



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandía

PoCAR: puntos de atención sanitaria en entornos rurales y remotos

Trabajo Fin de Grado - Anexos

Grado en Tecnologías Interactivas

AUTORA: Laura Melero Garrigós

Tutora: Pérez Pascual, M^a Asunción

Cotutor: Alberola Oltra, Juan Miguel

CURSO ACADÉMICO: 2021-2022

Contenido

1	Diseños aplicaciones	8
1.1	Diseño.....	8
1.1.1	Moodboard.....	8
1.1.2	Paleta de colores	8
1.1.3	Fuentes	9
1.2	Aplicación web médicos.....	9
1.2.1	Wireframes.....	9
1.2.2	Mockups	14
1.2.3	Resultado final	19
1.3	PWA pacientes	36
1.3.1	Wireframes.....	36
1.3.2	Mockups	37
1.3.3	Resultado final	37
2	Test Selenium.....	42
2.1	Login	42
2.2	Eventos.....	42
2.3	Pacientes	43
2.4	Médicos.....	45
2.4.1	Médico actual	45
2.4.2	Todos los médicos	46
2.5	Zonas de salud básicas (ZBS)	47
2.6	Centros	48
2.7	Salas.....	48
2.8	Dispositivos	49
2.9	Conclusiones	50
3	Instalaciones	52
3.1	Generales	52
3.1.1	Control de versiones Sourcetree	52
3.1.2	Visual Studio Code.....	52
3.1.3	Python	53
3.2	Raspberry Pi	53
3.2.1	Bluepy.....	53
3.2.2	PM2	53
3.3	Windows	54
3.3.1	Xampp.....	54
3.3.2	Node JS	54

3.3.3	Angular	55
3.3.4	Selenium	55
3.3.5	Axure	55
3.4	Librerías Angular	56
3.4.1	PrimeNG y PrimeFlex.....	56
3.4.2	Angular 12.0+ calendar.....	56
3.4.3	AWS IoT Device SDK V2	56
3.4.4	VideoSDK	57
4	Manuales de instrucciones	59
5	Referencias bibliográficas	60

Figuras

Figura 1. Moodboard	8
Figura 2. Paleta colores.....	9
Figura 3. Fuentes.....	9
Figura 4. Wireframe inicio sesión médicos	10
Figura 5. Wireframe inicio médicos	10
Figura 6. Wireframe pacientes médicos	11
Figura 7. Wireframe consulta médicos	11
Figura 8. Wireframe calendario médicos (v1).....	12
Figura 9. Wireframe calendario médicos (v2).....	13
Figura 10. Wireframe ajustes médicos	13
Figura 11. Wireframe modales médicos	14
Figura 12. Mockup inicio sesión médicos	15
Figura 13. Mockup inicio médicos	15
Figura 14. Mockup pacientes médico	16
Figura 15. Mockup consulta médicos	16
Figura 16. Mockup calendario médicos (v1)	17
Figura 17. Mockup calendario médicos (v2)	17
Figura 18. Mockup ajustes médicos.....	18
Figura 19. Mockup modales médicos	18
Figura 20. Resultado final inicio sesión médicos	19
Figura 21. Resultado final inicio médicos (consulta por realizar)	19
Figura 22. Resultado final inicio médicos (añadir evento).....	20
Figura 23. Resultado final pacientes médicos (paciente no seleccionado)	21
Figura 24. Resultado final pacientes médicos (crear paciente)	21
Figura 25. Resultado final pacientes médicos (editar paciente).....	22
Figura 26. Resultado final pacientes médicos (paciente seleccionado)	22
Figura 27. Resultado final pacientes médicos (añadir evento).....	23
Figura 28. Resultado final pacientes médicos (editar evento)	23
Figura 29. Resultado final pacientes médicos (filtro tiempo históricos)	24
Figura 30. Resultado final pacientes médicos (filtro tipos gráficos)	24
Figura 31. Resultado final consulta médicos (consulta general)	25
Figura 32. Resultado final inicio médicos (consulta finalizada)	25
Figura 33. Resultado final consulta médicos (toma medición correcta)	26
Figura 34. Resultado final consulta médicos (toma medición correcta 2)	26
Figura 35. Resultado final consulta médicos (toma medición errónea)	27
Figura 36. Resultado final consulta médicos (reintento toma medición).....	27
Figura 37. Resultado final consulta médicos (videollamada)	28
Figura 38. Resultado final ajustes médicos.....	28
Figura 39. Resultado final ajustes médicos (médico sin permisos de administrador).....	29
Figura 40. Resultado final ajustes médicos (editar médico).....	29
Figura 41. Resultado final ajustes médicos (crear médico)	30
Figura 42. Resultado final ajustes médicos (editar ZBS)	30
Figura 43. Resultado final ajustes médicos (crear ZBS)	31
Figura 44. Resultado final ajustes médicos (editar centro)	31
Figura 45. Resultado final ajustes médicos (crear centro).....	32
Figura 46. Resultado final ajustes médicos (editar sala).....	32
Figura 47. Resultado final ajustes médicos (crear sala)	33
Figura 48. Resultado final ajustes médicos (editar dispositivo).....	33

