



EasyPhysio



Grado en Ingeniería Informática

Kevin Real Alcalde

Tutorizado por: Abraham Rodríguez Rodríguez - Francisca Quintana Domínguez

Julio de 2021



Índice

1.	Introducción.....	1
1.1.	Contexto.....	1
1.2.	Objetivos.....	2
1.3.	Metodología.....	2
1.4.	Competencias específicas cubiertas.....	4
1.5.	Aportaciones al entorno socioeconómico, técnico o científico.....	5
1.6.	Estructura de la memoria.....	5
2.	Análisis.....	7
2.1.	Estudio de aplicaciones similares.....	7
2.2.	Requisitos.....	10
2.2.1.	Tabla de <i>dummies</i>	10
2.2.2.	Requisitos funcionales.....	11
2.2.3.	Requisitos no funcionales.....	12
2.2.4.	Mockups.....	13
2.3.	Modelo de negocio.....	19
2.4.	Normativa y legislación.....	19
3.	Iteraciones.....	20
3.1.	Alcance de la implementación.....	20
3.2.	Iteración 1.....	21
3.2.1.	Requisitos.....	21
3.2.2.	Diagrama de casos de uso.....	21
3.2.3.	Mockups.....	22
3.2.4.	Diseño arquitectónico.....	24
3.2.5.	Modelado de datos.....	26
3.2.6.	Detalles de implementación.....	27
3.3.	Iteración 2.....	28
3.3.1.	Requisitos.....	28
3.3.2.	Diagrama de caso de uso.....	29
3.3.3.	Mockups.....	30
3.3.4.	Diseño arquitectónico.....	31
3.3.5.	Detalles de implementación.....	34
3.4.	Iteración 3.....	35
3.4.1.	Requisitos.....	35
3.4.2.	Diagrama de caso de uso.....	35

3.4.3.	Mockups.....	36
3.4.4.	Diseño arquitectónico.....	37
3.4.5.	Detalles de implementación.....	41
4.	Tecnologías.....	42
4.1.	Angular.....	42
4.2.	Angular CLI.....	42
4.3.	AngularFireModule y AngularFireStoreModule.....	42
4.4.	Angular Material.....	42
4.4.1.	DragDropModule.....	43
4.4.2.	MatAutocompleteModule.....	43
4.4.3.	MatCheckboxModule.....	43
4.4.4.	MatDatepickerModule.....	43
4.4.5.	MatFormField.....	43
4.4.6.	MatInputModule.....	43
4.4.7.	MatRadioModule.....	43
4.4.8.	MatSelectModule.....	43
4.4.9.	MatSidenavModule.....	44
4.4.10.	MatTooltipModule.....	44
4.5.	CSS3.....	44
4.6.	ESLint.....	44
4.7.	Firebase.....	44
4.8.	FontAwesomeModule.....	44
4.9.	Git.....	45
4.10.	HTML5.....	45
4.11.	jQuery.....	45
4.12.	Lodash.....	45
4.13.	MomentJS.....	45
4.14.	NgxChartsModule.....	45
4.15.	Ng-bootstrap.....	45
4.16.	NodeJS.....	46
4.17.	PopperJS.....	46
4.18.	RXJS.....	46
4.19.	SCSS.....	46
4.20.	TypeScript.....	46
5.	Acceso al código y despliegue.....	47
6.	Conclusiones.....	48

7. Fuentes de información.....	50
8. Anexo.....	52

Tablas

Tabla 1. Planificación TFT01.	3
Tabla 2. Planificación final.....	4
Tabla 3. Primera tabla comparativa. Aplicaciones de gestión para negocios de salud y belleza.	8
Tabla 4. Segunda tabla comparativa. Aplicaciones de gestión para el ámbito sanitario.....	10
Tabla 5. Tabla de dummies.	11

Figuras

Figura 1. Formulario de registro.....	13
Figura 2. Formulario para iniciar sesión.....	14
Figura 3. Formulario de modificación de datos personales.....	15
Figura 4. Agenda.....	15
Figura 5. Agenda. Edición de cita.....	16
Figura 6. Histórico de citas.....	17
Figura 7. Compras realizadas.....	18
Figura 8. Información de interés.....	18
Figura 9. Diagramas de casos de uso iteración 1.....	22
Figura 10. Agenda.....	23
Figura 11. Agenda. Edición de cita.....	23
Figura 12. Diagrama de entidad-relación.....	27
Figura 13. Diagramas de casos de uso iteración 2.....	29
Figura 14. Formulario de registro.....	30
Figura 15. Formulario para iniciar sesión.....	31
Figura 16. Diagramas de casos de uso iteración 3.....	36
Figura 17. Formulario para iniciar sesión.....	37
Figura 18. Información de interés.....	37

1. Introducción.

1.1. Contexto.

En España, desde la Real Orden del 26 de junio de 1860, a la profesión de practicante se le atribuyeron algunas de las competencias actuales de la fisioterapia, como son los vendajes, para posteriormente convivir dicha profesión junto a la de matrona y enfermera bajo la titulación ATS (Asistente Técnico Sanitario) desde 1915 hasta 1953.

Hasta que en 1957 se reconoció la fisioterapia como especialidad separada, aunque todavía bajo la titulación ATS. Desde entonces, la fisioterapia no ha parado de sufrir modificaciones en sus atribuciones y formación necesaria para adquirir la titulación necesaria para ejercer dicha profesión *per se*.

Desde hace décadas los centros de fisioterapia, o los centros de rehabilitación que incluyen fisioterapia entre sus servicios, han venido para quedarse en la sociedad española. La fisioterapia es una disciplina en constante evolución en sus técnicas y estudios, así como en la manera de explotarla, adaptándose a las necesidades personales de cada individuo a través de un servicio y análisis cada vez más personalizado para el paciente.

Estas constantes evoluciones en todos los aspectos de la disciplina también han llevado a otros aspectos transversales a adaptarse, como puede ser la informatización de los sistemas de gestión de los centros; la diseminación de información relativa al centro a través de redes sociales; la capacidad de los pacientes de gestionar sus citas a través de aplicaciones de mensajería instantánea; el desarrollo aplicaciones web por parte de los centros con fines divulgativos o para mantener contacto con sus pacientes. Sin embargo, hay un nicho de mercado en ciernes, siendo este la explotación de las aplicaciones móviles o de escritorio para la gestión de la planificación de citas de un centro.

Con certeza, hoy en día hay diversas aplicaciones capaces de gestionar un centro de fisioterapia *grosso modo*, pero siendo la gran mayoría aplicaciones genéricas de gestión de un negocio con segmentación horaria en sus servicios, como puede ser una peluquería, una clínica dental... pero no específicas a las necesidades de un centro de fisioterapia, como puede ser el trazado de un tratamiento individualizado con una alta capacidad de flexibilización, según el avance que experimente el paciente bajo el diagnóstico del fisioterapeuta.

Adicionalmente, hay un vacío importante en el mercado de soluciones específicas para pacientes que deseen tener en su ordenador o móvil la capacidad de gestionar sus citas en su centro de fisioterapia habitual de manera autónoma, teniendo siempre que recurrir a contactar con el centro a través de aplicaciones de mensajería instantánea,

EasyPhysio

correo o en última instancia, a través de llamadas telefónicas, para poder reservar una nueva cita o tener que modificar o eliminar algunas de las ya existentes.

1.2. Objetivos.

En este proyecto se pretende llevar a cabo un profundo análisis de las necesidades de un centro de fisioterapia para mejorar su servicio e incluso aumentar su productividad y concentrar sus soluciones en una aplicación de escritorio y de móvil.

El usuario podrá prescindir del formato físico a la hora de acudir a su centro de fisioterapia y evitar las tentaciones de perder un bono de sesiones o la tarjeta de visita con las horas de las citas ya que con EasyPhysio toda la información será digitalizada y accesible en cualquier momento.

También se podrá olvidar de los mensajes SMS recordando las horas de las citas, cambios en citas por imprevistos o promociones del centro, porque toda esta información será cómodamente gestionada a través de notificaciones *push*.

Además, todo ello estará siempre visible en tan solo unos pocos toques en la pantalla del móvil o clics del ratón, pudiendo acceder a lo descrito anteriormente y también a otras muchas funcionalidades, por ejemplo, visualizar la agenda programada con sus citas; modificaciones realizadas por parte del centro; desplazamiento por parte del paciente de alguna cita; contratación de nuevas citas o bonos; ver el histórico de citas, etc.

Para validar el análisis y diseño *software* se implementará un prototipo de aplicación que contemple los casos de uso más relevantes descritos en el proyecto.

1.3. Metodología.

El proyecto ha sido planificado e implementado de manera iterativa e incremental a través de una versión simplificada de SCRUM en la cual los roles de *Product Owner* y el *SCRUM Master* han sido interpretados por el tutor del proyecto, y el equipo de desarrollo solo constó del alumno en cuestión.

Se llevaron a cabo reuniones de seguimiento en la cual se revisó el trabajo previo y las dificultades presentadas, para poder así planear una resolución para dichas dificultades y hacer una planificación para la siguiente reunión teniendo en cuenta todo lo anterior.

A continuación, se muestra la planificación original prevista adjuntada en el TFT01:

EasyPhysio

Tabla 1. Planificación TFT01.

Fases	Duración Estimada (horas)	Tareas (nombre y descripción, obligatorio al menos una por fase)
Estudio previo / Análisis	20	Tarea 1.1: Búsqueda de competencia en el mercado y comparativas entre ellas.
		Tarea 1.2: Comparativa de la competencia con las ideas expuestas para este proyecto, incluyéndose las funcionalidades que hacen destacarlo del resto
Diseño / Desarrollo / Implementación	200	Tarea 2.1: Desarrollo iterativo del prototipo para que satisfaga los casos de uso más representativos escogidos del análisis completo del proyecto
		Tarea 2.2: Puesta en común para elaborar un informe de correcciones de la actual iteración que será incluido en el lote de implementaciones de la siguiente iteración.
Evaluación / Validación / Prueba	40	Tarea 3.1: Codificación de pruebas unitarias.
Documentación / Presentación	40	Tarea 4.1: Elaboración del TFT01
		Tarea 4.2: Elaboración de la Memoria Final
		Tarea 4.3: Elaboración y preparación de la presentación

Sin embargo, debido a la alta cantidad de horas invertidas para el desarrollo y la baja relación de funcionalidades/tiempo empleado por la complejidad de haber generado un calendario y toda su gestión desde cero, se decidió no desarrollar pruebas unitarias y desviar la planificación horaria inicial para lograr tener un prototipo funcional con las características suficientes para poder ser una representación válida de los objetivos planteados por EasyPhysio.

A continuación, se muestra la planificación final:

EasyPhysio

Tabla 2. Planificación final.

Fases	Duración Estimada (horas)	Tareas (nombre y descripción, obligatorio al menos una por fase)
Estudio previo / Análisis	20	Tarea 1.1: Búsqueda de competencia en el mercado y comparativas entre ellas.
		Tarea 1.2: Comparativa de la competencia con las ideas expuestas para este proyecto, incluyéndose las funcionalidades que hacen destacarlo del resto
Diseño / Desarrollo / Implementación	240	Tarea 2.1: Desarrollo iterativo del prototipo para que satisfaga los casos de uso más representativos escogidos del análisis completo del proyecto
		Tarea 2.2: Puesta en común para elaborar un informe de correcciones de la actual iteración que será incluido en el lote de implementaciones de la siguiente iteración.
Documentación / Presentación	40	Tarea 4.1: Elaboración del TFT01
		Tarea 4.2: Elaboración de la Memoria Final
		Tarea 4.3: Elaboración y preparación de la presentación

1.4. Competencias específicas cubiertas.

La competencia específica TI06 “Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil” se ha conseguido cubrir gracias a la realización de este proyecto por diversos motivos explicados a continuación.

Para empezar, se ha llevado a cabo un profundo análisis de las necesidades de un mercado, para poder así idear y planificar una solución plasmada en una aplicación, englobando esta todos los aspectos necesarios para considerarse un producto final, listo para poder ser explotado.

Además, este análisis realizado en el proyecto es respaldado por la realización de un prototipo con los aspectos funcionales principales implementados, como puede ser la gestión de las citas por parte tanto de pacientes como de fisioterapeutas.

A pesar de que el prototipo no abarque todas las especificaciones técnicas ni funcionales para ser considerado un producto final, se puede apreciar que se podría conseguir dichos objetivos bajo la tutela del análisis realizado.

EasyPhysio

1.5. Aportaciones al entorno socioeconómico, técnico o científico.

Gracias a EasyPhysio, cualquier centro de fisioterapia humilde podrá permitirse tener una aplicación de escritorio y móvil con la que no solo poder gestionar su centro con todas sus funcionalidades, sino que también sus clientes podrán usar dicha aplicación y podrán sentir que no solo los centros más elitistas son capaces de permitirse algo similar.

Adicionalmente, las personas con disfunciones visuales tendrán una aplicación que se adapte a sus necesidades, permitiéndoles sentirse cómodos e independientes con su uso.

A nivel técnico, EasyPhysio podrá permitir la capacidad de realizar *rescheduling* automático, es decir, que gracias al procesamiento de la información pertinente, el sistema será capaz de detectar un cambio en la hoja de ruta de un tratamiento, bien por la incorporación, modificación o cancelación de una cita para dicho tratamiento, o bien porque con las evaluaciones realizadas por el fisioterapeuta, cree conveniente cambiar la planificación para conseguir una óptima recuperación del paciente; para poder así estimar un nuevo itinerario del tratamiento en base a las preferencias del paciente y las disponibilidades horarias de los fisioterapeutas involucrados.

1.6. Estructura de la memoria.

A lo largo de esta memoria se expondrán diversos aspectos del proyecto titulado EasyPhysio.

Primeramente, se redacta un bloque introductorio en el que se contextualiza la convergencia entre la Fisioterapia y la Informática; en el que se mide los objetivos a realizar y la metodología a seguir para su consecución; en el que se analiza las aportaciones que EasyPhysio es capaz de conseguir en diversos aspectos de la sociedad; y en el que se contrasta el trabajo realizado con la adquisición de competencias específicas establecidas por la institución pertinente.

Tras ello, se plasma las necesidades de un gran análisis previo a la realización técnica del proyecto, en el que se aglutinan el estudio de la competencia, el público objetivo del mercado, la toma de requisitos funcionales y no funcionales, un conjunto de *mockups* que reflejan las ideas iniciales, el modelo de negocio establecido y las normativas y legislaciones más relevantes que conciernen al proyecto, y que en su conjunto son la justificación de la existencia de EasyPhysio.

Seguidamente, se entra en cuestiones técnicas más específicas en las que se explica detalladamente el proceso de elaboración por iteraciones, las tecnologías empleadas a

EasyPhysio

lo largo de la vida del prototipo y el acceso público al repositorio que contiene el trabajo realizado.

Finalmente, se cierra la memoria con un conjunto de conclusiones en diversos aspectos, como el personal, el tecnológico referente al proyecto, de la consecución de algunos de los objetivos establecidos... Y la correspondiente bibliografía y anexos.

2. Análisis.

2.1. Estudio de aplicaciones similares.

Tras un exhaustivo análisis de la competencia, se decidió realizar dos tablas comparativas; una destinada a aplicaciones genéricas de gestión de un negocio de salud y belleza y otra destinada a aplicaciones de gestión centradas en el ámbito sanitario. De hecho, es reseñable señalar que, aunque éstas últimas sí incluyen la fisioterapia como parte del lote de servicios que ofrecen, no hay ninguna aplicación en el mercado destinado íntegra y exclusivamente a la gestión de un centro de fisioterapia *per se*.

En primera instancia, se procederá a destacar las fortalezas y debilidades del primer bloque de aplicaciones de la competencia, que son aquellas destinadas a la gestión de un comercio de salud y belleza:

Booksy es una aplicación con gran trayectoria, tiene una buena posición en el mercado gracias a su gran número de clientes habituales, debido seguramente a su informal, pero elegante diseño, y a su capacidad de poder reservar cita en el local, servicio, fecha y con el profesional que el usuario desee en tan solo unas pocas interacciones.

Sin embargo, y en comparación con otras alternativas, es probablemente la opción más cara de mantener para un negocio de pequeño calibre al tener que mantener mensualmente una suscripción.

Schedulicity posee un atractivo y reducido plan de micro suscripciones en base a los servicios requeridos por el negocio, cómodamente ajustables a lo largo del tiempo, que la convierte en una potente candidata entre los negocios más humildes.

