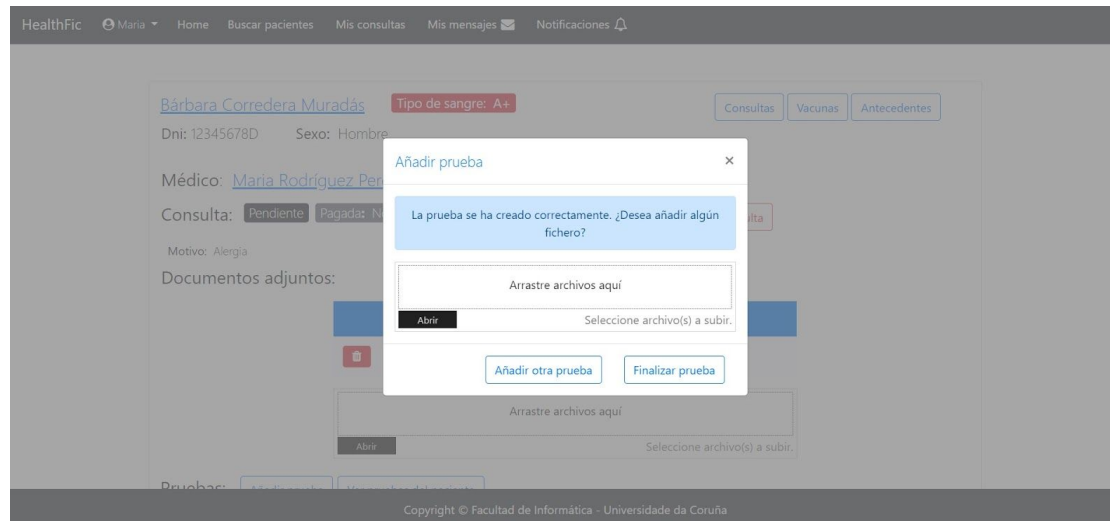
Al seleccionar “Ver consulta” se mostrarán todos los datos relacionados con esta misma. El usuario autenticado como médico podrá añadir pruebas (apartado 3.6.1), tratamientos (apartado 3.6.2) y un diagnóstico. Además podrá finalizar la consulta y generar un justificante médico.

Importante:

- Al finalizar la consulta, se podrá exportar sus datos a pdf a través del botón “Exportar a PDF”. Además los recepcionistas podrán cobrar las consultas (apartado 3.6.3).

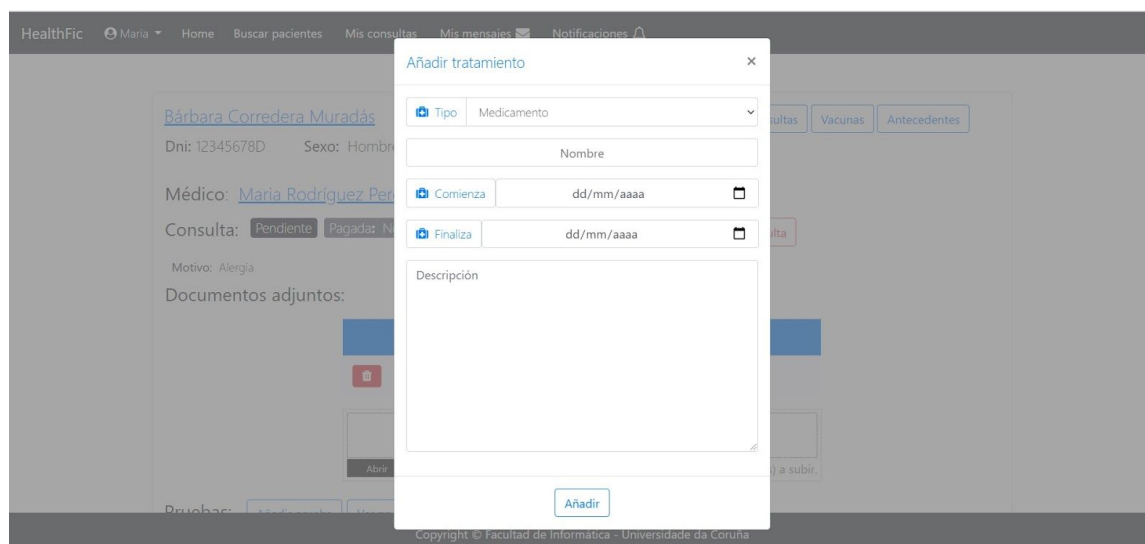
3.6.1 Añadir prueba

El usuario autenticado como médico, podrá añadir pruebas médicas a un paciente a través de un formulario.



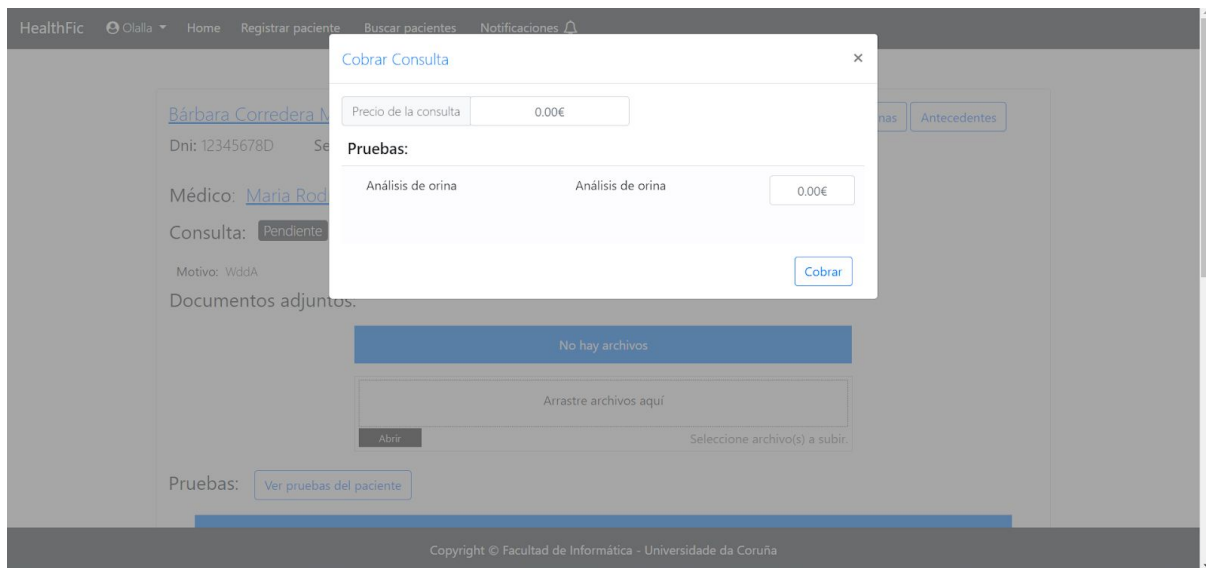
3.6.2 Añadir tratamiento

El usuario autenticado como médico, podrá añadir tratamiento médicos a un paciente a través de un formulario.