Figura 49. Resultado final ajustes médicos (crear dispositivo)	34
Figura 50. Resultado final móvil inicio sesión médicos.....	34
Figura 51. Resultado final móvil inicio médicos.....	34
Figura 52. Resultado final móvil inicio médicos (crear paciente)	34
Figura 53. Resultado final móvil calendario médicos	34
Figura 54. Resultado final móvil calendario médicos (editar evento)	35
Figura 55. Resultado final móvil pacientes médicos.....	35
Figura 56. Resultado final móvil ajustes médicos	35
Figura 57. Resultado final móvil ajustes médicos (crear médico)	35
Figura 58. Wireframe inicio sesión pacientes	36
Figura 59. Wireframe consulta pacientes	36
Figura 60. Mockup inicio sesión pacientes	37
Figura 61. Mockup consulta pacientes	37
Figura 62. Resultado final inicio sesión pacientes	38
Figura 63. Resultado final inicio sesión pacientes (lista espera).....	38
Figura 64. Resultado final consulta pacientes	39
Figura 65. Resultado final consulta pacientes (toma medición correcta)	39
Figura 66. Resultado final consulta pacientes (toma medición correcta 2)	40
Figura 67. Resultado final consulta pacientes (toma medición errónea)	40
Figura 68. Resultado final consulta pacientes (reintento toma medición).....	41
Figura 69. Resultado final consulta pacientes (videollamada)	41
Figura 70. Resultado testeo Login Selenium Python.	42
Figura 71. Resultado testeo Eventos Selenium Python	42
Figura 72. Resultado testeo Eventos Selenium Web 1	43
Figura 73. Resultado testeo Eventos Selenium Web 2	43
Figura 74. Resultado testeo Eventos Selenium Web 3	43
Figura 75. Error Pacientes detectado con Selenium	44
Figura 76. Error Pacientes solucionado.....	44
Figura 77. Resultado testeo Pacientes Selenium Python.....	44
Figura 78. Previo testeo Pacientes Selenium Web	44
Figura 79. Resultado testeo Pacientes Selenium Web 1	44
Figura 80. Resultado testeo Pacientes Selenium Web 2	45
Figura 81. Resultado testeo Médicos Selenium Python 1	45
Figura 82. Resultado testeo Médicos Selenium Web 1	45
Figura 83. Error Médicos detectado con Selenium.....	46
Figura 84. Error Médicos solucionado	46
Figura 85. Resultado testeo Médicos Selenium Python 2	46
Figura 86. Previo testeo Médicos Selenium Web	47
Figura 87. Resultado testeo Médicos Selenium Web 2	47
Figura 88. Resultado testeo Médicos Selenium Web 3	47
Figura 89. Resultado testeo ZBS Selenium Python	47
Figura 90. Resultado testeo ZBS Selenium Web	48
Figura 91. Resultado testeo Centros Selenium Python	48
Figura 92. Resultado testeo Centros Selenium Web 1	48
Figura 93. Resultado testeo Centros Selenium Web 2	48
Figura 94. Resultados testeo Salas Selenium Python	49
Figura 95. Resultados testeo Salas Selenium Web	49
Figura 96. Resultado testeo Dispositivos Selenium Python.....	50
Figura 97. Resultado testeo Dispositivos Selenium Web 1.....	50
Figura 98. Resultado testeo Dispositivos Selenium Web 2.....	50
Figura 99. Información Chrome	55

Tablas

Tabla 1. Resumen tests Selenium 50

Acrónimos

- **API:** Application Programming Interface.
- **ARM:** Advanced RISC Machine.
- **AWS:** Amazon Web Service.
- **BLE:** Bluetooth Low Energy.
- **GUI:** Graphical User Interface.
- **IoT:** Internet of Things.
- **IP:** Internet Protocol.
- **JS:** Java Script.
- **LTS:** Long Term Support.
- **OS:** Operating System.
- **PHP:** Hypertext Preprocessor.
- **PoCAR:** Point-of-care Attention in Rural environments.
- **SDK:** Software Development Kit.
- **UX:** User Experience.
- **ZBS:** Zona básica de salud.