En contra parte, es una aplicación que por el momento solo abarca Estados Unidos de América y Canadá.

Fresha es una aplicación de reservas con un coste económico bastante atractivo y competitivo, especialmente si la clientela del negocio es habitual, puesto que la mayor fuente de cobros a sus clientes es a través de comisiones por clientes nuevos.

No obstante, uno de los mayores inconvenientes que presenta esta aplicación es que no permite pago previo de las citas.

Tabla comparativa para aplicaciones de gestión para comercios de salud y belleza:

EasyPhysio

Tabla 3. Primera tabla comparativa. Aplicaciones de gestión para negocios de salud y belleza.

	Booksy (tarifa eco)	Schedulicity	Fresha	EasyPhysio
Precio	29€ (mensual)	Varía según servicios mensuales contratados	Gratuito (comisión por cliente nuevo + porcentaje por cada cita)	plan de negocio por definir
Nº clientes	Ilimitado	Ilimitado (20\$)	Ilimitado	Ilimitado
Nº usuarios	1	1	Ilimitado	Ilimitado
Servicio individualizado	No	No	No	Sí
Notificaciones push	Sí	Sí (5\$)	Sí	Sí
Prepago online de citas	Sí	Sí (5\$ + porcentajes variantes según servicio)	No	Sí
App para clientes	Sí	Sí	Sí	Sí
SO	Android / iOS / web	Android / iOS / web	Android / iOS / web	Android / iOS / web
Lanzamiento última versión	28/01/2021 (2.0_427)	20/01/2021 (0.7.8)	29/01/2021 (2.8.214)	
Desarrollador	Booksy	Schedulicity	Fresha	
Nº Descargas	500.000	100.000	50.000	

EasyPhysio

Posteriormente, se procede a destacar las fortalezas y debilidades del segundo bloque de aplicaciones de la competencia, que son aquellas destinadas a la gestión específicamente en el ámbito sanitario:

Archivex es la única de este grupo que posee una aplicación destinada a los pacientes para que puedan pedir cita.

Sin embargo, no pueden prepagar dicha cita, sino que el pago debe ser efectuado *in situ*, además de tener un costo adicional mensual.

Clinic Cloud es una aplicación de escritorio que en su plan básico ofrece 2 usuarios a la vez, a diferencia del resto del grupo comparativo, y adicionalmente con un precio inferior.

En contra parte, solo ofrece sus servicios a través de aplicación web.

NetClinicas es una aplicación interesante para un centro con muy pocos profesionales en un centro y con miras a tener una permanencia larga con sus servicios, puesto que el uso y disfrute de estos se consigue a través de un pago único.

No obstante, si se quiere gestionar uno o varios centros de gran tamaño, puede que el coste se dispare puesto que hay que adquirir una licencia por persona además de pagar asistencia por incidencias a partir de los 3 meses.

Tabla comparativa de aplicaciones centradas en el ámbito sanitario:

EasyPhysio

Tabla 4. Segunda tabla comparativa. Aplicaciones de gestión para el ámbito sanitario.

	Archivex (plan S)	Clinic Cloud (plan Mini)	NetClinicas (versión ELITE Monopuesto)	EasyPhysio
Precio	34,95€ (mensual)	29,65€ (mensual)	240 (pago único)	plan de negocio por definir
Nº pacientes	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Nº usuarios	1	2	1	Ilimitado
Reserva de citas	Sí	No	No	Sí
Notificaciones push	No	No	No	Sí
Prepago online de citas	No	No	No	Sí
App para pacientes	50€/mes adicional	No	No	Sí
SO	Android / iOS / web	Web	Web	Android / iOS / web
Lanzamiento última versión	29/01/2021 (2.4.72)			
Desarrollador	Xoborg	Clinic Cloud	NetClinicas Software	
Nº Descargas	1.000			

2.2. Requisitos.

Tras el estudio de la competencia en el mercado, se procede a desglosar el conjunto de requisitos necesarios para la consecución de los objetivos establecidos.

2.2.1. Tabla de *dummies*

A continuación, se plasmarán las características de los usuarios cuyas necesidades conllevará al uso de EasyPhysio, a través de una tabla de *dummies*.

Personas con lesiones: como cualquier centro de fisioterapia habitual, los pacientes en su gran mayoría son individuos con algún tipo de lesión, en el vasto sentido de la palabra, desde problemas articulares, hasta musculares, pasando por óseos o incluso psicomotrices.

EasyPhysio

Personas pertenecientes a la tercera edad: las personas mayores suelen ser personas con un hábito de continuidad fuerte y con alta dependencia en su día a día, es por ello una necesidad que otras personas sean capaces de establecer una prolongada planificación en sus sesiones de tratamiento en su nombre.

Deportistas: este colectivo, debido a su estilo de vida, acostumbra a prevenir antes que curar, y saben que la fisioterapia bien tutorizada, es aval a la hora de evitar riesgos por lesiones innecesarias. Además, cuando desgraciadamente caen en una lesión, son personas que requieren de un tratamiento acorde a la situación.

Personas con discapacidades visuales: usualmente, muchas aplicaciones tanto de escritorio como de móvil no son totalmente compatibles con lectores de pantallas, o no tienen un contraste suficientemente elevado en su contenido como para facilitar la navegación a personas con daltonismo.

Tabla 5. Tabla de dummies.

Nombre	Lesión	Deportista	Discapacidad visual	Tercera edad	Uso de EasyPhysio
Alex	Sí	No	Sí	Sí	2-3 días a la semana
Carlos	No	Sí	No	No	Bimensualmente
Felipe	Sí	Sí	No	No	Semanalmente
Gabriela	Sí	No	Sí	No	2 días a la semana
María	Sí	No	No	Sí	3 días a la semana

2.2.2. Requisitos funcionales.

A partir de la información extraída del análisis anterior, se deduce la siguiente lista de requisitos, los cuales son para una aplicación ideal y no todos se han implementado en el prototipo:

1. El administrador será capaz de realizar operaciones *CRUD* sobre los perfiles de fisioterapeutas.
2. El administrador será capaz de cambiar parámetros de configuración interna.
3. El fisioterapeuta será capaz de iniciar sesión en el aplicativo.
4. El fisioterapeuta será capaz de mantener activa su sesión en los dispositivos que así decida para futuras ocasiones.
5. El fisioterapeuta será capaz de cerrar sesión en los dispositivos que así decida.
6. El fisioterapeuta será capaz de añadir / modificar datos en su administración de perfil, como foto, información personal, cambiar contraseña...
7. El fisioterapeuta será capaz de visualizar su agenda de citas programadas.
8. El fisioterapeuta será capaz de visualizar las citas con sus pacientes de manera individualizada.
9. El fisioterapeuta será capaz de realizar operaciones *CRUD* sobre citas de sus pacientes.
10. El fisioterapeuta será capaz de solapar varios de sus pacientes en la misma franja horaria y que se vea reflejado en su agenda.

EasyPhysio

11. El fisioterapeuta será capaz de filtrar las citas de los pacientes en base al fisioterapeuta encargado de las mismas, pudiendo visualizar todas las citas del centro; además de poder ver cada una de ellas de manera individualizada.
12. El fisioterapeuta será capaz de realizar operaciones *CRUD* sobre citas en nombre de otros fisioterapeutas, basándose en los permisos que éstos le concedan y bajo la política de gestión de reservas marcada por el centro.
13. El fisioterapeuta será capaz de realizar actualizaciones de día y hora para sus citas o para las citas de otros fisioterapeutas -previa concesión de permiso-, directamente a través de la agenda.
14. El fisioterapeuta será capaz de crear una hoja de ruta para el tratamiento de un paciente, en la que se definirán los días de tratamiento, los tipos de terapia a realizar, la duración de las sesiones... Siendo modificable para poder adaptarse al paciente.
15. El fisioterapeuta será capaz de ver informes sobre cuántos pacientes ha atendido, qué días son los más ajetreados y cuáles son los más ociosos, ingresos pendientes...
16. El paciente será capaz de iniciar sesión en el aplicativo.
17. El paciente será capaz de mantener activa su sesión en un dispositivo para futuras ocasiones.
18. El paciente será capaz de cerrar sesión en un dispositivo.
19. El paciente será capaz de registrarse como usuario en el aplicativo.
20. El paciente será capaz de añadir / modificar datos en su administración de perfil, como foto, información personal, cambiar contraseña...
21. El paciente será capaz de visualizar su agenda de citas programadas.
22. El paciente será capaz de visualizar sus citas de manera individualizada.
23. El paciente será capaz de realizar operaciones *CRUD* sobre sus citas.
24. El paciente será capaz de visualizar el histórico de sus citas.
25. El paciente será capaz de filtrar las citas del centro en base al fisioterapeuta encargado de las mismas, pudiendo visualizar todas las citas del centro.
26. El paciente será capaz de realizar operaciones *CRUD* sobre citas en nombre de otros pacientes, basándose en los permisos que éstos le concedan.
27. El paciente será capaz de realizar actualizaciones de día y hora para sus citas o para las citas de otros pacientes -previa concesión de permiso-, directamente a través de la agenda.
28. El paciente será capaz de visualizar el estado de su bono de sesiones.
29. El paciente será capaz de contratar nuevos bonos de sesiones.
30. El paciente será capaz de realizar pagos a través de pasarela de pago segura.
31. El paciente será capaz de visualizar información adicional del centro, como otros servicios o información de interés.

2.2.3. Requisitos no funcionales.

1. El sistema será capaz de enviar correos electrónicos.
2. El sistema será capaz de cambiar entre diferentes idiomas, siendo el predeterminado el español
3. Toda funcionalidad del sistema y transacción de negocio debe responder en menos de 5 segundos al usuario.

EasyPhysio

4. El sistema poseerá un mínimo de seguridad.
5. Los datos de la aplicación provenientes de la base de datos deben ser actualizados para todos los usuarios con rapidez y asiduidad suficiente para que se eviten intentos de reservar simultáneamente una franja horaria de tratamiento.
6. El sistema tendrá diferentes roles de acceso al sistema, siendo el administrador el único capaz de alterar los permisos.
7. El sistema será capaz de calcular las vacantes a atribuir a una nueva cita de un paciente según la cantidad de minutos que el tipo de tratamiento requiera.
8. El sistema será capaz de enviar notificaciones.
9. El sistema será capaz de realizar *rescheduling* automático, es decir, que gracias al procesamiento de la información pertinente, el sistema será capaz de detectar un cambio en la hoja de ruta de un tratamiento, bien por la incorporación, modificación o cancelación de una cita para dicho tratamiento, o bien porque con las evaluaciones realizadas por el fisioterapeuta, cree conveniente cambiar la planificación para conseguir una óptima recuperación del paciente; para poder así estimar un nuevo itinerario del tratamiento en base a las preferencias del paciente y las disponibilidades horarias de los fisioterapeutas involucrados.
10. El sistema hará respaldo de la información almacenada, con una frecuencia configurable, en otro sistema de información diferente.
11. El sistema proporcionará mensajes de error que sean informativos y orientados al usuario final.
12. La aplicación debe poseer un diseño *responsive* para una correcta visualización de esta, independientemente del medio en el que se renderice.

2.2.4. Mockups.

A continuación, se procede a listar y describir una serie de *mockups* realizados previos a la implementación del prototipo y que son reflejo de las ideas que y características necesarias para la aplicación, en las cuales el diseño son un factor secundario y lo que prima es la usabilidad y experiencia del usuario.



The image shows a mobile application mockup for a registration form. The form is titled "Registrarse" and is set against a light blue background. It features several input fields: "Nombre" and "Apellidos" (Name and Surnames), "Correo electrónico" (Email), "Contraseña" (Password), and "Repetir contraseña" (Repeat password). Each password field has a small asterisk icon on the left. Below the input fields, there is a checkbox labeled "Acepto las Condiciones de uso y Políticas de privacidad" (I accept the Terms of use and Privacy Policies). At the bottom of the form is a large blue button labeled "Registrarse" (Register).

Figura 1. Formulario de registro.

EasyPhysio

La primera necesidad que tendrá el usuario al descargar la aplicación de móvil será la de crearse un usuario para poder reservar y gestionar citas. Para ello deberá cumplimentar el formulario de registro como se ve en la Figura 1.

Si el usuario accede a través de su versión web, lo primero que se encontrará no será el proceso de registro, será la información al respecto al centro, puesto que, al descargarse una aplicación móvil, el usuario ya sabe qué está descargando y el porqué; sin embargo, pueden ser diversos los motivos por los cuales ha conseguido acceder al dominio web y, por tanto, hay que contextualizar su experiencia.

Este procedimiento será algo distinto para los fisioterapeutas, ya que estos no pasarán por el formulario de creación de un usuario, sino que serán ingresados en el sistema por un administrador.

El formulario de inicio de sesión tiene un título "Iniciar sesión" en negrita. Debajo del título hay dos campos de entrada: "Correo electrónico" con un ícono de correo y "Contraseña" con un ícono de asterisco. Debajo de estos campos hay un checkbox con la etiqueta "Recuérdame" que está marcado. En la parte inferior del formulario hay un botón azul con el texto "Iniciar sesión". Al final del formulario, hay un enlace que dice "¿No tienes cuenta? [Créate una aquí](#)".

Figura 2. Formulario para iniciar sesión.

Si el usuario previamente ha creado un usuario no necesitará volver a realizar ese proceso, sino que podrá acceder directamente al aplicativo cumplimentando el formulario de inicio de sesión con sus credenciales, como se puede ver en la Figura 2. Además, tendrá la capacidad de mantener activa la sesión en el dispositivo si así lo desea.

EasyPhysio



Formulario de modificación de datos personales. El formulario está dividido en secciones. La parte superior tiene un fondo azul claro con un ícono de perfil y el texto "Nombre Apellidos" con un ícono de lápiz. Debajo, hay tres campos de texto con sus respectivos íconos de edición: "Correo electrónico:" (ejemplo@ejemplo.com), "Teléfono:" (123456789) y "Contraseña:" (*****). En la parte inferior, hay una barra de navegación con cinco íconos: Agenda, Histórico, Mis compras, Mi perfil (destacado en azul) e Información.

Figura 3. Formulario de modificación de datos personales.

Tras el registro, o en cualquier otro momento que se estime oportuno, el usuario tendrá la capacidad de cambiar los datos de su perfil aportados en el proceso de registro cumplimentando el formulario como se puede apreciar en la Figura 3.

Para un fisioterapeuta la usabilidad es la misma, lo único que cambia son los datos para mostrar y cuáles podrá modificar el profesional y cuáles no.

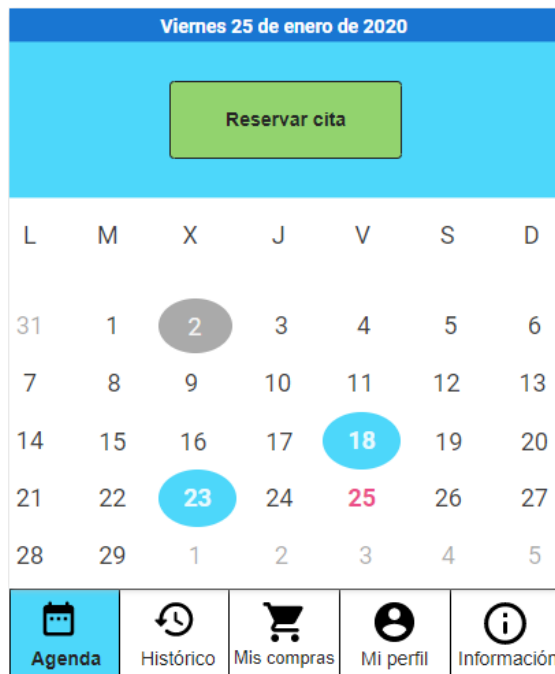


Figura 4. Agenda.

El pilar principal para la experiencia de los usuarios es la agenda, que es donde se muestran todas las citas que el usuario tiene asignadas, además de la capacidad de crear nuevas citas en espacios disponibles como se puede apreciar en la Figura 4.

EasyPhysio

La experiencia del usuario es esencialmente la misma tanto para pacientes como para fisioterapeutas, rigiéndose las posibles restricciones de reserva de nuevas en una política dictaminada por cada centro.



Figura 5. Agenda. Edición de cita.