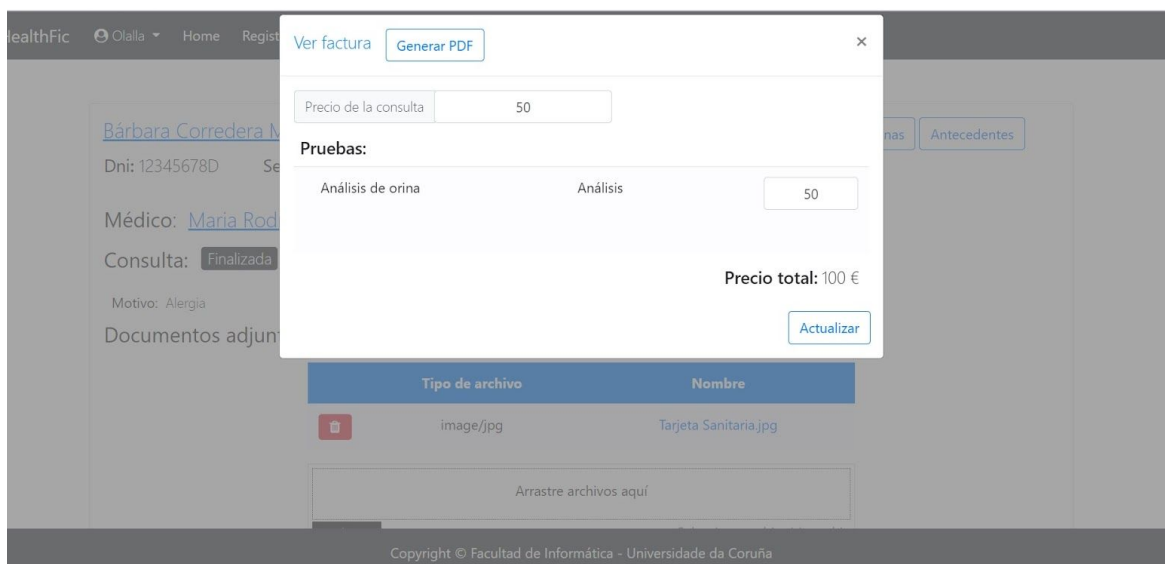


3.6.3 Cobro de consultas

Los recepcionistas, cuando acceden a una consulta, podrán cobrarla a través del botón "Cobrar", mostrando un formulario con la consulta y la lista de pruebas para establecer el precio de cada una de ellas.



Una vez realizado el cobro, aparecerá el botón de “Ver factura”, donde se podrá ver la factura, exportarla a Pdf, y en el caso de los recepcionistas, actualizar los precios.



3.6.4 Otras operaciones

Eliminación. Los usuarios autenticados como médicos podrán borrar pruebas y tratamientos a través del botón rojo situado a la izquierda de cada uno de ellos.

Modificación. Los usuarios autenticados como médicos podrán modificar pruebas y tratamientos a través del botón “Actualizar” situado a la derecha de cada uno de ellos.

Ver detalle. Los usuarios autenticados podrán ver los detalles de las pruebas y tratamientos a través del botón “Ver detalles” situado a la derecha de cada uno de ellos.

Generación de recetas médicas. Se podrá generar recetas médicas en los tratamientos que tengan como tipo “Medicamento” a través del botón Generar receta situado en los detalles del mismo

Exportación de los datos de la consulta. Se podrá exportar los datos de la consulta una vez finalizada. En el archivo generado se incluirán los datos del paciente y sus antecedentes, datos del médico y datos de la consulta junto con sus archivos adjuntos, pruebas y tratamientos.

3.7 Ver tratamientos del paciente

HealthFic María Home Buscar pacientes Mis consultas Mis mensajes Notificaciones

[Volver](#) Bárbara Corredera Muradás

Tipo: Todos Tratamientos activos

Mes: Todos Año:

Tipo	Nombre	Comienza	Termina		
Medicamento	Paracetamol	17/02/2021	24/02/2021	<input type="button" value="Ver detalles"/>	<input type="button" value="Ver consulta"/>

0

Copyright © Facultad de Informática - Universidade da Coruña

Al seleccionar “Ver tratamientos del paciente”, se mostrará una lista de los tratamientos médicos de un paciente, con la opción de filtrarlos por tipo de tratamiento, mes, año y si están activos o no.

Importante:

- Solo se puede filtrar por mes y año, cuando buscas los tratamientos que no están activos.

3.8 Ver pruebas del paciente

Al seleccionar “Ver pruebas del paciente”, se mostrará una lista de los tratamientos médicos de un paciente, con la opción de filtrarlos por tipo de prueba, mes y año.

HealthFic María Home Buscar pacientes Mis consultas Mis mensajes Notificaciones

[Volver](#) Bárbara Corredera Muradás

Tipo: Análisis de orina Mes: Todos Año:

Tipo	Nombre	Fecha		
Análisis de orina	Análisis	17/02/2021	<input type="button" value="Ver detalles"/>	<input type="button" value="Ver consulta"/>

0

Copyright © Facultad de Informática - Universidade da Coruña

3.9 Ver consultas del paciente

The screenshot shows the 'HealthFic' interface for a patient named Bárbara. The navigation bar includes 'Home', 'Solicitar consulta', 'Mis mensajes', and 'Notificaciones'. The main content area features a search filter with dropdowns for 'Especialidad' (set to ALL), 'Estado' (set to Pendiente), 'Mes' (set to Febrero), and 'Año'. A 'Buscar' button is present. Below the filters is a table of consultations:

Especialidad	Médico	Motivo	Fecha	Hora	Realizada	Pagada	
Alergología	Maria Rodriguez Pereira	Alergia	17/02/2021	10:00	✗	✗	Ver consulta

A 'Volver' link is located at the top left, and a '0' indicator is at the bottom right of the table area. The footer contains the copyright notice: 'Copyright © Facultad de Informática - Universidade da Coruña'.

Al seleccionar “Consultas” en el perfil del paciente, se mostrará una lista de todas sus consultas, con la opción de filtrarlas por especialidad, por su estado, por mes y año y por palabras clave contenidas en el nombre del médico.

3.10 Ver consultas del médico

The screenshot shows the 'HealthFic' interface for a doctor named María. The navigation bar includes 'Home', 'Buscar pacientes', 'Mis consultas', 'Mis mensajes', and 'Notificaciones'. The main content area features search filters for 'Mes' (Febrero) and 'Año'. Below the filters is a table of consultations:

Dni	Paciente	Motivo	Fecha	Hora	Realizada	Pagada	
12345678D	Bárbara Corredera Muradás	Alergia	17/02/2021	10:00	✗	✗	Ver consulta