1 Diseños aplicaciones

Se deciden realizar dos aplicaciones diferentes en lugar de una única con distintos motivos ya expuestos anteriormente, el más importante de ellos la experiencia de usuario. La aplicación web para los médicos requería una mayor complejidad para ofrecer todas las opciones y la de pacientes había de ser muy simple e intuitiva.

1.1 Diseño

1.1.1 Moodboard

Para elegir la estética de la página web, se realiza un moodboard con la final de obtener inspiración para desarrollar las páginas.



Figura 1. Moodboard

1.1.2 Paleta de colores

Gracias a la investigación previa (que se puede ver resumida en la Figura 1) se observa como para el ámbito sanitario predominan las tonalidades azules y verdes azuladas. También se observa que no se utilizan colores como el negro, porque tiene una connotación negativa en dicho ámbito, sino que el predominante es el blanco. Por ende, la paleta elegida para el desarrollo del proyecto es la representada en la Figura 2).

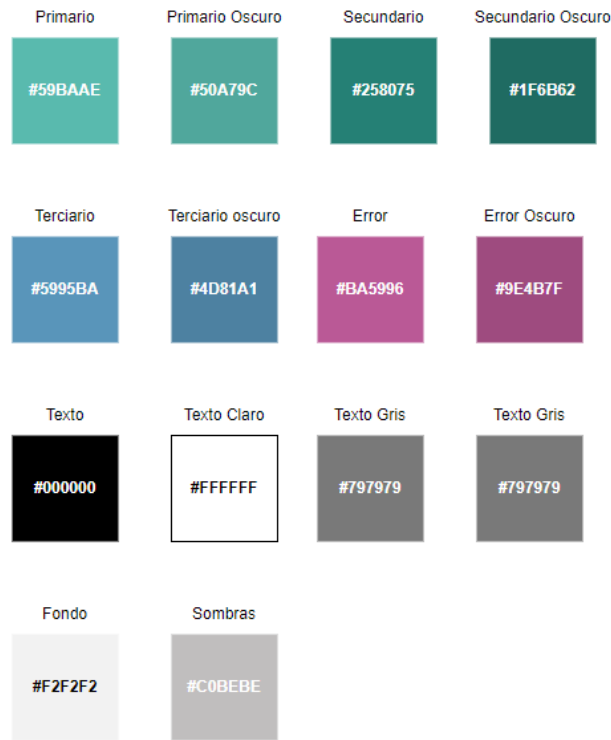


Figura 2. Paleta colores

1.1.3 Fuentes

Se observa en el moodboard representado en la Figura 1 que las fuentes son fuentes claras y fácil de leer. Esto es muy importante sobre todo para los pacientes, ya que necesitan que el texto sea fácilmente legible puesto que podrían tener cualquier tipo de problema visual (causado en muchos casos por la avanzada edad). Algunos tipos de fuentes compatibles con estos requisitos son Arial, Sans Serif y Roboto. Las fuentes se pueden ver en la Figura 3.



Figura 3. Fuentes

Todas las fuentes implementadas en el proyecto han de ser gratuitas.

1.2 Aplicación web médicos

1.2.1 Wireframes

En la Figura 4 se puede observar el login simple. La parte derecha desaparece cuando se utiliza en pantallas móviles con la finalidad de optimizar espacio. En la sección izquierda únicamente se solicitan dos valores, el del correo electrónico y la contraseña, cuando ambos sean correctos se inicia sesión utilizando un token. Cuando se inicia sesión de manera correcta se visualiza la página de la Figura 5.