Como se mencionó anteriormente, la experiencia principal del usuario es la capacidad de gestionar su agenda, por ejemplo, creando nuevas citas, pero también modificando las existentes.

Cada usuario tendrá la capacidad de modificar según sus necesidades sus citas asignadas, además de poder gestionar aquellas citas de otros usuarios sobre las cuales tengan permiso para hacerlo como se puede ver en la Figura 5.

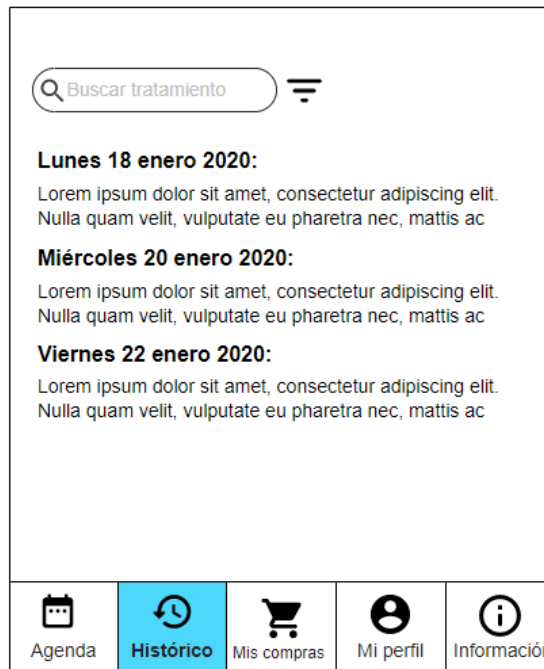


Figura 6. Histórico de citas.

Para el paciente es importante tener constancia de las citas a lo largo del tiempo y de manera cómoda, no teniendo que localizarlas manualmente en el conglomerado del calendario, y es por tanto necesario que tenga la capacidad de visualizarlas todas fácilmente y en un mismo lugar, e incluso que sea capaz de filtrarlas por algún tipo de parámetro como un rango de fechas. Esta posibilidad la puede localizar el paciente en la sección correspondiente, como se puede ver en la Figura 6.

Sin embargo, para el fisioterapeuta esta necesidad es bastante diferente, no necesita tener un registro de todas las citas que ha tenido, sino que necesita tener todo tipo de estadísticas como la cantidad de citas por semana que tiene, o qué día de la semana a lo largo de un año es el que más trabaja, o qué franja horaria es la más ociosa... Todo ello aglutinado en un cuadro de mando.

EasyPhysio

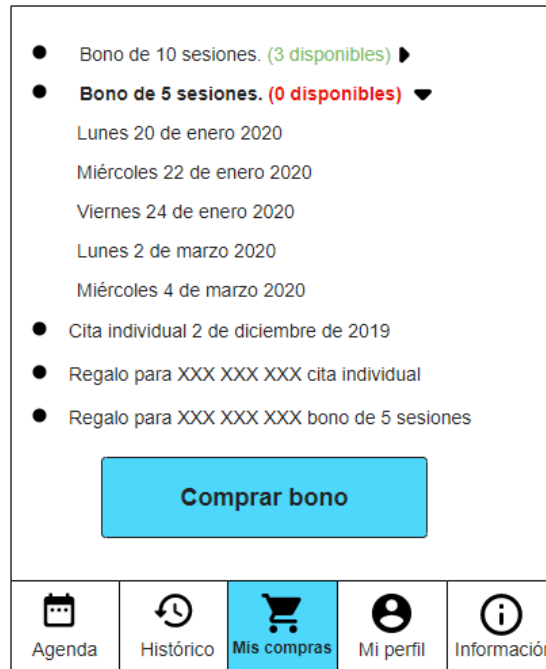


Figura 7. Compras realizadas.

Exclusivamente el paciente es capaz de comprar o regalar citas o bonos de sesiones, puesto que el fisioterapeuta es el profesional que ofrece los servicios, por lo que esta vista estaría disponible solo para pacientes, en la que se muestra el conjunto de compras o regalos realizados en su actividad en el aplicativo, mostrándose adicionalmente que bono de sesiones está todavía en activo o cuál ha sido totalmente gastado. Esta posibilidad la puede localizar el paciente en la sección correspondiente, como se puede apreciar en la Figura 7.

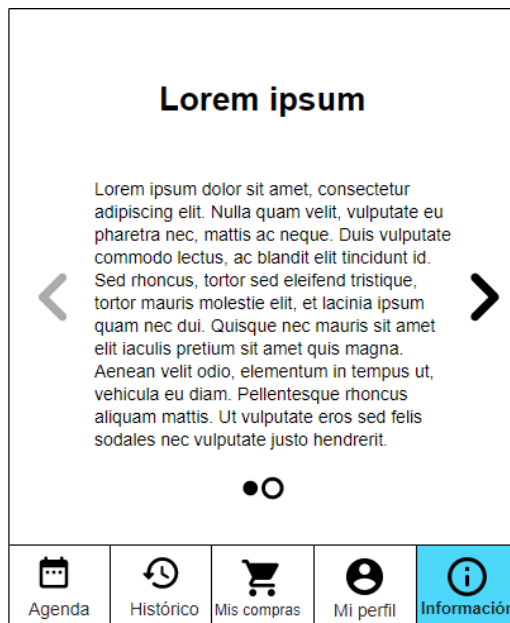


Figura 8. Información de interés.

EasyPhysio

Siempre que el usuario lo estime oportuno, podrá acceder al panel informativo del centro en el que se mostrará diversa información, como la localización u horario del centro, información divulgativa respecto a la fisioterapia, modificaciones en los servicios... como se puede ver en la Figura 8.

Como se puede apreciar en las distintas figuras referentes a los *mockups*, el color azul ha sido el predominante, elegido por sus cualidades entre las que se encuentran la serenidad, la confianza o la credibilidad y que es un referente habitual no solo entre la competencia, sino en cualquier negocio o incluso a lo largo de la historia, siendo el azul un representante de la gama cromática con mucho peso heráldico; sin embargo, *a posteriori* se decidió optar por una tríada de colores pastel, equidistantes entre sí y con armonía a la hora de combinarlos, con un elevado contraste entre fondos y letras para poder así facilitar la experiencia de los usuarios con algún tipo de daltonismo o de discapacidad visual.

2.3. Modelo de negocio.

A diferencia de la competencia, la cual en su mayoría ofrece sus servicios a través de métodos de suscripción u otros pocos establecen métodos de pago únicos por un producto, EasyPhysio planea ser gratuito tanto para los pacientes como para los profesionales y buscará financiarse a través de publicidad, de asistencia técnica y de donaciones contributivas de toda aquella persona que así lo estime oportuno.

Este modelo de negocio hará que la rentabilidad del servicio sea a largo plazo, puesto que, al no contar con unos ingresos elevados por cuotas o licencias, la táctica a utilizar es intentar atraer al máximo número de centros de fisioterapia posible para que cada aportación por publicidad sume su granito de arena.

2.4. Normativa y legislación.

La LOPDGDD (Ley de Protección de Datos personales y Garantía de Derechos Digitales, [BOE-A-2018-16673](#)) es la adaptación española de la ley europea RGPD (Reglamento General de Protección de Datos). Es muy importante tanto para EasyPhysio como para sus clientes el entendimiento y adaptaciones de los recursos necesarios para cumplir las directrices propuestas por dicha ley puesto que continuamente se recopila, almacena y procesa información personal de los pacientes y el personal, siendo además específicamente delicados los datos sanitarios de los pacientes al tratarse de datos con una alta protección.

3. Iteraciones.

3.1. Alcance de la implementación.

Anteriormente se establecieron unos objetivos y se realizaron una serie de análisis que son el marco bajo el que se resguarda EasyPhysio; sin embargo, debido a las limitaciones a las que el proyecto se somete, solo ha sido posible realizar un prototipo que, no obstante, ha conseguido validar algunos de las ideas principales citadas previamente.

Algunos de los requisitos listado anteriormente y que se han podido implementar al menos parcialmente en este prototipo son:

1. El fisioterapeuta será capaz de iniciar sesión en el aplicativo.
2. El fisioterapeuta será capaz de mantener activa su sesión en los dispositivos que así decida para futuras ocasiones.
3. El fisioterapeuta será capaz de cerrar sesión en los dispositivos que así decida.
4. El fisioterapeuta será capaz de visualizar su agenda de citas programadas.
5. El fisioterapeuta será capaz de visualizar las citas con sus pacientes de manera individualizada.
6. El fisioterapeuta será capaz de realizar operaciones *CRUD* sobre citas de sus pacientes.
7. El fisioterapeuta será capaz de solapar varios de sus pacientes en la misma franja horaria y que se vea reflejado en su agenda.
8. El fisioterapeuta será capaz de filtrar las citas de los pacientes en base al fisioterapeuta encargado de las mismas, pudiendo visualizar todas las citas del centro; además de poder ver cada una de ellas de manera individualizada.
9. El fisioterapeuta será capaz de realizar operaciones *CRUD* sobre citas en nombre de otros fisioterapeutas, basándose en los permisos que éstos le concedan y bajo la política de gestión de reservas marcada por el centro.
10. El fisioterapeuta será capaz de realizar actualizaciones de día y hora para sus citas o para las citas de otros fisioterapeutas -previa concesión de permiso-, directamente a través de la agenda.
11. El fisioterapeuta será capaz de ver informes sobre cuántos pacientes ha atendido, qué días son los más ajetreados y cuáles son los más ociosos, ingresos pendientes...
12. El paciente será capaz de iniciar sesión en el aplicativo.
13. El paciente será capaz de mantener activa su sesión en un dispositivo para futuras ocasiones.
14. El paciente será capaz de cerrar sesión en un dispositivo.
15. El paciente será capaz de registrarse como usuario en el aplicativo.
16. El paciente será capaz de visualizar su agenda de citas programadas.
17. El paciente será capaz de visualizar sus citas de manera individualizada.
18. El paciente será capaz de realizar operaciones *CRUD* sobre sus citas.
19. El paciente será capaz de filtrar las citas del centro en base al fisioterapeuta encargado de las mismas, pudiendo visualizar todas las citas del centro.
20. El paciente será capaz de realizar operaciones *CRUD* sobre citas en nombre de otros pacientes, basándose en los permisos que éstos le concedan.

EasyPhysio

21. El paciente será capaz de realizar actualizaciones de día y hora para sus citas o para las citas de otros pacientes -previa concesión de permiso-, directamente a través de la agenda.
22. El paciente será capaz de visualizar información adicional del centro, como otros servicios o información de interés.

3.2. Iteración 1.

3.2.1. Requisitos.

Durante la primera iteración se consiguieron implementar satisfactoriamente los siguientes requisitos:

1. El fisioterapeuta será capaz de visualizar su agenda de citas programadas.
2. El fisioterapeuta será capaz de visualizar las citas con sus pacientes de manera individualizada.
3. El fisioterapeuta será capaz de realizar operaciones *CRUD* sobre citas de sus pacientes.
4. El fisioterapeuta será capaz de solapar varios de sus pacientes en la misma franja horaria y que se vea reflejado en su agenda.
5. El fisioterapeuta será capaz de realizar actualizaciones de día y hora para sus citas o para las citas de otros fisioterapeutas -previa concesión de permiso-, directamente a través de la agenda.
6. El paciente será capaz de visualizar su agenda de citas programadas.
7. El paciente será capaz de visualizar sus citas de manera individualizada.
8. El paciente será capaz de realizar operaciones *CRUD* sobre sus citas.
9. El paciente será capaz de realizar actualizaciones de día y hora para sus citas o para las citas de otros pacientes -previa concesión de permiso-, directamente a través de la agenda.

3.2.2. Diagrama de casos de uso.

Para esta iteración, el diagrama de casos de uso es el siguiente:

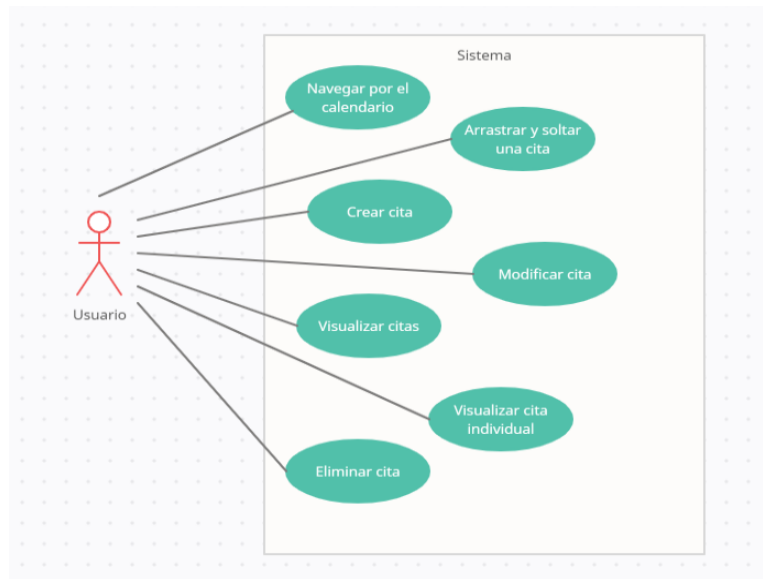


Figura 9. Diagramas de casos de uso iteración 1.

Como se puede apreciar en la Figura 9, durante esta iteración se han conseguido implementar diversos casos de uso, siendo todos ellos vinculados a la gestión de citas y del calendario.

Lo primordial para poder tener un calendario funcional de reserva de citas es, precisamente, poder tener un calendario navegable en el tiempo y que este sea interactivo para poder crear nuevas citas en las franjas horarias que el usuario crea conveniente.

Además, posteriormente, se debe permitir al usuario poder visualizar las citas, ver los detalles individuales de cada cita y si así lo estima oportuno, editar o borrar la cita pertinente.

Por otro lado, el usuario es capaz de interactuar con las citas del calendario no solo pinchando en ellas para ver los detalles y poder así modificarla, sino que adicionalmente, el usuario tiene la capacidad de poder arrastrar alguna cita desde una franja horaria origen y soltarla en otra franja horaria destino diferente para modificar el día y hora a su elección *ipso facto*.

3.2.3. Mockups.

Para esta iteración se utilizaron como referencia los siguientes mockups:

EasyPhysio



Figura 10. Agenda.

El pilar principal para la experiencia de los usuarios es la agenda, que es donde se muestran todas las citas que el usuario tiene asignadas, además de la capacidad de crear nuevas citas en espacios disponibles como se puede apreciar en la Figura 10.

La experiencia del usuario es esencialmente la misma tanto para pacientes como para fisioterapeutas, rigiéndose las posibles restricciones de reserva de nuevas en una política dictaminada por cada centro.



Figura 11. Agenda. Edición de cita.

Como se mencionó anteriormente, la experiencia principal del usuario es la capacidad de gestionar su agenda, por ejemplo, creando nuevas citas, pero también modificando las existentes.

EasyPhysio

Cada usuario tendrá la capacidad de modificar según sus necesidades sus citas asignadas, además de poder gestionar aquellas citas de otros usuarios sobre las cuales tengan permiso para hacerlo como se puede ver en la Figura 11Figura 5.

3.2.4. Diseño arquitectónico.

El prototipo está implementado en Angular, por tanto, el patrón de arquitectura *software* utilizado es MVVM (Modelo-Vista-Modelo de Vista en español) en el que destaca el concepto de *two-way binding* el cual permite realizar cambios en el modelo de vista desde la vista y viceversa.

Las 3 definiciones clave de patrón son las siguientes:

El modelo

Representa la lógica de negocio, por tanto, solo son datos, sin acciones ni manipulaciones y es totalmente independiente de la vista.

La vista

Representa visualmente los datos, sin embargo, en este patrón las vistas son activas, es decir, que a través de enlaces de datos tienen conocimiento del modelo.

Modelo de vista

Representa una abstracción de la interfaz, hace de intermediario entre modelo y vista y es con este modelo con quien se comunica la vista a través de enlaces de datos.

Durante el desarrollo de esta iteración, se crearon las diferentes interfaces TypeScript con el que poder tipar los datos a manejar y son representación del modelado de datos.

En cuanto a módulos, los más relevantes durante esta iteración son los siguientes:

SchedulerComponent

Es el componente principal de la aplicación y se encarga de mostrar al usuario el calendario con las citas pertinentes además de mostrar las opciones de navegabilidad a través de las semanas, meses o años con su reflejo en dicho calendario.