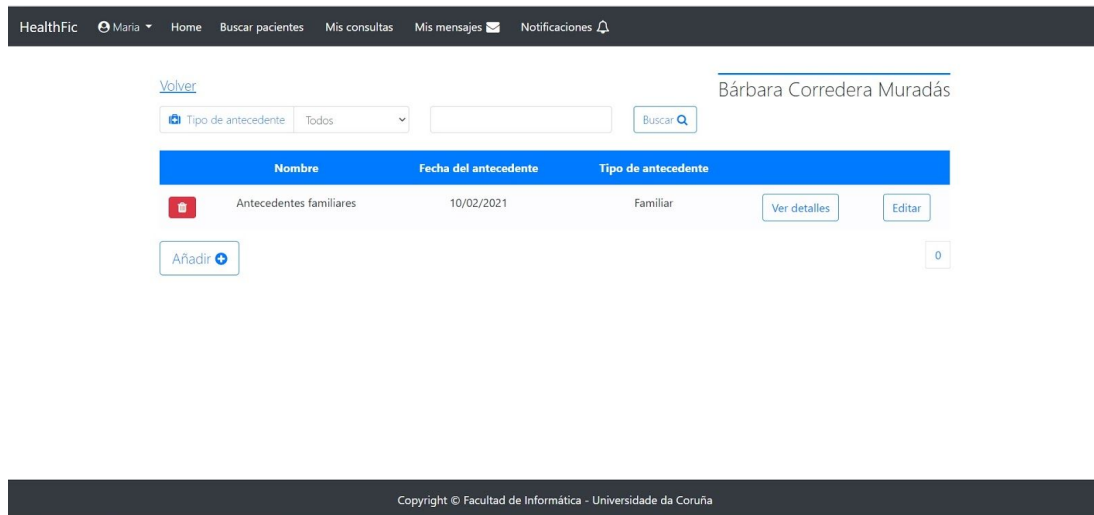
A '0' indicator is at the bottom right of the table area. The footer contains the copyright notice: 'Copyright © Facultad de Informática - Universidade da Coruña'.

El usuario autenticado como médico, al seleccionar “Mis consultas”, se mostrará una lista de todas sus consultas, con la opción de filtrarlas por su estado, por mes y año y por palabras clave contenidas en el nombre del paciente o su dni.

3.11 Antecedentes

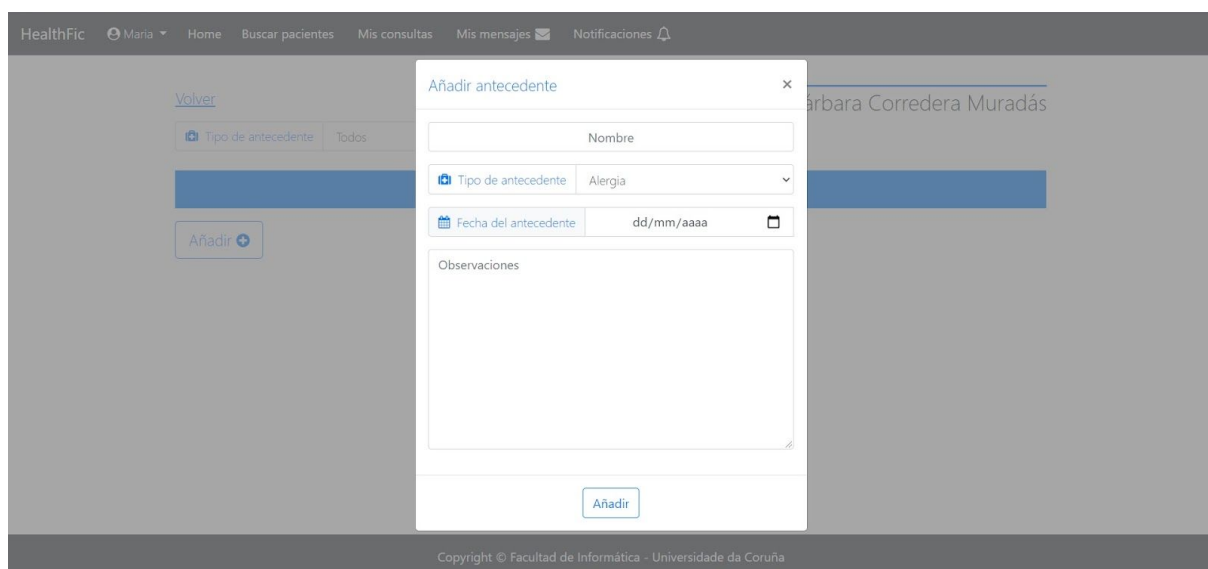
3.11.1 Ver antecedentes de un paciente

Al seleccionar “Antecedentes”, se mostrará una lista de los antecedentes médicos de un paciente, con la opción de filtrarlos por tipo de antecedente y por palabras clave contenidas en su nombre.



3.11.2 Añadir antecedente

El usuario autenticado como médico, podrá añadir antecedentes médicos a un paciente a través de un formulario.



3.11.3 Otras operaciones

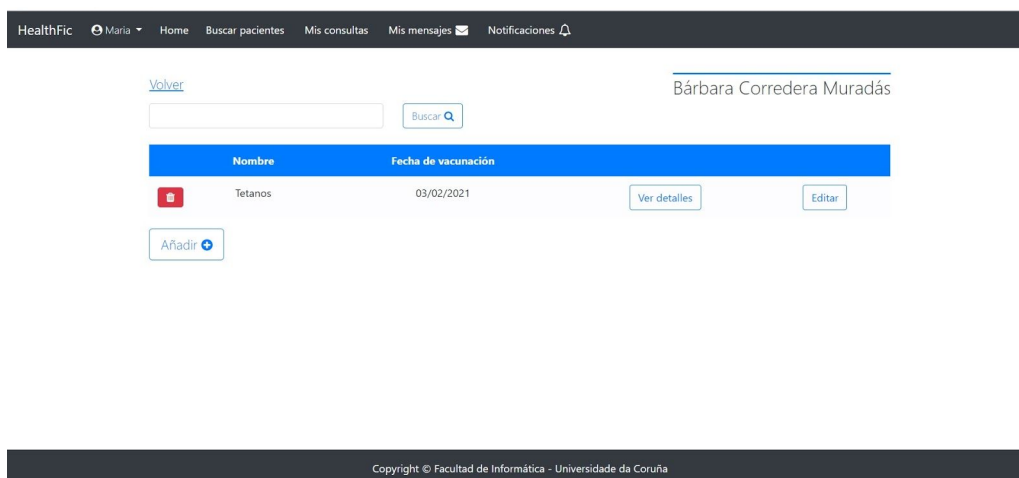
Eliminación. Los usuarios autenticados como médicos podrán borrar antecedentes a través del botón rojo situado a la izquierda de cada uno de ellos.

Modificación. Los usuarios autenticados como médicos podrán modificar antecedentes a través del botón “Editar” situado a la derecha de cada uno de ellos.

Ver detalle. Los usuarios autenticados podrán ver los detalles de una vacuna a través del botón “Ver detalles” situado a la derecha de cada uno de ellos.