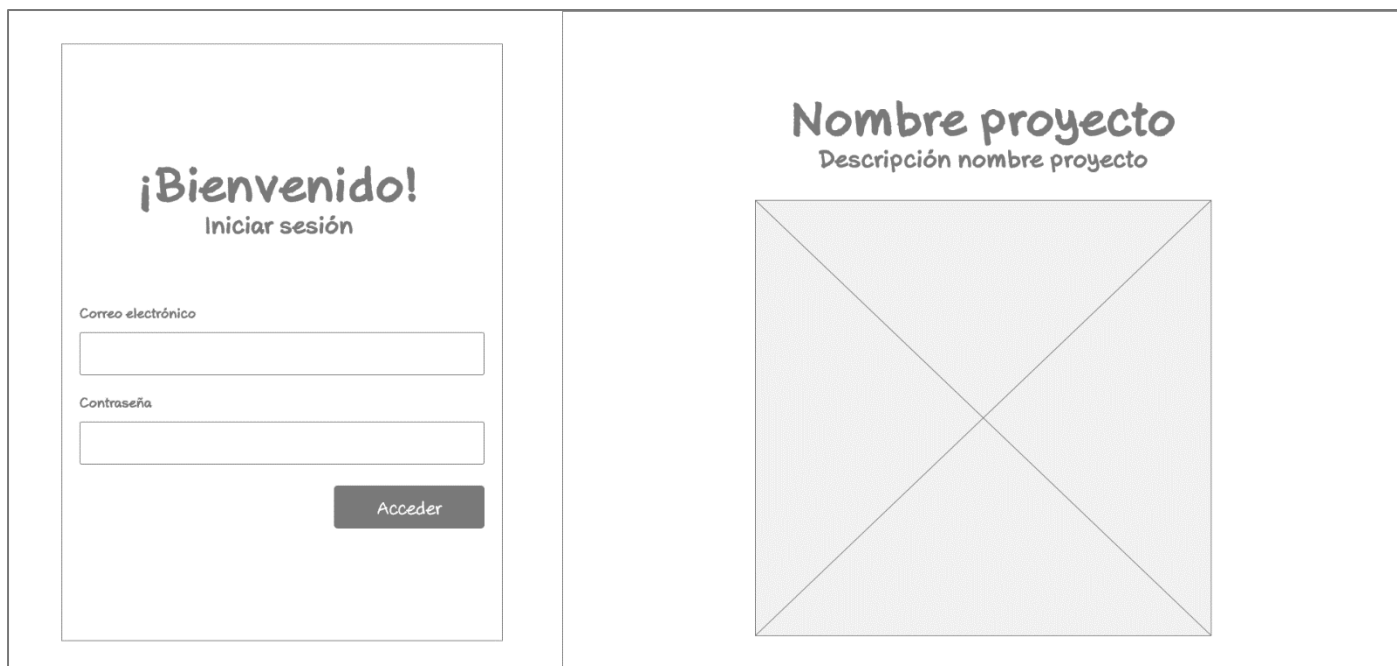


Figura 4. Wireframe inicio sesión médicos

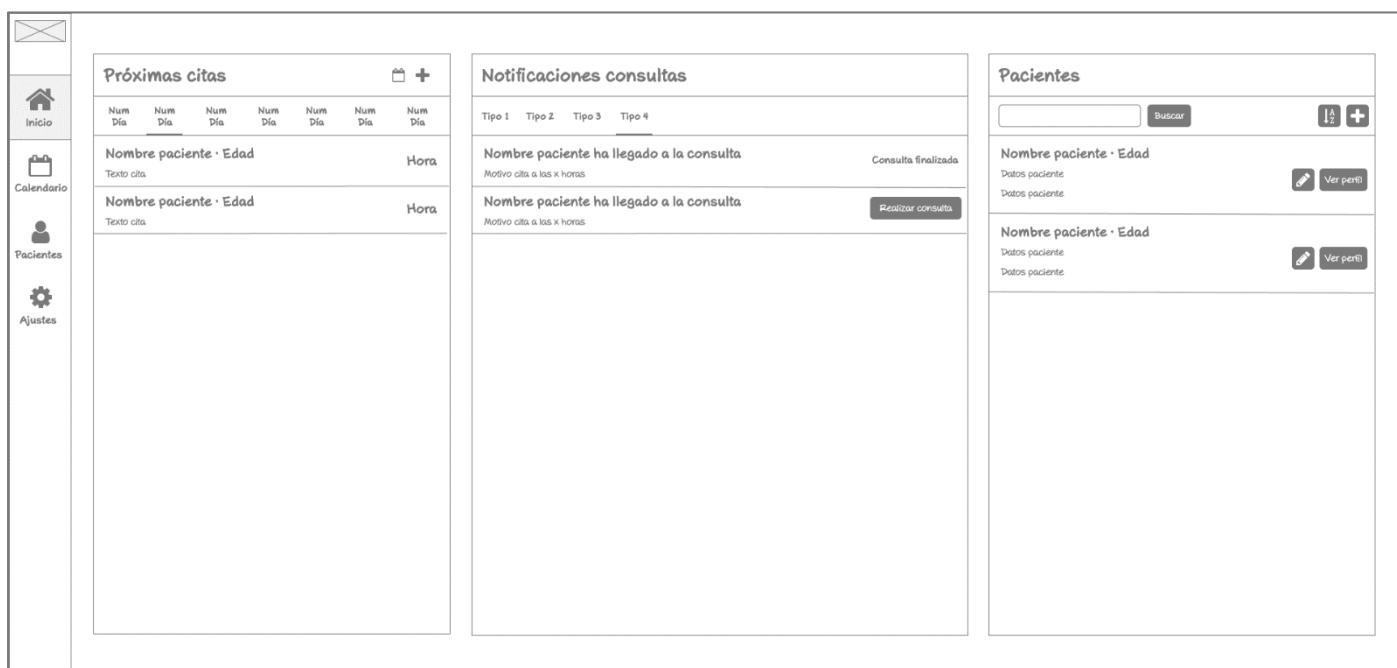


Figura 5. Wireframe inicio médicos

En esta sección Figura 5, se podrá visualizar las citas (panel izquierdo) que tiene dicho médico programadas para toda la semana, además de poder añadir eventos de forma sencilla a dichos días. El botón del calendario redirigirá a la página de este.

El panel central de notificaciones consultas permitirá visualizar las consultas diarias. Este