Por un lado, las franjas horarias muestran diferentes estilos y diferentes capacidades de interacción con el usuario gracias al procesamiento del calendario por parte de la tubería "CalculateWorkHoursPipe".

Adicionalmente, las citas son mostradas en su franja horaria correspondiente gracias a la interacción de este componente con la tubería "appointmentsPipe".

Por otro lado, cada franja horaria posee la capacidad de generar una ventana modal a través del componente "modalComponent" para poder crear una nueva cita en dicha ventana temporal.

EasyPhysio

Además, las citas mostradas en el calendario pueden ser arrastradas desde su franja horaria inicial y dejadas en cualquier franja válida con su correspondiente reflejo en la base de datos gracias a la intervención del servicio “appointmentService”.

ModalComponent

Es el componente encargado de mostrar al usuario una ventana modal con un formulario a rellenar para poder crear una nueva cita; o bien, mostrar el mismo formulario con los datos precargados de la cita con la que ha interactuado y poder así editar o borrar dicha cita.

Cada vez que el usuario genera una nueva cita, o edita o borra una existente, se refleja dicha acción en la base de datos gracias a la intervención del servicio “appointmentService”.

AppointmentsPipe

Es la tubería encargada de procesar las citas del centro para mostrar en cada franja horaria las citas correspondientes.

CalculateWorkHoursPipe

Es la tubería encargada de procesar los horarios de trabajo definidos por el centro y poder mostrar correctamente al usuario qué franjas horarias del calendario están disponibles para poder crear nuevas citas, y cuáles no.

AppointmentService

Es el servicio encargado de establecer conexión con el servidor de datos para poder así realizar las operaciones CRUD sobre el recurso “citas” desde el aplicativo.

ToastContainerComponent

Es el componente encargado de mostrar al usuario el resultado de sus interacciones a lo largo de la aplicación, como puede ser la confirmación de que una cita nueva se ha creado, o que una cita se ha borrado correctamente.

Estas notificaciones pueden ser generadas directamente desde el componente “schedulerComponent” a través de arrastrar y dejar una cita existente de una franja horaria inicial a otra diferente o bien pueden ser creadas por el componente “modalComponent” tras la pertinente acción del usuario para crear, borrar o editar una cita.

ToastService

Es el servicio encargado de proveer a “ToastContainerComponent” de los datos necesarios a mostrar, generados por algunas de las acciones del usuario con la aplicación y de remover aquellas notificaciones que han expirado.

EasyPhysio

Cada uno de los módulos descritos anteriormente implementa sus correspondientes métodos. A continuación, los más representativos van a ser brevemente descritos:

onSubmitAppointment (modal.component.ts)

Se encarga de instanciar los datos necesarios para posteriormente comunicarse con el servicio “appointmentService” y poder crear una nueva cita, o bien modificar o borrar una ya existente.

dropAppointment (scheduler.component.ts)

Se encarga de gestionar el evento resultante de la acción arrastrar y dejar por parte del usuario sobre una cita, con origen una franja horaria y destino otra diferente.

openModalEditDelete (scheduler.component.ts)

Se encarga de llamar al componente “modalComponent” tras la interacción del usuario con una cita para precargar la ventana modal con los datos de dicha cita y poder así editarla o borrarla.

Si el usuario ha decidido editar o borrar la cita, este método se encarga de procesar qué decisión ha tomado para llamar a la función correspondiente.

openModalCreate (scheduler.component.ts)

Se encarga de llamar al componente “modalComponent” para mostrar al usuario una ventana modal con un formulario de creación de nueva cita cuando este ha interactuado con alguna franja horaria.

Si el usuario ha decidido crear una nueva cita y ha rellenado correctamente el formulario correspondiente, este método se encarga de procesar la interacción para llamar a la función correspondiente.

openModalGroupedAppointments (scheduler.component.ts)

Se encarga de llamar al componente “modalComponent” tras la interacción del usuario con una lista de citas para precargar la ventana modal con los datos de dichas citas y poder así editar o borrar cualquiera de las presentadas.

Si el usuario ha decidido editar o borrar alguna de las citas, este método se encarga de procesar qué decisión ha tomado para llamar a la función correspondiente.

populateWeekDays (scheduler.component.ts)

Se encarga de popular los días a mostrar en el calendario, dependiendo de la interacción del usuario con este.

3.2.5. Modelado de datos.

El modelado de datos se ha reflejado en un diagrama de entidad-relación, que se muestra a continuación:

EasyPhysio

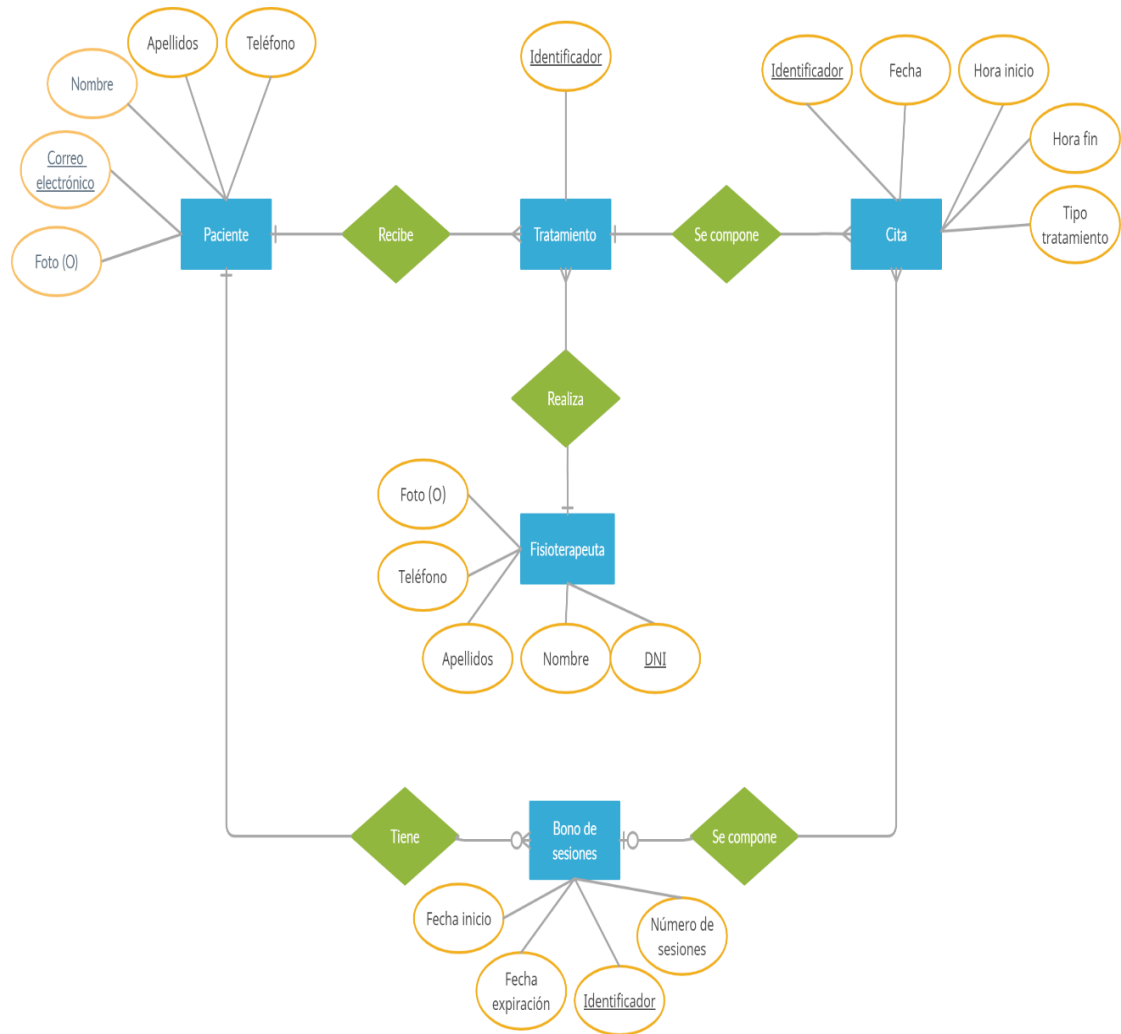


Figura 12. Diagrama de entidad-relación.

Como se puede ver en la Figura 12, el modelo de datos refleja la realidad de un centro de fisioterapia, puesto que está centrado en los tratamientos que los pacientes reciben gracias a la realización de dichos tratamientos por parte de los fisioterapeutas y que, según la complejidad del caso, se conforman por una o múltiples citas.

Por otro lado, y con menor peso en las relaciones en la entidad se puede apreciar que existen los bonos de sesiones, que los pacientes pueden comprar y que según la compra realizada, estos se componen de un diverso número de citas.

3.2.6. Detalles de implementación.

Durante la realización de esta iteración, se presentó la tesitura de implementar un diseño acoplado del recurso "citas", puesto que se buscó aprovechar el establecimiento de una matriz tridimensional para la gestión de las citas en el calendario con una versión tridimensional análoga de los datos. Sin embargo, se descartó porque a pesar de tener ambas estructuras una forma tridimensional, estas no casaban debido a que no se podía aparejar los índices de una y otra estructura y, por tanto, se optó por un diseño

EasyPhysio

desacoplado de los datos, los cuales *a posteriori* son procesados gracias a la tubería “AppointmentsPipe”, previamente explicada.

Por un lado, y en base a las ventajas competitivas establecidas previamente referidas a la alta comodidad para personas con discapacidades visuales, se decidió generar una plantilla y usar unos colores para el aplicativo que cumplieren con las directrices WCAG, y para poder corroborar las buenas prácticas llevadas a cabo, se instaló la extensión “WCAG color contrast checker” para Google Chrome.

Por otro lado, y debido a la alta complejidad y desempeño de horas necesario para elaborar desde cero un calendario y su correcta manipulación e integración con datos provenientes de un servidor, fue estipulado conveniente dejar la posibilidad de cambiar el modo de vista del calendario en formato mensual a otras versiones de formato como el semanal o el diario en modo estático y como objeto meramente estético, para poder conseguir implementar otras funciones establecidas como más importantes para este prototipo.

Adicionalmente, durante la implementación del módulo “Drag&Drop” de Angular Material, se pretendía editar el cursor durante la acción de arrastre de una cita por parte del usuario para que este representase el estado actual de la misma, pero el resultado final no es fiable debido a un bug sin arreglo oficial, dependiendo de *workarounds* y aquí reportado: <https://github.com/react-dnd/react-dnd/issues/476>.

Como se puede comprobar, en esta iteración se implementaron notificaciones al usuario cuando este realizase algunas acciones determinadas como crear una nueva cita; sin embargo, al no contar con un *backend* personalizado, no se ha realizado la versión análoga en el lado del servidor, que en un sistema real sería necesario para llevar a cabo diversas estadísticas o para poder realizar trazas de errores, etc.

Cabe destacar que todos los recursos son accesibles a través de CDN y no están descargados de forma interna en la aplicación. No se ha hecho uso de *webworkers* ni el manifiesto debido a que el prototipo implementado no dispone de servicio offline ni de la cuantiosa cantidad de tiempo que sería necesario invertir.

3.3. Iteración 2.

3.3.1. Requisitos.

Durante la segunda iteración se consiguieron implementar satisfactoriamente los siguientes requisitos:

1. El fisioterapeuta será capaz de iniciar sesión en el aplicativo.
2. El fisioterapeuta será capaz de cerrar sesión en los dispositivos que así decida.
3. El fisioterapeuta será capaz de filtrar las citas de los pacientes en base al fisioterapeuta encargado de las mismas, pudiendo visualizar todas las citas del centro; además de poder ver cada una de ellas de manera individualizada.

EasyPhysio

4. El fisioterapeuta será capaz de realizar operaciones *CRUD* sobre citas en nombre de otros fisioterapeutas, basándose en los permisos que éstos le concedan y bajo la política de gestión de reservas marcada por el centro.
5. El paciente será capaz de iniciar sesión en el aplicativo.
6. El paciente será capaz de cerrar sesión en un dispositivo.
7. El paciente será capaz de registrarse como usuario en el aplicativo.
8. El paciente será capaz de filtrar las citas del centro en base al fisioterapeuta encargado de las mismas, pudiendo visualizar todas las citas del centro.
9. El paciente será capaz de realizar operaciones *CRUD* sobre citas en nombre de otros pacientes, basándose en los permisos que éstos le concedan.

3.3.2. Diagrama de caso de uso.

Para esta iteración, el diagrama de casos de uso es el siguiente:

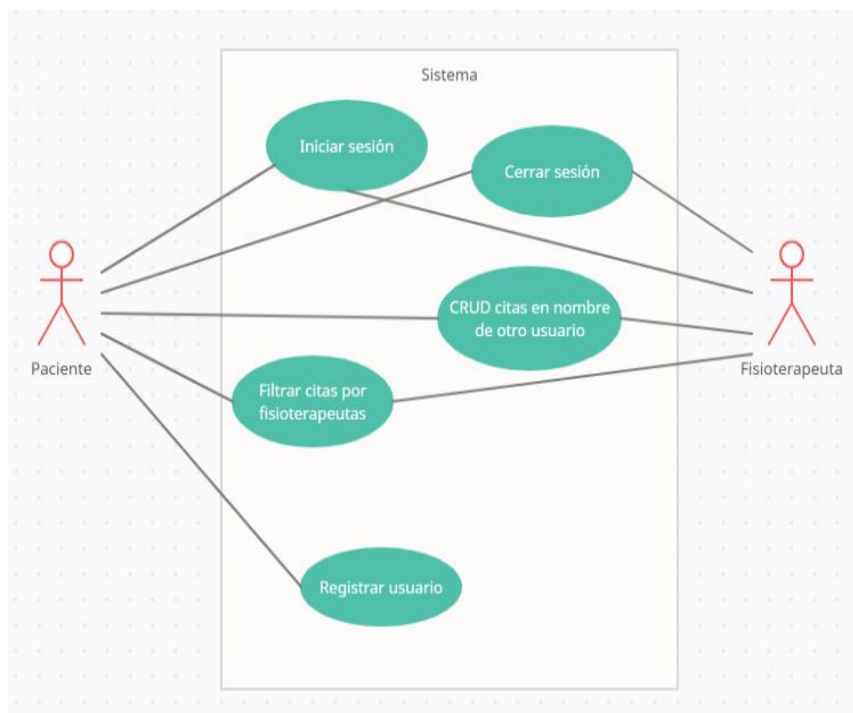


Figura 13. Diagramas de casos de uso iteración 2.

Como se puede apreciar en la Figura 13, durante esta iteración se han conseguido implementar diversos casos de uso.

Estos se dividen en tres bloques principales:

La gestión de la sesión del usuario, en la que destacan la creación de nuevos pacientes a través del formulario de registro; la capacidad de iniciar sesión a través del formulario de inicio de sesión tanto para pacientes como fisioterapeutas; y la capacidad de cerrar sesión del aplicativo tanto para pacientes como fisioterapeutas. Cabe mencionar que la creación de nuevos fisioterapeutas solo es posible a través de la consola de administración y solo el personal autorizado es capaz de hacerlo.

El filtrado de citas a mostrar en el calendario, basado en el fisioterapeuta que las realiza.

EasyPhysio

La posibilidad de crear, visualizar, editar y borrar citas en nombre de otros pacientes cuando el usuario activo es un paciente, y para otros fisioterapeutas cuando el usuario activo es un fisioterapeuta. Cabe destacar que para poder realizar alguna o todas de las acciones mencionadas, es necesario que el usuario implicado provea la concesión de permisos pertinentes a cada una de las acciones.

3.3.3. Mockups.

Para esta iteración se utilizaron como referencia los siguientes mockups:

The image shows a mobile app registration form titled "Registrarse". It features a back arrow icon at the top left. The form contains several input fields: "Nombre" and "Apellidos" (two separate boxes), "Correo electrónico" (with an envelope icon), "Contraseña" (with a star icon), and "Repetir contraseña" (with a star icon). Below the fields is a checkbox labeled "Acepto las [Condiciones de uso](#) y [Políticas de privacidad](#)". At the bottom is a large blue button labeled "Registrarse".

Figura 14. Formulario de registro.

La primera necesidad que tendrá el usuario al descargar la aplicación de móvil será la de crearse un usuario para poder reservar y gestionar citas. Para ello deberá cumplimentar el formulario de registro como se ve en la Figura 14.