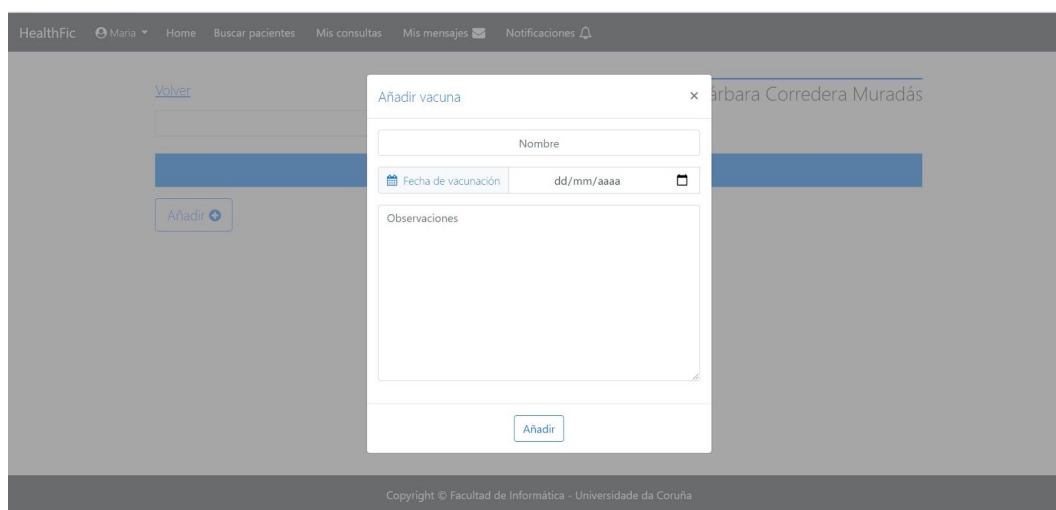
3.12 Vacunas

3.12.1 Ver vacunas de un paciente



Al seleccionar “Vacunas”, se mostrará una lista de las vacunas de un paciente, con la opción de filtrarlas por palabras clave contenidas en su nombre.

3.12.2 Añadir vacuna



El usuario autenticado como médico, podrá añadir vacunas a un paciente a través de un formulario.

3.12.3 Otras operaciones

Eliminación. Los usuarios autenticados como médicos podrán borrar vacunas a través del botón rojo situado a la izquierda de cada una de ellas.

Modificación. Los usuarios autenticados como médicos podrán modificar vacunas a través del botón “Editar” situado a la derecha de cada una de ellas.

Ver detalle. Los usuarios autenticados podrán ver los detalles de una vacuna a través del botón “Ver detalles” situado a la derecha de cada una de ellas.

3.13 Notificaciones

Los usuarios autenticados como médico, paciente o recepcionista, recibirán notificaciones cuando se cree, se modifique la fecha o se cancele una consulta.

HealthFic Andrea Home Buscar pacientes Mis consultas Mis mensajes Notificaciones

Tipo Todas Fecha dd/mm/aaaa Notificaciones vistas Buscar

Tipo	Fecha	Descripción
Solicitud de consulta por el recepcionista	15/02/2021	Solicitud de consulta el día 16/02/2021 a las 10:00 por el recepcionista.
Modificación del horario por vacaciones	17/02/2021	Se ha modificado la fecha de la consulta del día 19/02/2021 a las 10:00, al día 22/02/2021 a las 10:00 debido a vacaciones del médico.

0

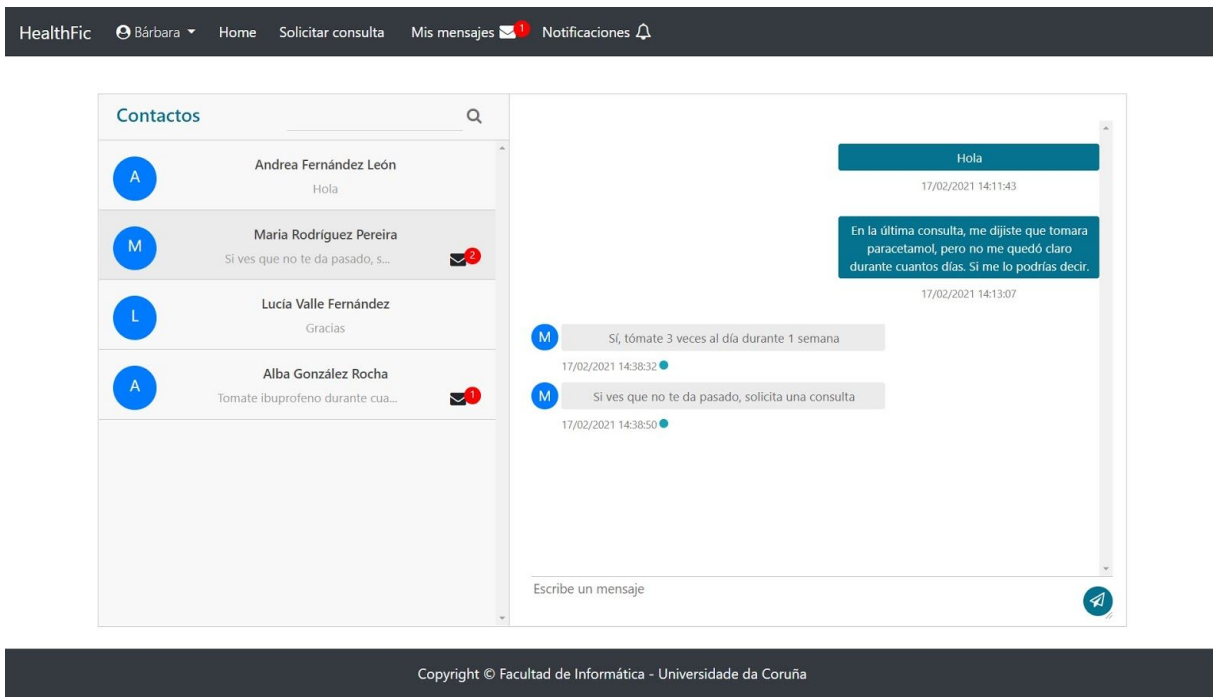
Copyright © Facultad de Informática - Universidade da Coruña

3.14 Mis mensajes

Tanto los pacientes, como los médicos podrán mandar mensajes entre ellos. Para seleccionar una conversación, hay que seleccionar en el nombre del contacto con el que se desea hablar.

Importante:

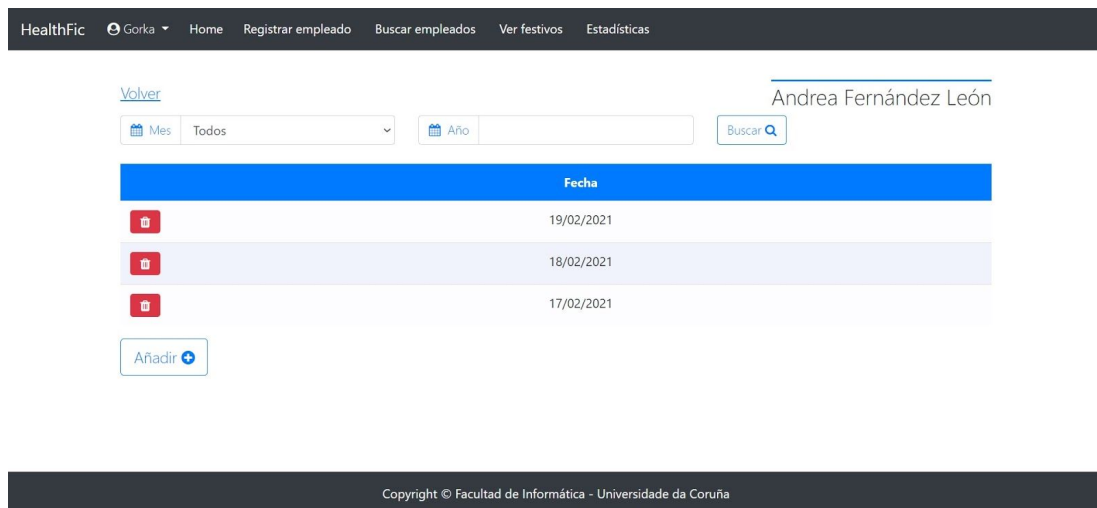
- En la lista de contactos, únicamente aparecerán los médicos con los que el paciente ha tenido consultas, y en el caso de los médicos, los pacientes con los que ha tenido consultas.
- Si con un contacto, tienes mensajes sin leer, aparecerá un símbolo con el número de mensajes que el usuario no ha leído. Una vez seleccionada la conversación, se verán los nuevos mensajes marcados con un círculo azul.