panel será en tiempo real, ya que la idea es que al iniciar sesión en la aplicación un paciente, le llegue de manera instantánea al médico. La parte de pacientes (panel derecho) permitirá visualizar todos los pacientes de las ZBS sobre las cuales tenga permisos de visualización los médicos. También podrá visualizarlos y añadir nuevos.

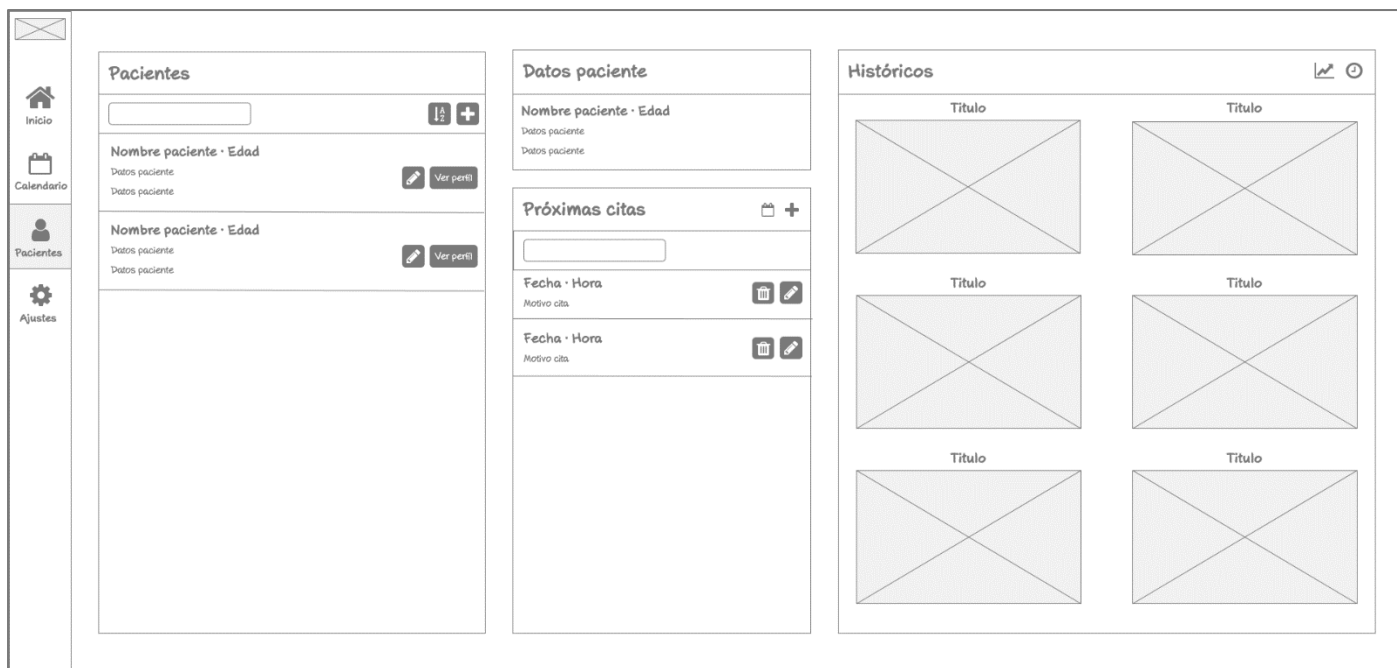


Figura 6. Wireframe pacientes médicos

En la Figura 6 se pueden visualizar todos los datos relacionados con un paciente.