Si el usuario accede a través de su versión web, lo primero que se encontrará no será el proceso de registro, será la información al respecto al centro, puesto que, al descargarse una aplicación móvil, el usuario ya sabe qué está descargando y el porqué; sin embargo, pueden ser diversos los motivos por los cuales ha conseguido acceder al dominio web y, por tanto, hay que contextualizar su experiencia.

Este procedimiento será algo distinto para los fisioterapeutas, ya que estos no pasarán por el formulario de creación de un usuario, sino que serán ingresados en el sistema por un administrador.

Créate una aquí'." data-bbox="325 81 664 322"/>

Figura 15. Formulario para iniciar sesión.

Si el usuario previamente ha creado un usuario no necesitará volver a realizar ese proceso, sino que accederá directamente al aplicativo con sus credenciales como se puede apreciar en la Figura 15.

3.3.4. Diseño arquitectónico.

En cuanto a módulos, los más relevantes durante esta iteración son los siguientes:

MatchPasswordDirective

Se encarga de hacer de pasarela entre el *template* y el servicio “CustomValidationService” para validar la información introducida por el usuario en los formularios en los que haga falta comprobar que dos campos de contraseña son iguales. Adicionalmente se implementaron otras directivas para sus respectivos campos: “DnipatternDirective”, “EmailPatternDirective”, “PasswordPatternDirective”, “RequiredDateAppointmentDirective” y “SelectPatientFromListDirective”.

AuthLoggedGuard

Guarda que se encarga de comprobar que el usuario está *logeado* y por tanto puede acceder a la dirección pertinente, o si bien, si no está *logeado*, redirigirlo al formulario de inicio de sesión.

AuthNotLoggedGuard

Guarda que se encarga de comprobar que el usuario no está *logeado* y por tanto puede acceder a los formularios de registro y de inicio de sesión correctamente, o si bien, si está *logeado*, redirigirlo al calendario principal.

LoginComponent

EasyPhysio

Es el componente encargado de mostrar el formulario de inicio de sesión y de enviar la información cumplimentada al servicio “FirebaseAuthService” para que este se comunique con el servidor.

SignupComponent

Es el componente encargado de mostrar el formulario de registro de un nuevo usuario y de enviar la información cumplimentada al servicio “FirebaseAuthService” para que este se comunique con el servidor.

EditAppointmentByLoggedInUserPipe

Es la tubería encargada de comprobar si el usuario activo tiene la potestad de hacer la acción arrastrar y dejar sobre cada cita del calendario.

CheckOtherPatientsPermissionsPipe

Es la tubería encargada de comprobar qué permisos de acciones sobre las citas de otros pacientes tiene el paciente activo.

CheckOtherPhysiosPermissionsPipe

Es la tubería encargada de comprobar qué permisos de acciones sobre las citas de otros fisioterapeutas tiene el fisioterapeuta activo.

CustomValidationService

Es el servicio encargado de recibir los valores introducidos por el usuario en los diferentes formularios del aplicativo y de validarlos según corresponda para cada tipo de campo. En caso de encontrar errores se devuelven duplas código:*boolean* para ser procesados por la aplicación; en caso de no existir errores no devuelve nada.

FirebaseAuthService

Es el servicio encargado de establecer conexión con el servidor de datos para procesar la información rellena por el usuario de los formularios de registro y de inicio de sesión.

PatientService

Es el servicio encargado de establecer conexión con el servidor de datos para poder así realizar las operaciones CRUD sobre el recurso “pacientes” desde el aplicativo. De forma análoga se implementaron los servicios “RoomService” y “PhysioService” para encargarse respectivamente del recurso “habitaciones” y “fisioterapeutas”.

SchedulerComponent

Se modificó el componente para poder filtrar las citas del centro en base a los fisioterapeutas de este, o bien por las citas de uno mismo si el usuario activo es un paciente. Para poder distinguir de qué fisioterapeuta es cada cita, se modificó la interfaz pertinente para añadir un código de color, el cual es insertado dinámicamente en la

EasyPhysio

aplicación. Para las citas propias de un paciente, se ha mezclado dicho código de color y el color negro, para poder así diferenciar claramente unas citas de otras.

Además, se implementó un botón para que el usuario pueda cerrar sesión de la aplicación correctamente.

ModalComponent

Se modificó el componente para que el paciente activo pueda crear nuevas citas para otros pacientes -con previa concesión de permiso-, o bien, para que pueda modificar o borrar citas existentes de otros pacientes -con previa concesión de permiso-.

Adicionalmente, se hizo lo análogo para un fisioterapeuta respecto a otros fisioterapeutas.

Cabe resaltar que, aunque son capacidades similares, se implementó con algunas diferencias, entre las que destaca que un fisioterapeuta busca pacientes a través de un *autocomplete* en el cual el sistema le sugiere nombres según escribe; y selecciona a qué fisioterapeuta va a crear la cita a través de una lista desplegable de valores. El paciente para ambos casos se le muestra una lista desplegable de valores, puesto que un paciente no debe poder buscar a otros pacientes que no sean aquellos que le han dado permiso para poder crear nuevas citas en su nombre.

Por otro lado, mencionar que se añadieron las siguientes comprobaciones en la creación y edición de citas: no poder crear dos citas en la misma franja horaria para el mismo paciente; no poder tener dos citas en la misma sala en la misma franja horaria; no poder reservar más citas para un fisioterapeuta cuyo cupo esté completo para una franja horaria; no poder reservar una cita para un fisioterapeuta que no trabaje ese día; y no poder crear una cita para un fisioterapeuta que trabaje ese día pero no a la hora seleccionada.

AppRoutingModule

Se modificó la gestión de rutas para que el usuario pueda navegar entre el formulario de registro y de inicio de sesión y bloquear el acceso al calendario principal si no existe un usuario activo, comprobado a través de la guarda "AuthLoggedGuard". De forma análoga se permite el acceso al calendario principal si existe un usuario activo y se bloquea la navegación al formulario de registro y de inicio de sesión a través de la guarda "AuthNotLoggedGuard".

Adicionalmente, se estipuló que el formulario de inicio de sesión fuese la página para mostrar cuando un usuario accede al dominio y que esta fuese la vista a mostrar si se interactuaba con el navegador con cualquier ruta no válida para el acceso, a excepción de cuando existiese un usuario activo, que la página por defecto a mostrar tras cualquier interacción inválida del usuario con el navegador es su calendario de citas.

EasyPhysio

Cada uno de los módulos descritos anteriormente implementa sus correspondientes métodos. A continuación, los más representativos van a ser brevemente descritos:

onSignUp (signup.component.ts)

Es el método encargado de llamar al servicio “FirebaseAuthService” para crear un nuevo usuario en el sistema con los datos rellenos en el formulario de registro.

Adicionalmente, En caso de producirse algún error como que el correo electrónico introducido ya está registrado, se genera el correspondiente mensaje de error para notificar al usuario.

onSignIn (login.component.ts)

Es el método encargado de llamar al servicio “FirebaseAuthyService” para iniciar sesión por parte de un usuario que ha relleno el formulario correspondiente. Se comprueba que los datos introducidos son correspondientes o bien a un fisioterapeuta o bien a un paciente para proceder a procesar dicha información.

Adicionalmente, en caso de producirse algún error como haber introducido incorrectamente la contraseña, se genera el correspondiente mensaje de error para notificar al usuario.

physioGridColor (scheduler.component.ts)

Es el método encargado de procesar el atributo de color del JSON de cada modelo para los fisioterapeutas para poder mostrar en la aplicación los colores correspondientes a cada profesional.

Adicionalmente, si el usuario activo es un paciente, al color de cada uno de sus citas se la añade a la franja de color correspondiente al fisioterapeuta en cuestión, una franja de color negra para poder distinguirla del resto de citas del calendario.

changeFilterPhysios (scheduler.component.ts)

Es el método encargado de añadir o remover del filtrado al fisioterapeuta con el que el usuario ha interactuado en los filtros situados en el calendario.

3.3.5. Detalles de implementación.

Durante la realización de esta iteración se encontró con la necesidad de tener un *backend*, puesto que el contenedor web de Webpack que viene integrado en Angular para poder desplegar la aplicación y poder hacer pruebas es muy completo, pero no deja de ser un envoltorio no apto para entornos de producción. Por lo que para poder realizar implementaciones más elaboradas para los recursos a almacenar en bases de datos era necesario implementar un *backend*.

Debido a la falta de tiempo para poder implementar un servidor decente con el cual poder satisfacer las necesidades del prototipo, se decidió escoger Firebase como servidor de datos, puesto que no requiere de apenas conocimientos ni configuraciones

EasyPhysio

para poder tener todo lo necesario para desarrollar satisfactoriamente el prototipo. Sin embargo, esto conllevó a que hubo que cambiar ligeramente los servicios e interfaces implementados hasta el momento puesto que ahora las peticiones no iban a través del módulo de HTTP de Angular, sino a través de los módulos de Angular Firebase.

Debido a que la creación de fisioterapeutas solo es permitida de manos de un usuario administrador, para poder añadir nuevos fisios, se debe hacer manualmente a través de la consola de Firebase. Además, como es posible que un mismo usuario pueda ser fisioterapeuta y cliente (según políticas de algunas empresas, puede darse el caso de tener descuento en servicios de la corporación, por ejemplo), se decidió que los fisioterapeutas y los pacientes accedan al aplicativo a través de la misma pantalla.

Por un lado, en esta iteración se añadieron a la pila de tarea algunos de los bugs generados durante la iteración anterior, como la incorrecta gestión de fechas en el calendario, en la cual independientemente del mes en el que el usuario creaba una nueva cita, se generaba en el mes actual; o como la incorrecta gestión de datos inválidos introducidos en el modal de creación o edición de citas, que resultaba en datos de citas incongruentes.

Por otro lado, debido a las nuevas implementaciones de esta iteración, en las que se permitió a los usuarios crear, editar o borrar citas para otros usuarios -según concesión de permisos previos- conllevó a que los modelos de datos e interfaces de TypeScript fuesen ampliados con los atributos necesarios para poder gestionar dichas funcionalidades.

3.4. Iteración 3.

3.4.1. Requisitos.

Durante la tercera iteración se consiguieron implementar satisfactoriamente los siguientes requisitos:

1. El fisioterapeuta será capaz de mantener activa su sesión en los dispositivos que así decida para futuras ocasiones.
2. El fisioterapeuta será capaz de ver informes sobre cuántos pacientes ha atendido, qué días son los más ajetreados y cuáles son los más ociosos, ingresos pendientes...
3. El paciente será capaz de mantener activa su sesión en un dispositivo para futuras ocasiones.
4. El paciente será capaz de visualizar información adicional del centro, como otros servicios o información de interés.

3.4.2. Diagrama de caso de uso.

Para esta iteración, el diagrama de casos de uso es el siguiente:

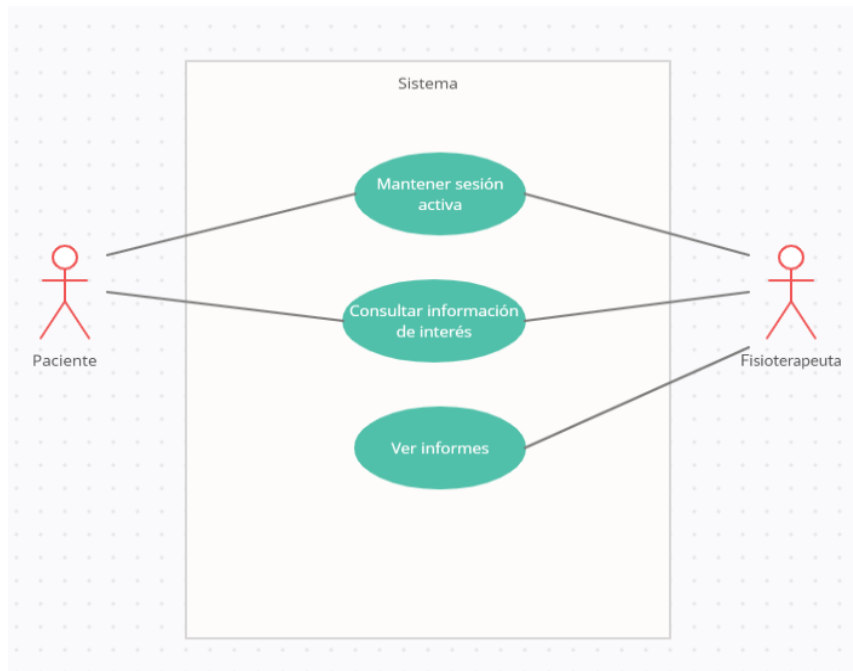


Figura 16. Diagramas de casos de uso iteración 3.

Como se puede apreciar en la Figura 16, durante esta iteración se han conseguido implementar diversos casos de uso.

Estos se dividen en tres bloques principales:

La posibilidad de mantener activa la sesión para que, en próximas visitas con el mismo dispositivo, el usuario no tenga que rellenar el formulario de inicio de sesión.

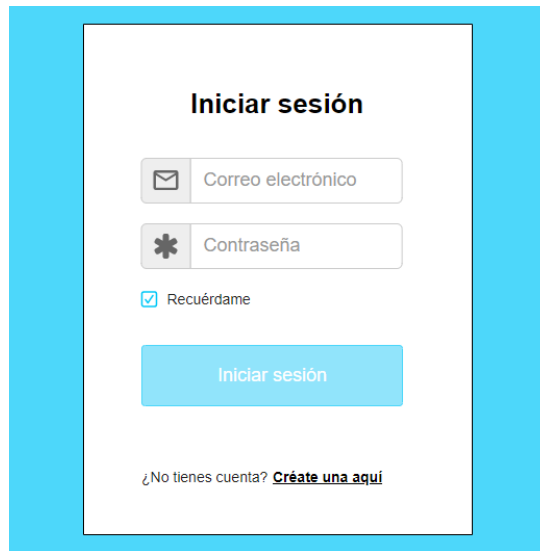
Poder navegar hasta la ventana de consulta de información de interés donde se aloja diversa información del centro, como dónde se sitúa, donde poder registrarse en el aplicativo, etc.

Para los fisioterapeutas exclusivamente, poder navegar hasta la ventana de informes, donde se aloja un cuadro de mando con diversos gráficos de distinta índole.

3.4.3. Mockups.

Para esta iteración se utilizaron como referencia los siguientes mockups:

EasyPhysio



El formulario de inicio de sesión está centrado en una tarjeta blanca con un borde azul. Encabezado: "Iniciar sesión". Campos de entrada: "Correo electrónico" (con ícono de correo) y "Contraseña" (con ícono de asterisco). Opción: "Recuérdame" (con casilla de verificación marcada). Botón: "Iniciar sesión" (en azul). Enlace: "¿No tienes cuenta? [Créate una aquí](#)".

Figura 17. Formulario para iniciar sesión..

El usuario podrá decidir si quiere que el dispositivo recuerde sus credenciales para mantener la sesión activa para próximos usos como se puede apreciar en la Figura 17.



Figura 18. Información de interés.

Siempre que el usuario lo estime oportuno, podrá acceder al panel informativo del centro en el que se mostrará diversa información, como la localización u horario del centro, información divulgativa respecto a la fisioterapia, modificaciones en los servicios... como se puede apreciar en la Figura 18.

3.4.4. Diseño arquitectónico.

En cuanto a módulos, los más relevantes durante esta iteración son los siguientes:

EasyPhysio

AboutUsComponent

Es el componente encargado de mostrar información de interés para el usuario, como dónde encontrar el centro, o información relevante sobre el centro y sus servicios, etc. Actualmente es un contenido estático con propósito estructural.

FooterComponent

Es el componente encargado de mostrar el pie de página corporativo a lo largo de la aplicación. Actualmente la mayoría del contenido no es funcional, puesto que no está contemplado en el prototipo dichas implementaciones.

ReportsComponent

Es el componente encargado de mostrar estadísticas en el cuadro de mando. Solo es visitable por parte de fisioterapeutas y para el prototipo solo están implementadas cuatro gráficas a modo de ejemplificación.

LoggedUserIsPhysioGuard

Guarda que se encarga de comprobar que el usuario activo es un fisioterapeuta para que este pueda acceder a pantallas de acceso restringido, de no serlo, se redirige al usuario al calendario de citas.

AppComponent

Se modificó el componente para mostrar la cabecera corporativa a través de todo el aplicativo.