3.15 Vacaciones

3.15.1 Ver vacaciones de un empleado

Al seleccionar “Vacaciones” en el perfil del empleado, se mostrará una lista de los días que tiene vacaciones un empleado, con la opción de filtrarlos por mes y año.

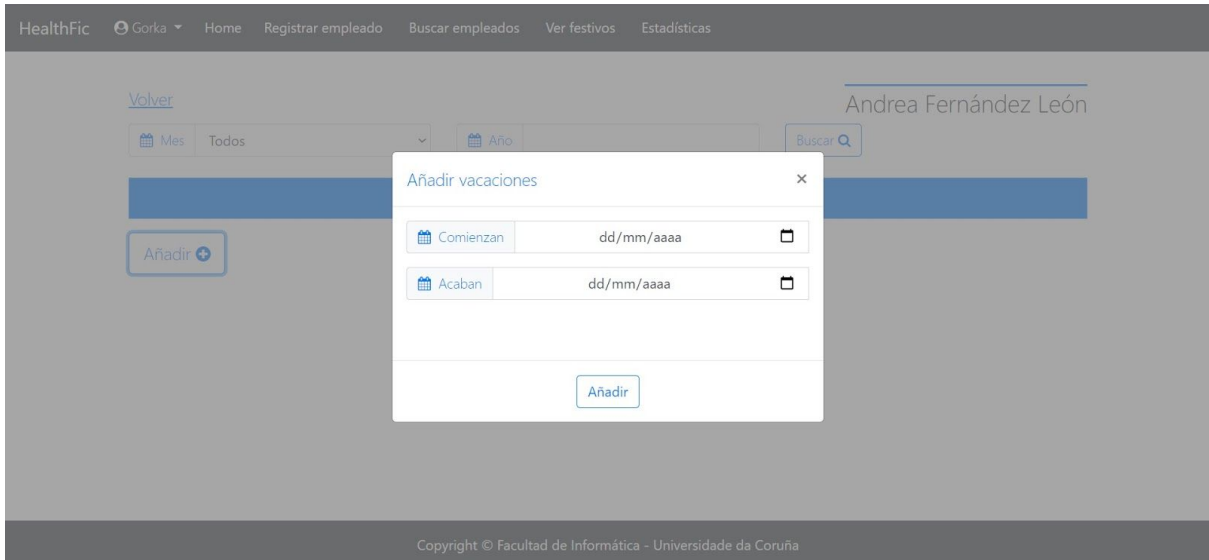


3.15.2 Añadir vacaciones

El usuario autenticado como administrador, podrá añadir vacaciones a un empleado a través de un formulario.

Importante.

- Si se le añaden vacaciones a un médico, todas las consultas pendientes de dicho médico que se vean afectadas, es decir, que se realicen en esos días, se actualizarán su fecha a una que tenga libre el médico.



3.16.3 Otras operaciones

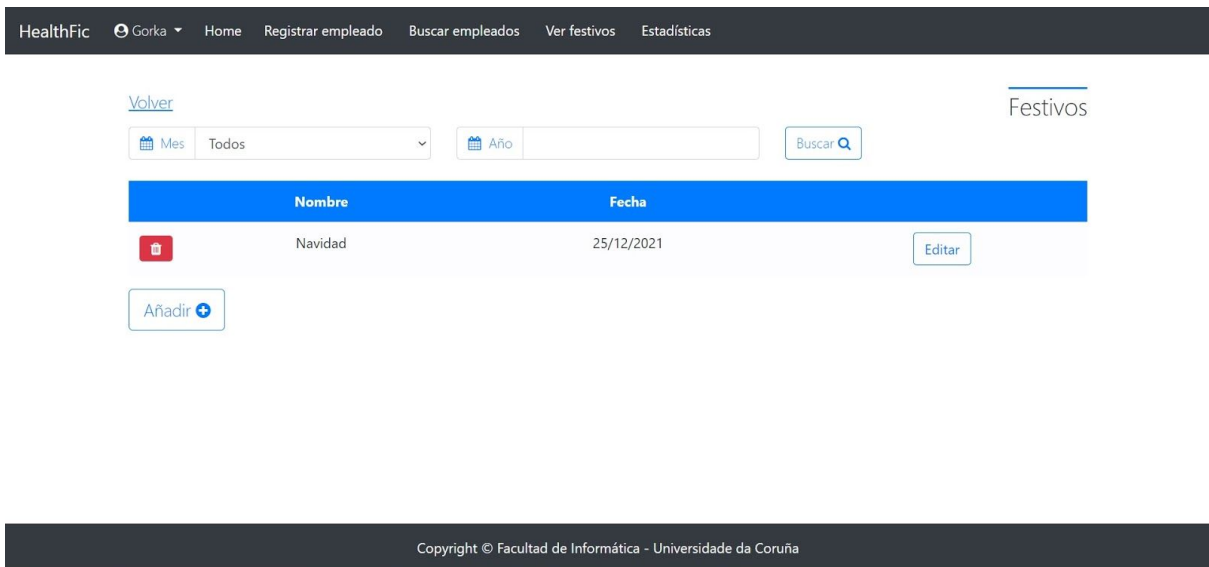
Eliminación. Los usuarios autenticados como administradores podrán borrar vacaciones a través del botón rojo situado a la izquierda de cada uno de ellos.

Modificación. Los usuarios autenticados como administradores podrán modificar festivos a través del botón “Editar” situado a la derecha de cada uno de ellos. Al igual que al añadir, se modificará la fecha de las consultas que se vean afectadas.

3.17 Festivos

3.17.1 Ver festivos

El usuario autenticado como administrador al seleccionar “Ver festivos” en su barra de navegación, se mostrará una lista de los días festivos que tiene la clínica, con la opción de filtrarlos por mes y año.