El panel de la izquierda es el mismo que el encontrado en inicio de la Figura 5. En ambos casos al pulsar en ver perfil se redirige visualiza en esta página los datos particulares de dichos pacientes (segunda y tercera columna). En el panel central superior (datos paciente) se puede ver un resumen de los datos del paciente, estos datos son los básicos e indispensables, puesto que no se requieren otros y sería más complejo intentar cumplir las legislaciones sobre protección de datos. Justo debajo, se puede ver la sección de próximas citas, donde se pueden visualizar, editar y añadir citas para el paciente seleccionado. Una de las secciones más importantes, es el panel derecho, donde se quiere visualizar históricos de las distintas medidas tomadas con los dispositivos. En la parte superior del panel, se pueden ver iconos que servirán para filtrar las gráficas.

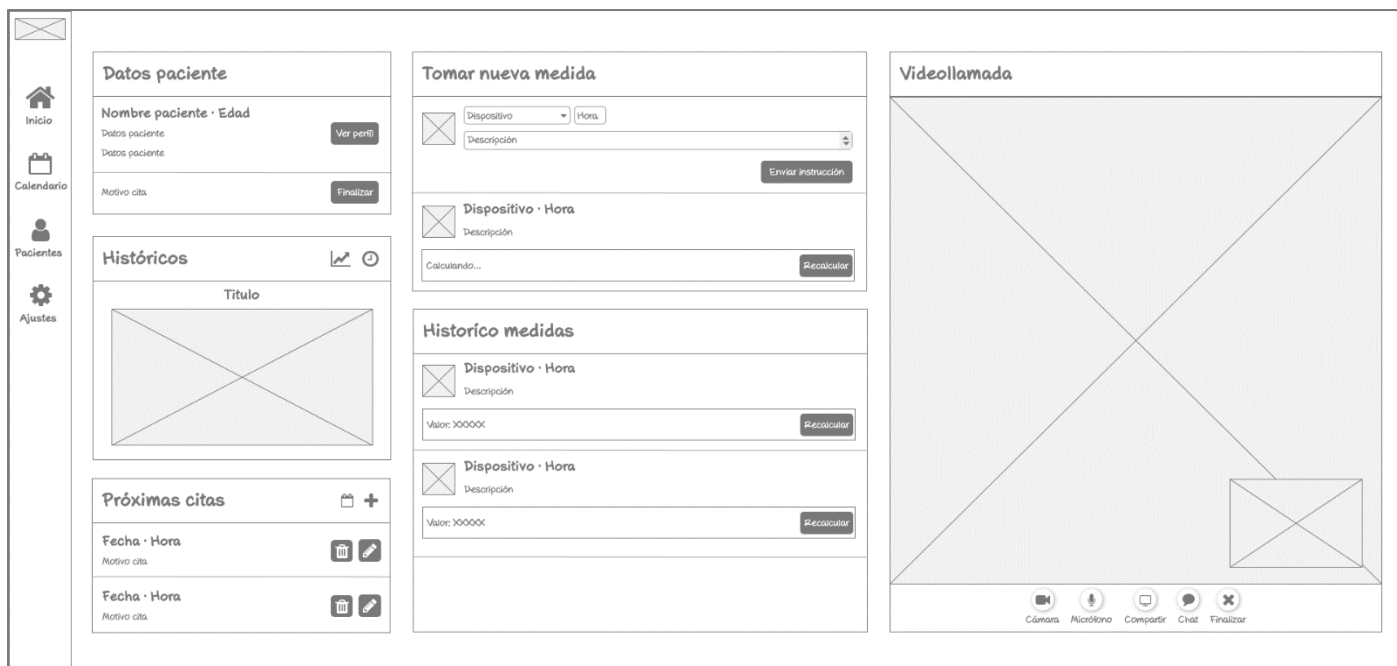


Figura 7. Wireframe consulta médicos

La Figura 7 muestra la página de consulta, donde solo se puede acceder cuando hay una notificación de consulta por parte de paciente. La idea es que en esta página se pueda visualizar todo el contenido relacionado con el paciente necesario para la correcta atención del medio.