Adicionalmente, la cabecera muestra una botonera en la parte superior derecha cuando un usuario tiene una sesión activa y le permite cerrar sesión, o bien desplegar un menú lateral con diferentes opciones según si es fisioterapeuta o paciente con el cual poder navegar. Algunas de estas opciones no son funcionales puesto que no está contemplado en el prototipo dichas implementaciones.

AppRoutingModule

Se modificó la gestión de rutas para que el usuario pueda navegar entre las rutas ya gestionadas por el componente previamente, o las adicionales en esta iteración, como es la pantalla de información de interés o la pantalla con el cuadro de mando para los fisioterapeutas.

Adicionalmente, se estipuló que la nueva página a mostrar cuando el usuario accede al dominio es la pantalla de información de interés, puesto que al ser el prototipo implementado un prototipo web y no una aplicación móvil, el usuario puede haber llegado al dominio por muchas y diversas maneras y hay que contextualizarle su experiencia, mientras que esto no sería necesario de hacer cuando una persona descarga una aplicación móvil, porque ha habido una intencionalidad de ejecutar dicha acción de descarga.

AppointmentService

EasyPhysio

Se modificó el servicio para poder solicitar al servidor de datos diferentes tipos de información respecto al recurso “citas”.

SchedulerComponent

Se modificó el componente con mejoras funcionales sobre la acción arrastrar y soltar una cita. Además, se implementó la capacidad de filtrar los días de vacaciones en el calendario de navegación.

En la iteración anterior se implementaron restricciones en la creación y edición de citas siendo estas: no poder crear dos citas en la misma franja horaria para el mismo paciente; no poder tener dos citas en la misma sala en la misma franja horaria; no poder reservar más citas para un fisioterapeuta cuyo cupo esté completo para una franja horaria; no poder reservar una cita para un fisioterapeuta que no trabaje ese día; y no poder crear una cita para un fisioterapeuta que trabaje ese día pero no a la hora seleccionada. En esta iteración, se ha implementado dichas casuísticas para la acción arrastrar y soltar y así tener ambas funcionalidades en sintonía.

Adicionalmente y sobre la acción de arrastrar y soltar, se implementó el bloqueo de arrastrar una cita a una franja horaria que corresponda a un día en el que el centro esté cerrado por vacaciones.

También se mejoró la acción de arrastrar y soltar para que un usuario no pueda arrastrar una cita a días anteriores al día actual.

ModalComponent

Se modificó el componente para añadirle unos estilos coherentes con las acciones permitidas a los usuarios, por ejemplo, si el usuario no puede crear una cita nueva porque está intentándolo hacer en un día festivo, los estilos del formulario ahora son acordes a dicho impedimento.

Además, se implementó el bloqueo de los días de vacaciones del centro a la hora de seleccionar un día para una nueva cita.

Cada uno de los módulos descritos anteriormente implementa sus correspondientes métodos. A continuación, los más representativos van a ser brevemente descritos:

getAppointmentsByPhysioAndRangeOfDates (reports.component.ts)

Se encarga de llamar a “appointmentService” para cada fisioterapeuta y con el multiplicador de semana introducido por la interacción del usuario.

Al recibir respuesta, la procesa para elaborar los datos necesarios para la gráfica correspondiente.

getAppointmentsByPhysioAndDay (reports.component.ts)

EasyPhysio

Se encarga de llamar a “appointmentService” para cada fisioterapeuta, por cada día de la semana y con el multiplicador de semana introducido por la interacción del usuario.

Al recibir respuesta, la procesa para elaborar los datos necesarios para la gráfica correspondiente.

getAppointmentsByPatientAndRangeOfDates (reports.component.ts)

Se encarga de llamar a “appointmentService” para cada paciente y con el multiplicador de semana introducido por la interacción del usuario.

Al recibir respuesta, la procesa para elaborar los datos necesarios para la gráfica correspondiente.

getAppointmentsByRoomAndRangeOfDates (reports.component.ts)

Se encarga de llamar a “appointmentService” para cada sala del centro y con el multiplicador de semana introducido por la interacción del usuario.

Al recibir respuesta, la procesa para elaborar los datos necesarios para la gráfica correspondiente.

auxRangeOfDates (appointmentService.ts)

Los métodos descritos anteriormente, llaman a su correspondiente función de “appointmentService” la cual establece conexión con el servidor de datos para que este les proporcione los datos necesarios para cada operación según los parámetros establecidos.

Este método auxiliar es el encargado de popular los rangos de fecha necesarios para dichas llamadas y que se sustenta de los parámetros de multiplicación de semanas enviados a cada método citado para poder establecer correctamente cuántas semanas han de solicitarse en las peticiones.

disableDropAppointmentsByConditions (scheduler.component.ts)

Es el método auxiliar encargado de comprobar si la cita arrastrada a una franja horaria es válida. Se comprueba que la cita no ha sido situada en un día festivo; ni en un día en el que el fisioterapeuta encargado de dicha cita no trabaje; ni en un día válido para el fisioterapeuta encargado, pero fuera de su horario laboral; ni que se superen las citas máximas permitidas para el fisioterapeuta en dicha franja horaria; ni que el paciente ya tenga previamente una cita en dicha franja horaria; ni que la sala de la cita ya esté previamente ocupada por otra cita existente en dicha franja horaria.

Adicionalmente, si se detecta algún error por el cual no se puede actualizar la cita al horario seleccionado, se muestra una notificación al usuario a través del componente “ToastContainer” con la información pertinente a la casuística.

FilteringNonWorkingDays (modal.component.ts)

EasyPhysio

Es el método encargado de bloquear días inválidos en el calendario de selección de día para una cita en el componente “modalComponent”. Los días festivos, los fines de semana y todos los días anteriores al día actual son aquellos que son bloqueados en el calendario de selección.

changeWorkingHoursAndWorkingDate (modal.component.ts)

Es el método encargado de popular las horas a mostrar en la lista de valores desplegable en el modal para crear nuevas citas o para editar las ya existentes y de filtrar en el calendario de selección los días que no trabaja el fisioterapeuta en cuestión.

Adicionalmente, según qué fisioterapeuta seleccione el usuario, aparte de mostrarse el horario pertinente en la lista de valores desplegable, si el día previamente seleccionado no es un día válido para dicho profesional, se fuerza al usuario a tener que seleccionar otro día del calendario.

3.4.5. Detalles de implementación.

Previamente, se describió que se implementó la capacidad de bloquear la creación de nuevas citas a lo largo de ciertos días del año, por ejemplo, debido a vacaciones. De manera análoga, sería reseñable poder permitir que cada fisioterapeuta pudiese elegir sus días de vacaciones y plasmarlas en el aplicativo; sin embargo, se decidió excluir del prototipo por priorización de otras funcionalidades más acuciantes.

Cabe destacar que debido a la alta cantidad de horas invertidas para el desarrollo y la baja relación de funcionalidades/tiempo empleado por la complejidad de haber generado un calendario y toda su gestión desde cero, se decidió no desarrollar *tests* y desviar la planificación horaria inicial para lograr tener un prototipo funcional con las características suficientes para poder ser una representación válida de los objetivos planteados por EasyPhysio.

4. Tecnologías

4.1. Angular.

Angular es un entorno de trabajo de código abierto desarrollado por Google para crear aplicaciones *Single-page* implementados en TypeScript.

El paradigma para generar aplicaciones en Angular es la creación de componentes. Estos están ideados para que cumplan un propósito funcional cada uno y están compuestos por: un fichero TypeScript que hace de controlador y que contiene variables y métodos con la lógica de negocio pertinente; un fichero HTML que muestra la vista al usuario y que a través de etiquetas personalizadas se puede crear acciones JavaScript e incluso llamar al controlador mencionado anteriormente para acceder o tratar datos del modelado; un fichero CSS (o SCSS o SASS según la configuración inicial seleccionada) para poder decorar adecuadamente el componente; y un fichero de pruebas spec.ts que a través de Karma y Jasmine se puede realizar cualquier tipo de *test suite*, desde pruebas unitarias hasta *end to end* pasando por pruebas integradas.

4.2. Angular CLI.

Angular CLI es la interfaz de línea de comandos de Angular, la cual incluye Webpack (<https://webpack.js.org/>) para poder generar un *bundle* de JavaScript para poder ejecutarlo en el navegador de manera cómoda.

Adicionalmente, Angular CLI asiste al programador con la creación ágil de proyectos y código, a la hora de crear componentes, test, directivas... que son generados cumpliendo las buenas prácticas de <https://angular.io>.

4.3. AngularFireModule y AngularFirestoreModule.

AngularFire aligera la complejidad que un desarrollador de Angular puede encontrar al implementar el SDK de Firebase JS, que es independiente del entorno de trabajo, y tiene como objetivo proporcionar una experiencia al desarrollador más natural, al cumplir con las convenciones de Angular.

Esta librería se encarga de la inyección de dependencias, de los envoltorios para Zone.js, del *lazy-loading*, de desplegar esquemas, de Google Analytics, de aportar guardias prehechas para la autenticación, de usar NgRX y, además, está basado en *Observable*.

4.4. Angular Material.

Angular Material es una librería de estilos (como Bootstrap) basada en la guía de diseño de Material Design, realizado por el equipo de Angular para integrarse perfectamente con Angular.

EasyPhysio

4.4.1. DragDropModule.

El módulo *drag & drop* proporciona una manera fácil de crear declarativamente interfaces de arrastrar y soltar, con soporte para arrastrar libremente, ordenar dentro de una lista, transferir elementos entre listas, animaciones, soporte a dispositivos táctiles, manejadores de arrastre personalizados, vistas previas y marcadores de posición, además de listas horizontales y bloqueo a lo largo de un eje.

4.4.2. MatAutocompleteModule

El autocomplete es una entrada de texto normal mejorada con un panel de opciones sugeridas que pueden tener una lógica adicional para mostrar una parte del modelo y que, al seleccionar un valor, en el controlador se guarde otra parte del modelo diferente de dicho elemento.

4.4.3. MatCheckboxModule

Proporciona la misma funcionalidad que un *checkbox* nativo de HTML pero está mejorado con estilos y animaciones de Material Design.

4.4.4. MatDatepickerModule.

El selector de fecha permite a los usuarios ingresar una fecha a través de la entrada de texto o eligiendo una fecha del calendario. Se compone de varios componentes, directivas y el módulo de implementación de fecha que trabajan juntos.

4.4.5. MatFormField.

Es un componente que se utiliza para envolver varios componentes de Angular Material, normalmente elementos de un formulario, y aplicar estilos de campo de texto comunes, como el subrayado, la etiqueta flotante y los mensajes de sugerencia.

4.4.6. MatInputModule.

`matInput` es una directiva que permite que los elementos nativos *input* y *textarea* funcionen con el componente *MatFormField* previamente citado.

4.4.7. MatRadioModule

Proporciona la misma funcionalidad que un *radio button* nativo, pero está mejorado con estilos y animaciones de Material Design.

4.4.8. MatSelectModule

Es un control de formulario para seleccionar un valor de un conjunto de opciones, similar al elemento nativo *select*.

EasyPhysio

Para agregar opciones a la selección, solo hace falta crear elementos *mat-option* dentro del *mat-select*. Cada *mat-option* tiene una propiedad de valor que se puede utilizar para establecer el valor que se seleccionará si el usuario elige esta opción. El contenido de *mat-option* es lo que se le mostrará al usuario.

4.4.9. MatSidenavModule.

Angular Material proporciona dos conjuntos de componentes diseñados para agregar contenido lateral plegable (a menudo navegación, aunque puede ser cualquier contenido) junto al contenido principal. Estos son los componentes de *sidenav* y *drawer*.

Para este propósito, se ha utilizado el *sidenav* puesto que están diseñados para agregar contenido lateral a una aplicación de pantalla completa. Para configurar un *sidenav* se usa tres componentes: `<mat-sidenav-container>` que actúa como un contenedor estructural para el contenido y *sidenav*, `<mat-sidenav-content>` que representa el contenido principal, y `<mat-sidenav>` que representa el contenido lateral añadido.

4.4.10. MatTooltipModule.

El *tooltip* de Angular Material proporciona una etiqueta de texto que se muestra cuando el usuario pasa el cursor sobre un elemento o lo mantiene presionado.

4.5. CSS3.

Es un lenguaje de diseño creado para mejorar la presentación de un documento implementado en un lenguaje de marcas, siendo los predominantes HTML y XHTML.

4.6. ESLint.

ESLint es un analizador sintáctico de código para JavaScript altamente personalizable, pudiendo añadir el usuario cualquier regla que considere necesario para amoldar la librería a su proyecto. Se puede encontrar en la mayoría de los editores de texto actuales, pudiendo ejecutar análisis cómodamente.

4.7. Firebase.

Firebase es una plataforma en la nube para desarrollar aplicaciones web y móvil. Es un proyecto realizado por Google y su función principal es facilitar al programador la creación de aplicaciones, pero sin renunciar calidad. Además, al ser un producto de Google es fácilmente vinculable con otros aspectos de la compañía, como el análisis de datos o la monetización.

4.8. FontAwesomeModule.

Componente oficial para Angular de FontAwesome, que es una corporación centrada en el diseño de iconos y logos que se pueden localizar en <https://fontawesome.com>.

4.9. Git.

Git es un *software* de control de versiones mundialmente utilizado por su eficiencia, mantenimiento fácil de aplicaciones con un elevado número de archivos de código, la agilización de las tareas de coordinación entre compañeros de trabajo y otras muchas cosas.

4.10. HTML5.

Es la quinta revisión de relevancia de HTML, el lenguaje de marcas regulado por el consorcio W3C y que es el lenguaje básico de internet. Esta nueva versión publicada en 2014 no es compatible con antiguas versiones de navegadores por sus nuevas etiquetas.

4.11. jQuery.

jQuery es una biblioteca de JavaScript que agiliza y facilita la manipulación del DOM, del manejo de eventos, de las peticiones AJAX y mucho más, aportando funciones propias con una alta eficiencia en complejidad y tiempo.

Es software libre y de código abierto, y gracias a sus licencias MIT y GNU v2 se puede utilizar para proyectos libres y privados.

4.12. Lodash.

Lodash es una librería moderna de JavaScript, sucesora de Underscore.js, que aporta legibilidad al código, rendimiento y variados tipos de utilidad al programador.

4.13. MomentJS.

MomentJS es una librería de JavaScript para aportar facilidad en el uso y tratamiento de fechas. Hace fácilmente manipulables las fechas para poder restar o sumar días, e incluso, poder controlar cómodamente los husos horarios del usuario, como también mostrar fechas en diversos formatos según el idioma/cultura pertinente.

4.14. NgxChartsModule.

Ngx-charts es un envoltorio para D3 (<https://d3js.org>) creado para Angular con el que poder generar gráficas procesando elemento VSG, además de utilizar unas estéticas visualmente atractivas y unos estilos CSS personalizables. Además, permite la auto creación de gráficas enteramente personalizables desde cero.

4.15. Ng-bootstrap.

Ng-bootstrap es una librería que integra nativamente Bootstrap (<https://getbootstrap.com>) con Angular, sin la necesidad de utilizar las librerías JQuery, Popper y la librería Javascript propia de Bootstrap original.

4.16. NodeJS.

NodeJS es un entorno en tiempo de ejecución multiplataforma, de código abierto, orientado al lado del servidor, aunque con capacidad también de hacerlo del lado del cliente. Está basado en JavaScript y en el motor V8 de Google. Además, es asíncrono y su entrada/salida de datos está orientada a eventos.

4.17. PopperJS.

Popper es una librería de JavaScript para poder utilizar *tooltips* y *popovers* en HTML. Es ligera y ofrece una gran abstracción a diferencia de otras librerías con propósitos similares. Además, es modular y con altas capacidades de personalizar el contenido.

4.18. RXJS.

RxJS es una biblioteca para programación reactiva que utiliza *Observables*, para facilitar la composición de código asíncrono o basado en devolución de llamadas. Este proyecto es una reescritura de Reactive-Extensions / RxJS con mejor rendimiento, mejor modularidad, mejores pilas de llamadas depurables, mientras se mantiene en su mayoría compatible con versiones anteriores, con algunos cambios importantes que reducen la API.

4.19. SCSS.

SCSS es un metalenguaje de CSS que da la capacidad de generar mixins o anidar estilos, sin embargo, al ser transpilado genera CSS compatible con los estándares originales.