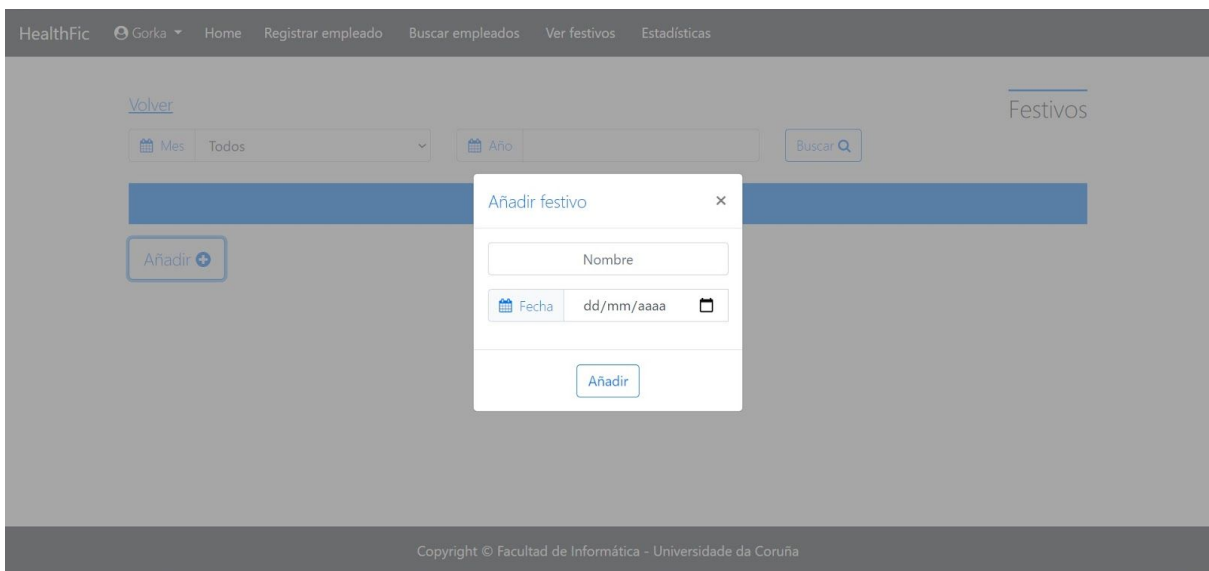


3.17.2 Añadir Festivo

El usuario autenticado como administrador, podrá añadir festivos a través de un formulario.

Importante.

- Si se añaden festivos, todas las consultas pendientes que se vean afectadas, es decir, que se realicen en esos días, se actualizarán su fecha a una que tengan libres los médicos.

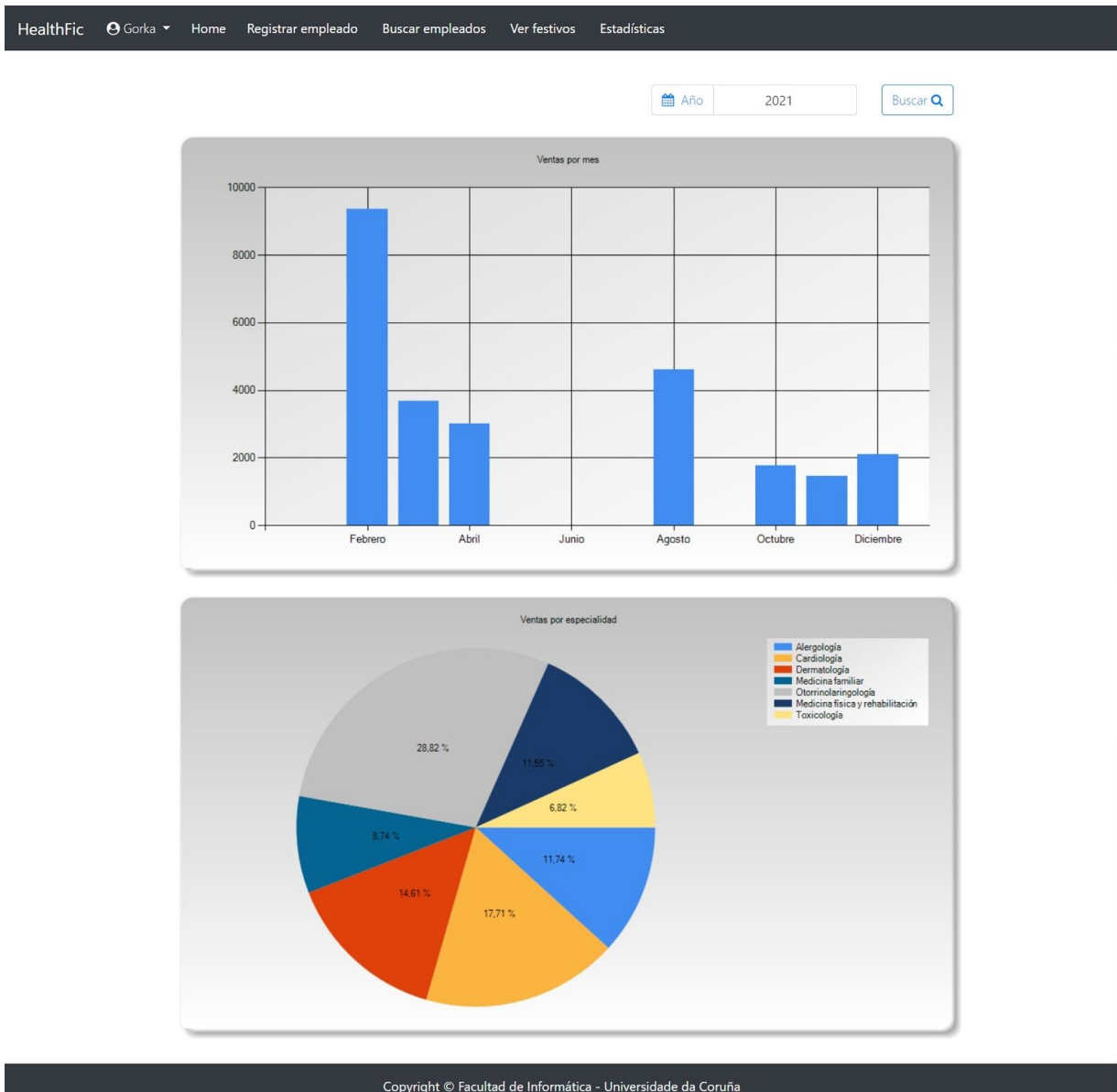


3.17.3 Otras operaciones

Eliminación. Los usuarios autenticados como administradores podrán borrar festivos a través del botón rojo situado a la izquierda de cada uno de ellos.

3.18 Estadísticas

El usuario autenticado como administrador, podrá ver estadísticas de las ganancias de la clínica filtrando por año.



Apéndice B

Exportación PDF de una consulta

CONSULTA

1. Datos generales de la consulta

1.1 Paciente

Nombre: Mireia Marti Vilas

Tipo de Sangre: A+ **Trabajo:** Estudiante

DNI: 12345678L **Sexo:** Mujer **Fecha de nacimiento:** 01/01/1991

Correo electrónico: mireimarti@udc.es **Nº de Teléfono:** 988123456

Dirección: Plaza Pontevedra Nº1 5ºB , A Coruña, A Coruña, España, CP: 15001

1.1.1 Antecedentes médicos

1.1.1.1 Antecedentes heredofamiliares

Tipo: Familiar **Fecha:** 18/06/2020

Observaciones:

Padre de 55 años que goza de buena salud. Madre de 45 años que padece asma de los 15 años diagnosticada por facultativo y tratada desde entonces por médico familiar controlándose con broncodilatadores en aerosol, sin haber presentado exarberaciones en los últimos 10 años.

El paciente no tiene hermanos y no se proporcionan datos de otros familiares.