Los paneles de históricos, próximas citas y datos paciente son los mismos que en la página de pacientes (ver Figura 6). La única diferencia es que en el panel de Datos paciente (superior izquierdo) se visualiza también el motivo de la cita y un botón para finalizarla. Esto implica que se cierre definitivamente la consulta tanto para el médico como para el paciente.

Los paneles centrales (Tomar nueva medida e Histórico medidas) serán los encargados de enviar ordenes de toma de mediciones tanto al paciente como al coordinador de la sala donde se encuentre. Cuando una medida no sea correcta se podrá recalcular, en caso de serlo, se añadirá al histórico de medidas y se actualizarán las gráficas en tiempo real. El último panel (derecho) era un requisito indispensable por parte de los médicos para que el médico pudiese en todo momento guiar y comunicarse con el paciente. En este caso, se tiene pensado dar total control al médico, pero que el paciente no lo tenga para facilitarle el uso.

Inicio

Calendario

Pacientes

Ajustes

Mes

Semana

Día

Cerrar día

Añadir evento

< Mes Año >

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Fecha

Buscar

1/2

+

Nombre paciente · Edad

Texto cita

Hora

Ver perfil

Nombre paciente · Edad

Texto cita

Hora

Ver perfil

Nombre paciente · Edad

Texto cita

Hora

Ver perfil

Figura 8. Wireframe calendario médicos (v1)

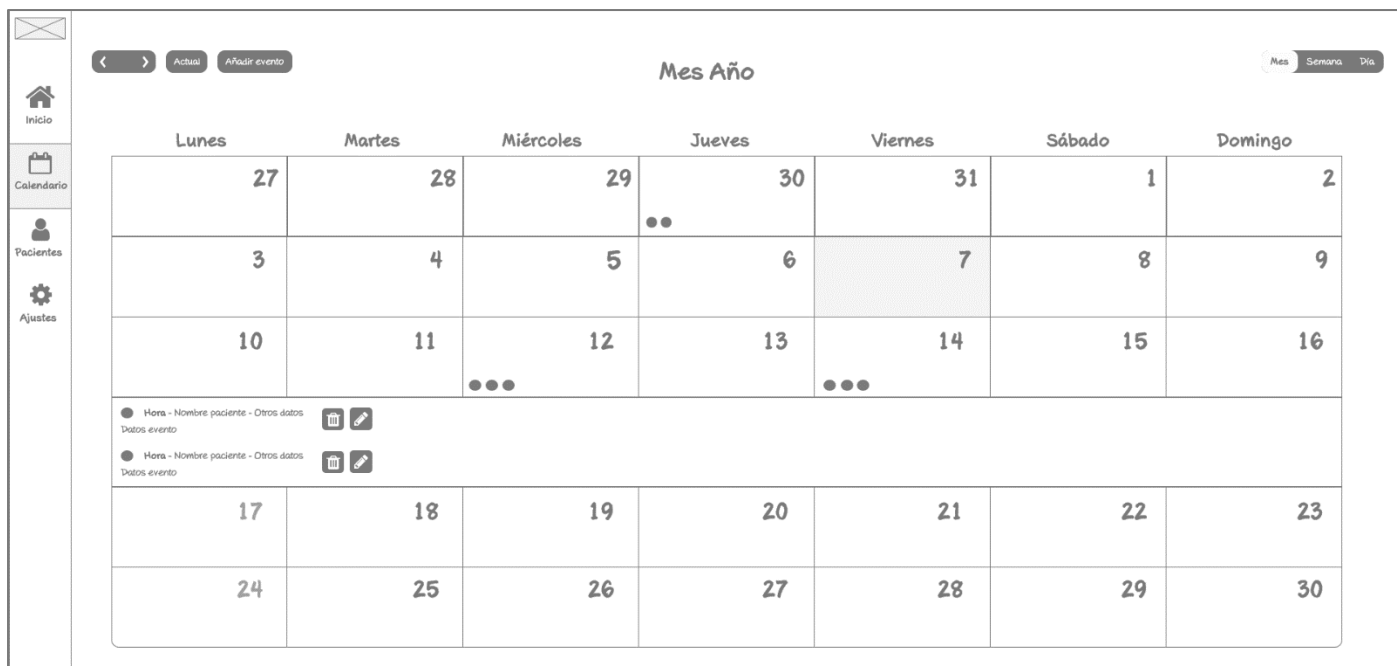


Figura 9. Wireframe calendario médicos (v2)

La Figura 8 y Figura 9 pertenecen a la sección de calendario. La Figura 9 es el formato visual que más se adapta a la API además de optimizar en mayor cantidad el espacio. En ambos casos se pueden ver botones para cambiar de mes, para centrar el día en el actual, para cambiar el tipo de vista (mensual, semanal o diaria) y para añadir un nuevo evento.