4.20. TypeScript.

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft. Es un superconjunto de JavaScript y sus principales ventajas es que ofrece tipado de datos estáticos y objetos basados en clase. Como es lógico, cualquier código de JavaScript es válido para TypeScript puesto que el transpilador convierte todo el código a JavaScript.

Se puede utilizar tanto para aplicaciones en el lado del servidor como en el lado del cliente.

5. Acceso al código y despliegue.

Se deja enlace al repositorio abierto a continuación:

<https://github.com/KevinReal/EasyPhysio>

Para poder realizar pruebas con el prototipo, hace falta tener algún entorno de desarrollo para lanzar el contenedor. Una vez en ejecución, en el puerto local 4200, introducir en el formulario de inicio de sesión las siguientes credenciales:

Fisioterapeuta

Correo electrónico: kevinrealcalde@gmail.com

DNI/NIE: 12345678K

Contraseña: 12345678aA

Paciente (se puede crear un nuevo usuario en el formulario de registro si así se desea)

Correo electrónico: aldacruzcuadrado@hotmail.es

Contraseña: 12345678aA

6. Conclusiones.

A lo largo del proyecto EasyPhysio se puede comprobar que el prototipo valida perfectamente los objetivos a conseguir en la medida de lo abarcable para un prototipo. Asimismo, al no ser un producto final, elaborado en las horas y con los medios necesarios para la ambición planteada, se puede apreciar que lo desarrollado se ajusta a las capacidades disponibles.

Bien es cierto que, ante el proyecto aquí presentado, cabe la posibilidad de encontrarse con limitaciones reales, puesto que al final y al cabo, todo el proceso expuesto tiene como base contextos con un componente tecnológico perteneciente a los ámbitos de redacción de pliegos, administración de empresas y/o proyecciones a medio o largo plazo, los cuales llevan aparejados diversos aspectos transversales como la gestión de un equipo de desarrollo o el ajuste de una planificación con base en hitos supeditados por la temporalidad y el plan presupuestario. Sin embargo, no es de menospreciar el trasfondo técnico y concienzudo presente en todos los aspectos tangibles de EasyPhysio: la consecución de un prototipo estable y funcional en plazo y forma estipulados; la capacidad analítica de afrontar problemas fácilmente localizables en el mercado laboral para su resolución; y la capacidad de ajustarse a unas necesidades protocolarias establecidas por un ente corporativo o institucional.

Una vez contextualizados las limitaciones y el potencial, es posible dilucidar que EasyPhysio es un proyecto que tiene unas altas cuotas de mejora, pero que establece unos mínimos creíbles y verosímiles en sus pretensiones aquí plasmadas.

Tecnológicamente, este es un proyecto con posibilidad real de inserción en el mercado, puesto que se usan tecnologías punteras y competentes con solvencia demostrada, como puede ser Angular, que adicionalmente cuenta con el respaldo de nada menos que de una gran entidad como Google.

En cuanto al nicho de mercado establecido, y en relación con lo ya comentado anteriormente, este se presenta con una capacidad de explotación elevada, con soluciones que satisfacen las demandas existentes a falta de mejores aportaciones, pero que, con una correcta consecución de los objetivos establecidos en este proyecto, se podría generar un importante hueco entre la competencia.

Por tanto, es seguro estimar que EasyPhysio, con el correcto desarrollo de todos sus ámbitos, es un proyecto con un valor real del que se puede generar un futuro prometedor.

Finalmente, y desde un punto de vista más personal, la realización de EasyPhysio esclarece ligeramente cuán difícil debe ser generar un proyecto profesional que produzca un valor real en un mundo en el que “está todo inventado” como bien sugiere el dicho popular. La similitud de la realización de este proyecto en comparativa con algún pliego generado por personal adecuado y formado profesionalmente para conseguir unas metas ambiciosas hace reflexionar cuanto menos, lo dificultoso y epopéyico que dicha tarea debe ser.

EasyPhysio

Sin embargo, no ha de estar subyugado por la incertidumbre o lo hipotético de un ámbito tan cambiante y de obligada actualización, puesto que, al ejecutar un análisis retrospectivo y su posterior reflexión, estos hacen valorar debidamente los objetivos conseguidos no solo durante la realización de EasyPhysio, sino a lo largo de la travesía de los estudios.

7. Fuentes de información.

Todas las fuentes de información listadas a continuación fueron validadas por última vez el 22/06/2021.

Adobe Color. <https://color.adobe.com/es/create/color-wheel>

Advisera. <https://advisera.com/eugdpracademy/es/knowledgebase/un-resumen-de-10-requisitos-clave-del-rgpd>

Angular. <https://angular.io>

Angular Material. <https://material.angular.io>

ArchivexClinical. <https://archivexclinical.com>

Booksy. <https://booksy.com/es-es>

ClinicCloud. <https://clinic-cloud.com>

CodeGrepper. <https://www.codegrepper.com>

CodeInGame. <https://www.codingame.com>

DevelopersGoogle. <https://developers.google.com>

DevelopersMozilla. <https://developer.mozilla.org/es>

Dropbox. <https://www.dropbox.com>

EII ULPGC. https://www.eii.ulpgc.es/tb_university_ex/?q=objetivos-y-competencias-del-gii

ESLint. <https://eslint.org>

Firebase. <https://firebase.google.com>

Fireship. <https://fireship.io>

FontAwesome. <https://fontawesome.com>

FreeCodeCamp. <https://www.freecodecamp.org>

Fresha. <https://es.fresha.com>

Github. <https://github.com>

Juristr. <https://juristr.com>

jQuery. <https://jquery.com>

Lodash. <https://lodash.com/docs/4.17.15>

NetClinicas. <https://www.netclinicas.com>

NGBootstrap. <https://ng-bootstrap.github.io/#/home>

EasyPhysio

NodeJS. <https://nodejs.org/es>

Medium. <https://medium.com>

MomentJS. <https://momentjs.com>

OMTSpain. <https://omtspain.es/historia-de-la-fisioterapia-en-espana>

PopperJS. <https://popper.js.org>

Regexr. <https://regexr.com>

RXJS. <https://rxjs.dev>

Schedulicity. <https://www.schedulicity.com>

StackOverflow. <https://stackoverflow.com>

Swimlane. <https://swimlane.gitbook.io/ngx-charts>

SASSLang. <https://sass-lang.com>

Tektutorialshub. <https://www.tektutorialshub.com>

TypeScriptLang. <https://www.typescriptlang.org>

W3C. <https://www.w3c.es>

Wikipedia. <https://es.wikipedia.org>

Youtube. <https://www.youtube.com>

8. Anexo.

i. Manual de usuario

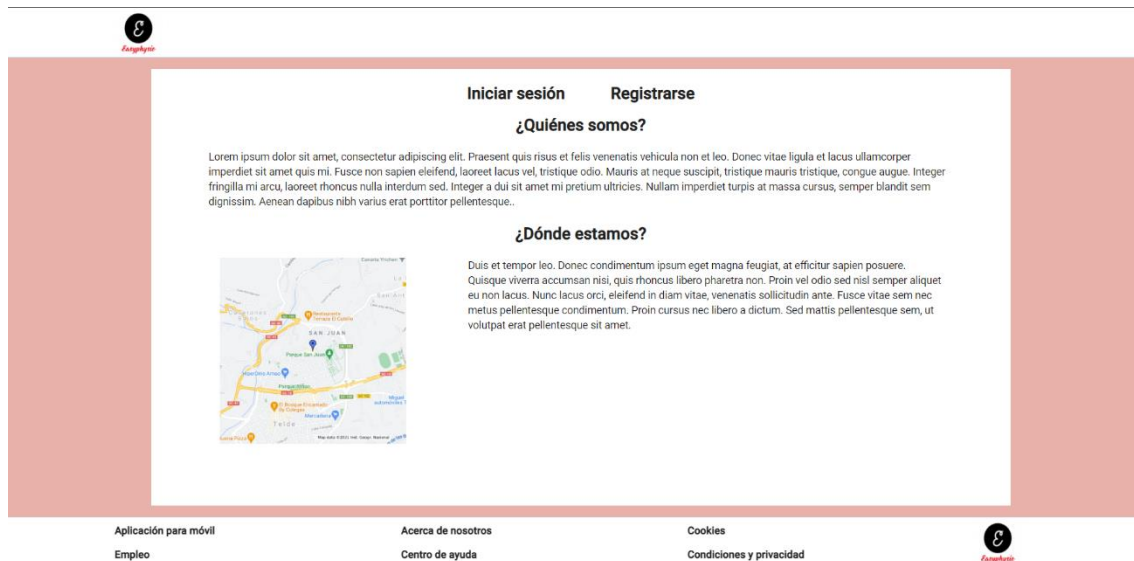


Figura 1. Información de interés.

El usuario al entrar en el aplicativo de escritorio aterriza en una página en la que se muestra información diversa sobre el centro, como dónde se sitúa o cómo contactar con la empresa, además de enlaces a las pantallas para iniciar sesión o para registrarse en el sistema como se puede apreciar en la Figura 1.

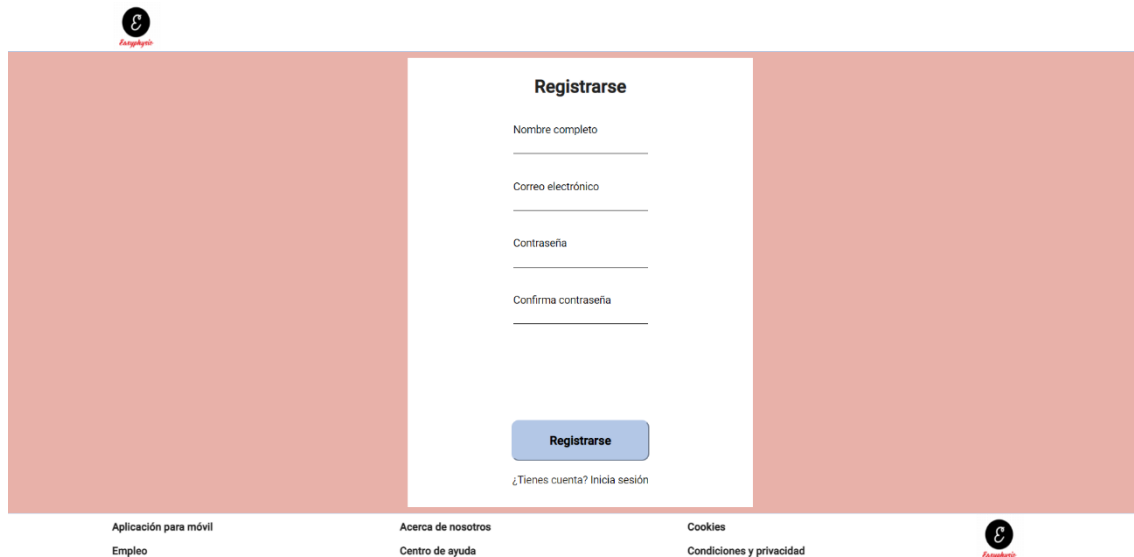


Figura 2. Formulario de registro.

En el formulario de registro, el nuevo usuario rellena los campos correspondientes como se puede ver en la Figura 2 y es dado de alta en el sistema para futuros ingresos con sus recién creadas credenciales.

The screenshot shows the registration page with the following elements and errors:

- Form Title:** Registrarse
- Nombre completo:** Input field with a red underline.
- Correo electrónico:** Input field with a red underline and an error message: "El formato del correo electrónico no es válido."
- Contraseña:** Input field with a red underline and an error message: "La contraseña debe tener mínimo 8 caracteres con al menos 1 letra minúscula, 1 letra mayúscula y 1 dígito."
- Confirma contraseña:** Input field with a red underline and an error message: "Las contraseñas no coinciden."
- Buttons:** A blue "Registrarse" button and a link "¿Tienes cuenta? Inicia sesión".
- Footer:** Navigation links for "Aplicación para móvil", "Empleo", "Acerca de nosotros", "Centro de ayuda", "Cookies", and "Condiciones y privacidad".

Figura 3. Errores de validación del formulario de registro.

Si se intenta crear un nuevo usuario, pero los datos introducidos en el formulario no son válidos en su formato, se le indicará al usuario los errores pertinentes para que pueda proceder a su corrección como se puede apreciar en la Figura 3.

The screenshot shows the login page with the following elements:

- Form Title:** Iniciar sesión
- Correo electrónico:** Input field.
- Contraseña:** Input field.
- Options:** Two checkboxes: "Recuérdame" and "Soy fisioterapeuta".
- Buttons:** A blue "Iniciar sesión" button and a link "¿No tienes cuenta? Créate una".
- Footer:** Navigation links for "Aplicación para móvil", "Empleo", "Acerca de nosotros", "Centro de ayuda", "Cookies", and "Condiciones y privacidad".

Figura 4. Formulario de inicio de sesión para pacientes.

Por su parte, en el formulario de inicio de sesión, el usuario una vez registrado será capaz de introducir sus credenciales para volver a ingresar en el aplicativo como se puede apreciar en la Figura 4. Adicionalmente el usuario tendrá la posibilidad de marcar la casilla "Recuérdame" para mantener activa la sesión en dicho dispositivo.

The screenshot shows the 'Iniciar sesión' (Login) form for physiotherapists. The form is centered on a white background with a light red border. It includes the following elements:

- Logo: EasyPhysio logo in the top left corner.
- Form Title: 'Iniciar sesión'.
- Fields: 'Correo electrónico' (Email), 'NIF/NIE', and 'Contraseña' (Password).
- Checkboxes: 'Recuérdame' (Remember me) and 'Soy fisioterapeuta' (I am a physiotherapist).
- Button: 'Iniciar sesión' (Login).
- Link: '¿No tienes cuenta? Créate una' (Don't have an account? Create one).
- Footer: Navigation links for 'Aplicación para móvil', 'Empleo', 'Acerca de nosotros', 'Centro de ayuda', 'Cookies', and 'Condiciones y privacidad', along with the EasyPhysio logo.

Figura 5. Formulario de inicio de sesión para fisioterapeutas.

Alternativamente, para el inicio de sesión cabe destacar que el fisioterapeuta podrá hacerlo a través del mismo formulario ampliado como se puede apreciar en la Figura 5; sin embargo, la entrada de nuevos usuarios al sistema no será posible a través del registro habitual, solo será realizado a través de un administrador.

The screenshot shows the 'Iniciar sesión' (Login) form for physiotherapists with validation errors. The form is centered on a white background with a light red border. It includes the following elements:

- Logo: EasyPhysio logo in the top left corner.
- Form Title: 'Iniciar sesión'.
- Fields: 'Correo electrónico' (Email), 'NIF/NIE', and 'Contraseña' (Password).
- Validation Messages: 'El formato del correo electrónico no es válido.' (Email format is invalid), 'El formato del NIF/NIE no es válido.' (NIF/NIE format is invalid), and 'La contraseña debe tener mínimo 9 caracteres con al menos 1 letra minúscula, 1 letra mayúscula y 1 dígito.' (Password must be at least 9 characters with at least 1 lowercase letter, 1 uppercase letter, and 1 digit).
- Checkboxes: 'Recuérdame' (Remember me) and 'Soy fisioterapeuta' (I am a physiotherapist).
- Button: 'Iniciar sesión' (Login).
- Link: '¿No tienes cuenta? Créate una' (Don't have an account? Create one).
- Footer: Navigation links for 'Aplicación para móvil', 'Empleo', 'Acerca de nosotros', 'Centro de ayuda', 'Cookies', and 'Condiciones y privacidad', along with the EasyPhysio logo.

Figura 6. Errores de validación del formulario de inicio de sesión.

Tanto para el formulario de inicio de sesión para pacientes como para el de fisioterapeutas, se intenta introducir datos con un formato incorrecto, se le indicará al

usuario los errores pertinentes para que pueda proceder a su corrección como se puede ver en la Figura 6.



Figura 7. Error en el inicio de sesión.

Tanto si el usuario ha intentado registrarse, como si ha intentado iniciar sesión, existe la posibilidad de producirse diversos errores en el proceso, por ejemplo, que introduzca incorrectamente el NIF/NIE al intentar acceder como fisioterapeuta como se puede apreciar en la Figura 7. Adicionalmente existen otros controles de errores como haber introducido una cuenta o contraseña incorrectas; intentar crear un nuevo usuario con un correo electrónico ya registrado; intentar entrar como fisioterapeuta con una cuenta que no tiene permiso para ello; etc.