1.1.1.2 Diabetes Tipo 2

Tipo: Personal **Fecha:** 08/01/2011

Observaciones:

Diabetes diagnosticada a los 20 años de edad. Al inicio la sintomatología que presentaba la paciente (poldipsia ,polifagia, poliurea) fue necesaria para el diagnóstico. Refiere que desde un inicio fue tratada con insulina endovenosa.

Forma de Inicio: insidioso con curso progresivo

Signos y síntomas principales: dolor

1.2 Médico

Nombre: Andrea Fernández León

Especialidad: Medicina familiar **Nuss:** 1234123412341230

Correo electrónico: andreafernandezleon@udc.es **Nº de Teléfono:** 988123456

1.3 Consulta

Especialidad: Medicina familiar **Fecha:** 01/02/2021 - 16:30

Motivo de la consulta: Dolor abdominal

Diagnóstico:

No se ha encontrado nada en la actual TC que explique las quejas clínicas sobre el dolor abdominal del paciente. Para calmar el dolor se le ha recetado tomar Ibuprofeno y paracetamol durante 7 días.

Por otro lado, hay una lesión nueva de 2 cm en el hígado que se considera indeterminada (no se puede diagnosticar en forma definitiva con el estudio).

Observaciones:

RECOMENDACIÓN: teniendo en cuenta la historia personal de cáncer de seno del paciente, se recomienda una RMN del hígado para caracterizar mejor la lesión indeterminada del hígado, para excluir la posibilidad de metástasis (o diseminación del cáncer).

Se ha recomendado al paciente que descanse durante estos días para observar mejor la evolución de dolor. Se adjunta el parte de baja.

1.4 Imágenes adjuntas

1.4.1 Tarjeta sanitaria.jpg



1.5 Anexos

La consula tiene adjuntados los siguientes archivos que se pueden ver al final del documento en el apartado de Anexos:

-Anexo 1. Parte de baja.pdf

2. Pruebas

2.1 Tomografía computarizada (TC) del abdomen y pelvis con contraste

Tipo: Tomografía computarizada **Fecha:** 30/01/2021

Descripción:

Imágenes axiales de 5 mm de la base de los pulmones a través de la sínfisis pública fueron adquiridas luego de la administración de contraste intravenoso y oral. Imágenes coronal y sagital reformateadas creadas a partir de datos de origen.

Se han observado los siguientes resultados:

- Cardíaco: la base del corazón se encuentra dentro de los límites normales. No se observa derrame pericárdico.
- Hígado: contorno y tamaño normales. Hay una nueva zona hipoatenuante de 2 cm. en el segmento 8. La vesícula biliar está ausente debido a cirugía.
- Biliar: no hay dilatación biliar intra o extra hepática.
- Bazo: no hay esplenomegalia.
- Páncreas: no hay masa o dilatación ductal.
- Riñones y adrenales: no hay masas, piedras o hidronefrosis. No hay nódulos adrenales.
- Ganglios linfáticos: no hay linfadenopatía.
- Intestinos: no hay dilatación o engrosamiento de pared.
- Vejiga: dentro de límites normales.
- Útero y anexos: el útero y los ovarios bilaterales están dentro de los límites normales para la edad.
- Otros: no hay líquido suelto dentro de la pelvis.

2.1.1 Tomografía abdomen.jpg



2.2 Análisis de orina

Tipo: Análisis de orina **Fecha:** 30/01/2021

Descripción:

Para el análisis de la orina se han realizado diferentes tipos de análisis.

El primero se basó en los aspectos físicos, se observó un aspecto bastante oscuro en la orina, el paciente comentó que no ha bebido mucho en estos días por lo que puede que esté concentrada. El olor era normal.

Posteriormente, se llevó al laboratorio a analizar. No se encontró ninguna anomalía.

2.2.1 Anexos

La prueba tiene adjuntados los siguientes archivos que se pueden ver al final del documento en el apartado de Anexos:

-Anexo 2. Análisis de orina.pdf

3. Tratamientos

2.1 Paracetamol 1gr.

Tipo: Medicamento

Duración: Del 01/02/2021 a 08/02/2021 (7 días).

Descripción:

Tomar 3 pastillas diarias cada 8h durante los 7 días.

2.2 Ibuprofeno 1gr

Tipo: Medicamento

Duración: Del 01/02/2021 a 08/02/2021 (7 días).

Descripción:

Tomar 3 pastillas diarias cada 8h durante los 7 días.

Anexos

-Anexo 1 - Parte de baja.pdf



FECHA DE INICIO DE LA SITUACIÓN (dd/mm/aaaa)

PARTE INICIAL PARTE CONFIRMACIÓN Nº Marcar si RECAIDA

PARTE ALTA – Fecha (dd/mm/aaaa)

Causa del alta:

<input type="checkbox"/> Curación/Mejoría que permite el trabajo habitual	<input type="checkbox"/> Fallecimiento
<input type="checkbox"/> Pase a jubilación por edad o por IP para el servicio	<input type="checkbox"/> Agotamiento del plazo máximo
<input type="checkbox"/> Posible nueva situación de IT	<input type="checkbox"/> Comienzo del permiso por maternidad

1. DATOS IDENTIFICATIVOS

1.1 MUTUALISTA

Primer apellido Segundo apellido Nombre

Número de afiliación

Nº de DNI:

Teléfono: Correo @ Entidad

1.2 FACULTATIVO

Nº de colegiado

Apellidos y nombre:

Especialidad:

2. DATOS MÉDICOS

2.1 CIE 10 ES diagnóstico •

2.2 Duración probable (días)

2.3 Descripción del diagnóstico (dolencias y su evolución):

2.4 Descripción de la limitación en la capacidad funcional

2.5 Indicar, si procede:

<input type="checkbox"/> Intervención quirúrgica	<input type="checkbox"/> Hospitalización
<input type="checkbox"/> Tratamiento quimio-radioterapia	<input type="checkbox"/> Otro procedimiento

DATOS ESPECÍFICOS

2.6 Se dan circunstancias que recomiendan ampliación plazo de expedición del próximo parte a días (máximo 30)

2.7 Sin variaciones

3. INFORME MÉDICO ADICIONAL DE RATIFICACIÓN: que acompaña al parte de confirmación del 10º o el 16º mes

Valoración de la situación (marcar una opción):

- 3.1 Posible alta por curación o mejoría antes de los 545 días naturales desde el inicio de la situación.
3.2 Posible incapacidad permanente.
3.3 Necesidad mantenimiento de los efectos de IT más allá del periodo de 545 días.

Justificación de la opción elegida

..... de de 20....

Firma del facultativo del parte



IMPORTANTE

- **Finalidad del parte:** el contenido de este parte es el asesoramiento médico para que el órgano de personal donde el mutualista presta servicios resuelva la concesión de la licencia por enfermedad, requisito imprescindible para declarar la situación de incapacidad temporal (art. 88 del Reglamento General del Mutualismo Administrativo, RD 375/2010, de 28 de marzo)
- **Plazos:** el ejemplar para la Administración del parte inicial debe aportarlo el mutualista a su órgano de personal durante los cuatro primeros días hábiles desde el inicio de la situación. El del parte de confirmación, durante los tres días hábiles siguientes a su expedición. El del parte de alta, al día hábil siguiente a su expedición.
- **Validez del parte:** La información solicitada es obligatoria para la validez del parte, que debe ser cumplimentado y firmado por el médico que atiende al mutualista.
- **Definición de recaída:** es el proceso patológico que sucede en los 180 días naturales siguientes a la finalización de uno anterior de la misma o similar patología. El médico indicará en el parte si el nuevo proceso iniciado es recaída de otro anterior.

2.1 Codificación del parte: se requiere la codificación del **diagnóstico** en **CIE 10 ES**. El formato admitido se muestra en la siguiente tabla. Los campos a la derecha del punto pueden ir en blanco o cumplimentados desde uno a cuatro caracteres alfanuméricos.

x	x	x	•	x	x	x	x
---	---	---	---	---	---	---	---

Los **códigos de procedimiento no se admiten** en los partes de IT. Si es el caso, **ampliar el diagnóstico codificado** usando el apartado 2.5

2.3 Descripción del diagnóstico: solo en el ejemplar destinado al mutualista.

2.5 Información para completar el diagnóstico: intervención quirúrgica, hospitalización u otra circunstancia, procedimiento o técnica diagnóstica.

2.6 Periodicidad de los partes: deben expedirse quincenalmente, pero si por motivos de mejor asistencia al paciente es preciso demorar esta frecuencia, podrán expedirse por más tiempo, con un máximo de 30 días entre cada parte.

2.7 Casilla “sin variaciones”: se utilizará en los partes de confirmación firmados por el mismo facultativo que expidió el inicial o el inmediato anterior de confirmación, si el diagnóstico es el mismo. De este modo no será necesario rellenar de nuevo los datos médicos ya aportados.

3. El informe adicional de ratificación acompaña a los partes de confirmación del 10º y 16º mes, y solicitan del médico su criterio sobre la evolución del proceso y las posibilidades de recuperación o de incapacidad permanente. Si es el de **10º mes**, se pronunciará sobre **3.1 o 3.2**. Si es el de **16º mes**, sobre **3.2 y 3.3**.

-Anexo 2 - Análisis de orina.pdf



COMPLETE URINALYSIS PANEL

INTERPRETATION GUIDE

Scroll down or click on the following parameters to quickly access content

A Complete Urinalysis is threefold:

- *Physical exam*

Color

[Clarity - Turbidity](#)

[Urine specific gravity](#)

- *Chemical exam*

[pH](#)

[PRO](#) (protein)

[GLU](#) (glucose)

[KET](#) (ketones)

[UBG](#) (urobilinogen)

[BIL](#) (bilirubin)

[Blood](#)

[LEU](#)

CHARLIE

Species: **Feline**
 Gender: **Male**
 Year of Birth: **2001**
 Client: **Carlos Animal Hospital**

IDEXX Representative

Veterinarian: **Peter Parker, DVM ABVP**
 Instrument: **ProCyte Dx Hematology Analyzer**
Catalyst Dx Chemistry Analyzer
IDEXX VetLab UA Analyzer

Urinalysis

3/7/2015 @ 2:49 PM

pH	5.0	
Protein	30	mg/dL
Glucose	neg	
Ketones	neg	
Blood / Hemoglobin	50	Ery/kL
Bilirubin	neg	
Urobilinogen	nom	

3/17/2015
11:12 PM

5.0

neg

neg

neg

neg

neg

neg

neg

norm

norm

SG = 1.027

- *Sediment exam* (see [urine sediment guide](#))

Cells, bacteria, casts, crystals and miscellaneous elements

Lista de acrónimos

- API** Application Programming Interface, interfaz de programación de aplicaciones. 36
- ASP** Active Server Page, página de servidor activa. 14
- BD** Base de Datos. 38, 67
- CLR** Common Language Runtime, entorno en tiempo de ejecución de lenguaje común. 13
- CRUD** Create Read Update Delete, operaciones básicas de acceso a datos (crear, leer, actualizar y eliminar). 33, 38, 45, 47, 53, 57
- CSS** Cascading Style Sheets, hojas de estilo en cascada. 12
- DAO** Data Access Object, objeto de acceso a datos. 32, 33, 38, 39, 67
- DICOM** Digital Imaging and Communication On Medicine, imagen digital y comunicación en medicina. 77
- DTO** Data Transfer Object, objeto de transferencia de datos. 36, 37
- HTML** HyperText Markup Language, lenguaje de marcas de hipertexto. 12, 14, 15, 37, 40, 41
- HTTP** HyperText Transfer Protocol, protocolo de transferencia de hipertexto. 15, 37, 39, 40
- IL** Intermediate Language. 13
- IU** Interfaz de Usuario. 1, 32, 37, 39
- JIT** Just In Time, método justo a tiempo. 13
- JPEG** Joint Photographic Experts Group, unión de grupo de expertos fotográfico. 77
- LINQ** Language INtegrated Query. 14, 38

ORM Object Relational Mapping, Mapeo Objeto-Relacional. 13

SaaS Software as a Service, software como servicio. 6, 8

SMTP Simple Mail Transfer Protocol, protocolo simple de transferencia de correo. 77

SQL Structured Query Language, lenguaje de consulta estructurada. ii, 18, 38

URI Uniform Resource Identifier, identificador de recursos uniforme. 40

URL Uniform Resource Locator, localizador de recursos uniforme. 40

Bibliografía

- [1] “Dricloud. software médico con telemedicina para tu clínica.” [En línea]. Disponible en: <https://dricloud.com/>
- [2] “Nubimed. software de gestión de clínicas en la nube.” [En línea]. Disponible en: <https://www.nubimed.com/>
- [3] “Ofimedic. software médico.” [En línea]. Disponible en: <https://www.ofimedic.com/>
- [4] “Clinic-cloud. programa de gestión de clínicas en la nube.” [En línea]. Disponible en: <https://dricloud.com/>
- [5] Microsoft, “Documentación de asp.net.” [En línea]. Disponible en: <https://docs.microsoft.com/es-es/aspnet/core/?view=aspnetcore-5.0>
- [6] “¿qué son los formularios web forms?” [En línea]. Disponible en: <https://docs.microsoft.com/es-es/aspnet/web-forms/what-is-web-forms>
- [7] “Asp.net ajax control toolkit.” [En línea]. Disponible en: <http://www.ajaxtoolkit.net/>
- [8] “Documentación de itext pdf.” [En línea]. Disponible en: <http://itextpdf.com/es/resources/api-documentation>
- [9] Microsoft, “Documentación de visual studio.” [En línea]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/windows/?view=vs-2019&preserve-view=true>
- [10] “Github guides.” [En línea]. Disponible en: <https://guides.github.com/>
- [11] K. Schwaber and J. Sutherland, “Scrum guide.” [En línea]. Disponible en: <https://www.scrumguides.org/index.html>
- [12] “Guía salarial sector ti galicia 2015-2016.” [En línea]. Disponible en: <https://es.scribd.com/document/288511179/Guia-Salarial-Sector-TI-Galicia-2015-2016>

- [13] “Cifrado transparente de una base de datos sql server.” [En línea]. Disponible en:
<https://www.jmsolanes.net/es/cifrado-base-datos-sql-server-tde/>