Figura 8. Agenda mostrada a los pacientes.

Una vez iniciada la sesión, el paciente llega a la pantalla en la que se muestra su agenda, la cual inicialmente mostrará únicamente sus citas reservadas como se puede apreciar en la Figura 8.

Sin embargo, tendrá la capacidad de modificar lo que se muestra en el calendario a través de los filtros del menú, con los que podrá llegar a ver todas las citas del centro si lo estima oportuno, para poder así reservar cómodamente en la franja horaria que desee.

A lo largo de toda la aplicación se podrá apreciar que al contenido principal mostrado en la pantalla le acompañan un pie de página y una cabecera la cual cambiará según si hay o no un usuario con una sesión activa.

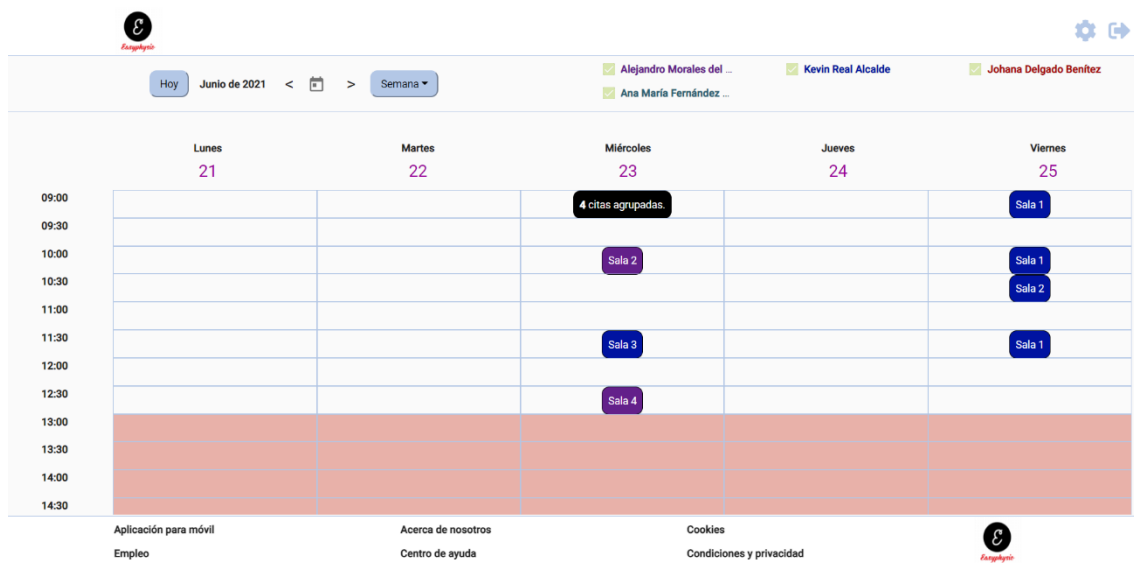


Figura 9. Agenda mostrada a los fisioterapeutas.

El fisioterapeuta tendrá una vista similar tras iniciar sesión, diferenciándose en que inicialmente se mostrarán todas las citas del centro pudiendo filtrar aquellas que le resulte conveniente como se puede apreciar en la Figura 9.

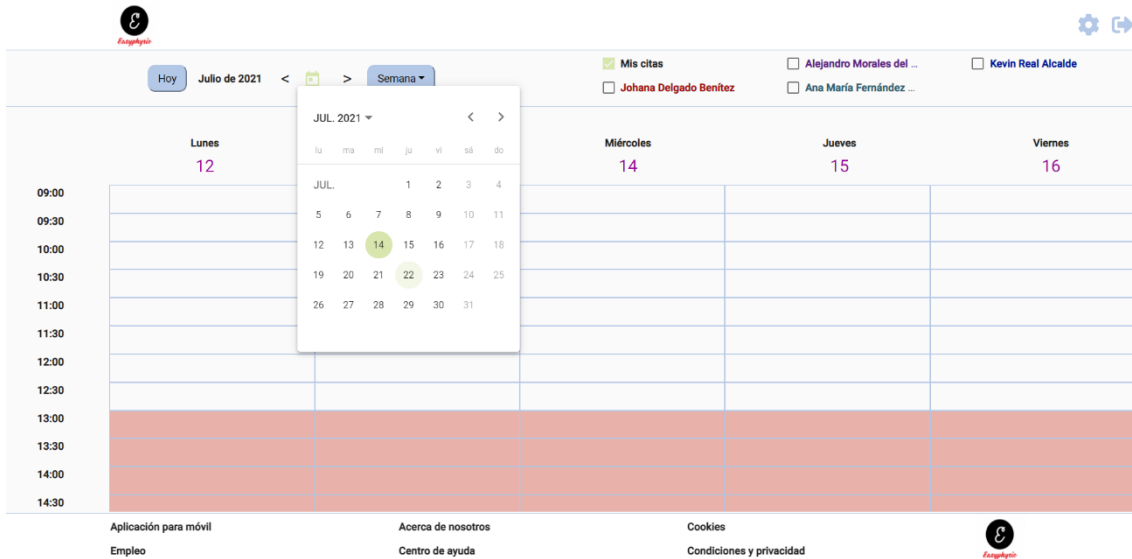


Figura 10. Interacción con el calendario.

Tanto los pacientes como los fisioterapeutas serán capaces de trasladarse en el tiempo a la semana que gusten para poder visualizar las citas de dicha semana, o bien para poder reservar una cita para algún día. Tendrán diferentes posibilidades para ello, pudiendo navegar a la semana anterior o a la siguiente a la actual o bien a través de la interacción con el calendario como se puede apreciar en la Figura 10.

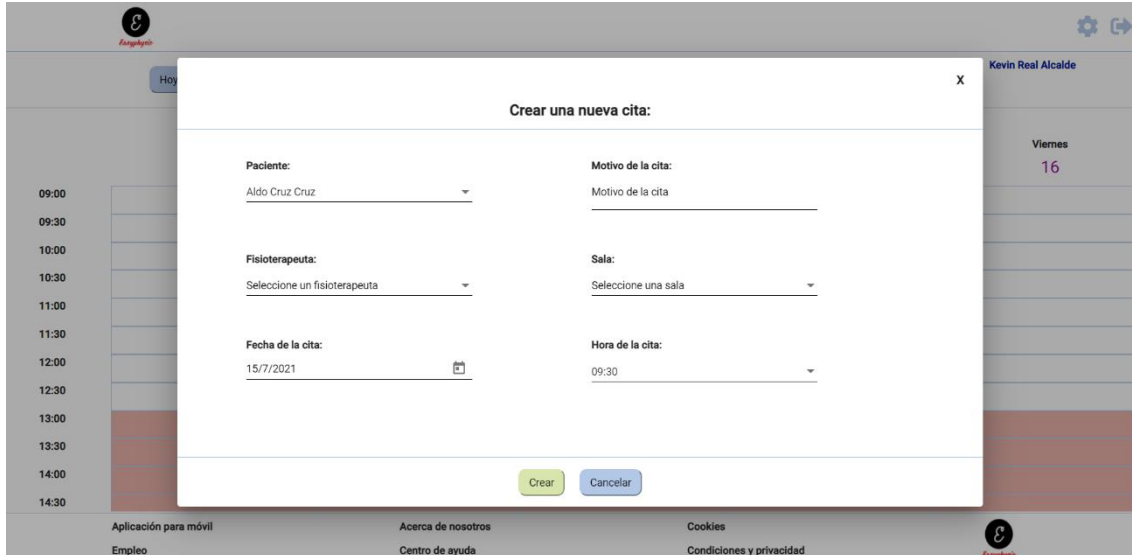


Figura 11. Formulario para la creación de una nueva cita.

Una vez elegido el día y hora para reservar una nueva cita, bien sea el usuario activo un paciente o un fisioterapeuta, habrá que cumplimentar el formulario pertinente para ello como se puede apreciar en la Figura 11.

Crear una nueva cita:

Paciente:
Seleccione un paciente

Motivo de la cita:
Motivo de la cita

Fisioterapeuta:
Johana Delgado Benítez

Sala:
Seleccione una sala

Fecha de la cita:
Seleccione una fecha

Hora de la cita:
Seleccione una hora

Crear Cancelar

Figura 12. Errores de validación del formulario para la creación de una nueva cita.

Si el usuario intenta introducir unos datos inválidos en el formulario de creación de una nueva cita, se mostrarán errores para que este pueda proceder a su corrección como se puede apreciar en la Figura 12.

De manera análoga, si el usuario está intentando modificar una cita existente e introduce unos datos inválidos en dicho formulario, se le mostrarán errores para que pueda proceder a su corrección.

Detalles de la cita:

Paciente:
Aldo Cruz Cruz

Motivo de la cita:
asd

Fisioterapeuta:
Alejandro Morales del Cristo

Sala:
2 | armonía

Fecha de la cita:
23/6/2021

Hora de la cita:
10:00

Editar Borrar Cancelar

Figura 13. Formulario para la edición o eliminación de una cita ya existente.

Una vez creada una nueva cita, se podrá modificar los datos de esta, o incluso se podrá borrar si así es necesario como se puede apreciar en la Figura 13.

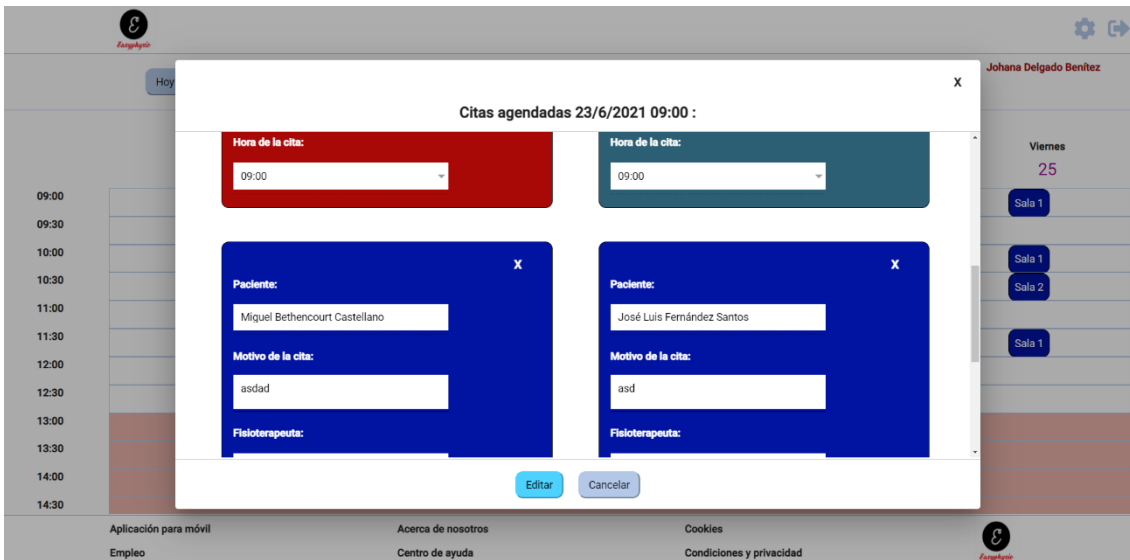


Figura 14. Lista de citas agrupadas.

Si una franja horaria acumulase muchas citas, se mostraría un listado de las citas pertinentes como se puede apreciar en la Figura 14.

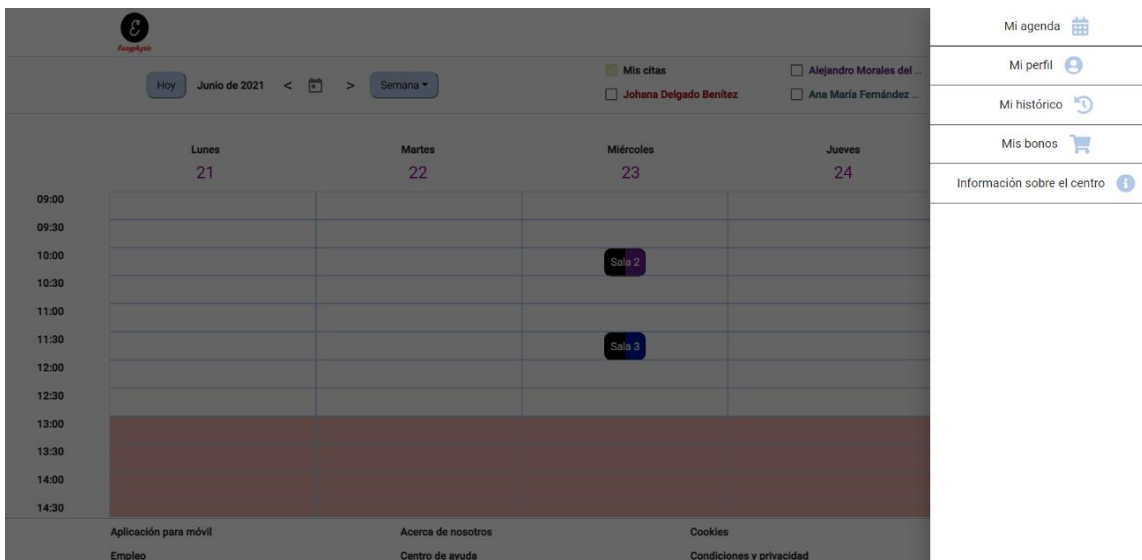


Figura 15. Opciones del menú lateral.

Todos los usuarios tendrán la capacidad de navegar a través del aplicativo a través del menú lateral como se puede apreciar en la Figura 15.

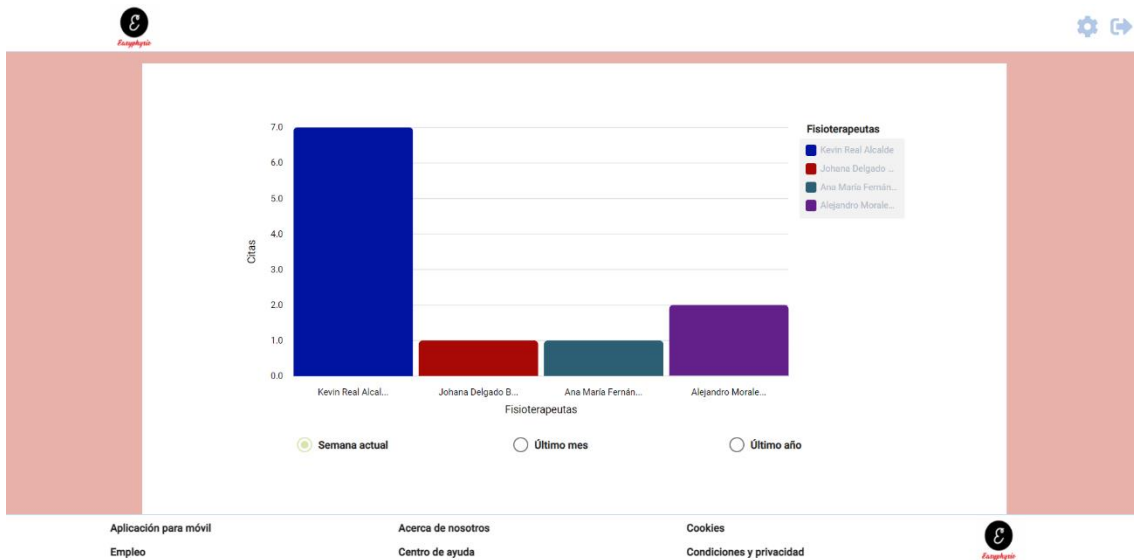


Figura 16. Estadísticas mostradas en el cuadro de mando.

Exclusivamente los fisioterapeutas serán capaces de entrar en el cuadro de mando, en el que se mostrarán diferentes gráficos, mostrando variadas estadísticas como se puede apreciar en la Figura 16.



Figura 17. Error al acceder a ruta sin permiso.

Adicionalmente, existe un control de rutas para que el usuario no acceda a vistas que no debe poder ver cómo se puede apreciar en la Figura 17. Por ejemplo, si el usuario activo es un paciente, no debe poder entrar a ver los informes, aunque en el navegador escriba la ruta exacta correspondiente.

Otros ejemplos de control de ruta se aprecian cuando un usuario sin sesión activa intenta acceder al calendario de citas a través del navegador; o cuando un usuario con una sesión activa intenta acceder al formulario de inicio de sesión o al de registro de un nuevo usuario.