La API permite arrastrar eventos (los puntos que aparecen en cada día) de un día a otro de forma simple para actualizar (en el caso de la vista de día o semana se pueden redimensionar también). Al pulsar en un día que tenga eventos, se debería poder visualizar más información de dichos eventos así como editarlos o eliminarlos.

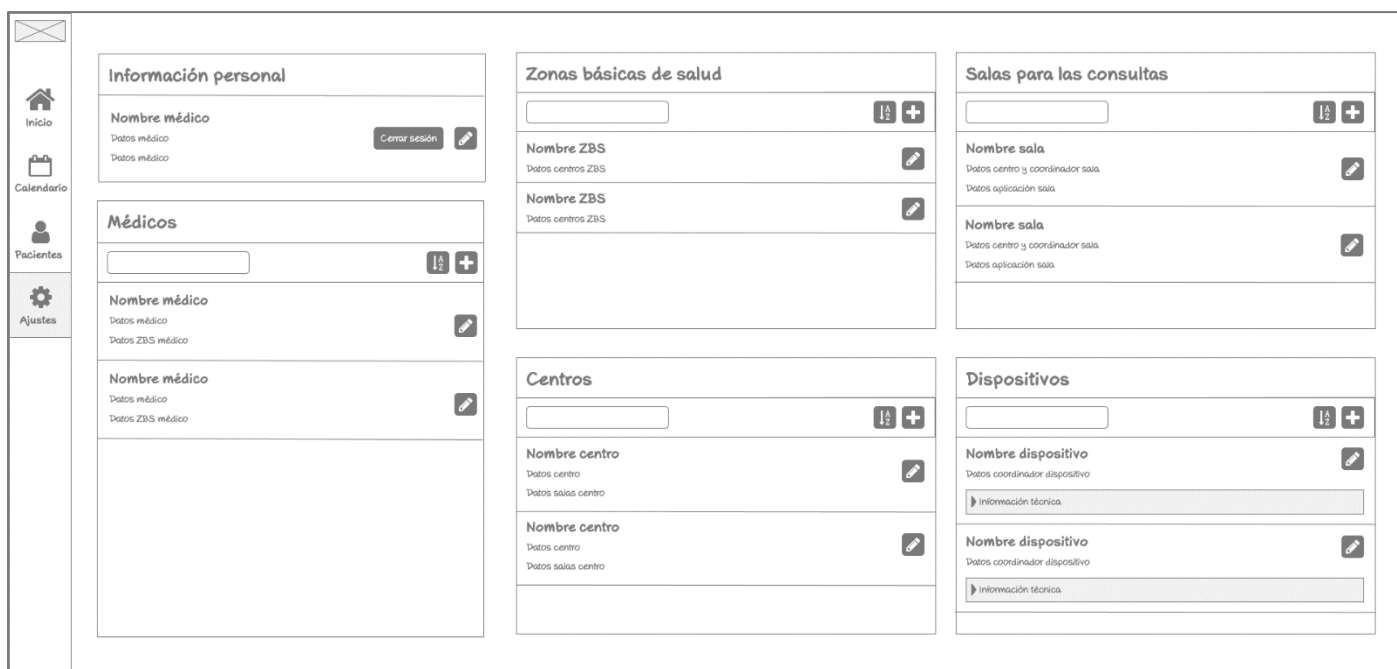


Figura 10. Wireframe ajustes médicos

La página de la Figura 10 es la que permite administrar todo el sistema. Desde ella se puede modificar la información personal del propio médico así como cerrar sesión. En caso de que el médico tenga permisos de administrador podrá editar y crear: médicos, zonas básicas de salud (ZBS), centros, salas para las consultas (correspondientes a un determinado centro) y dispositivos. Cada uno de los paneles tiene filtros para facilitar la búsqueda ante la posibilidad de manejar grandes cantidades de datos.

Añadir/Editar evento

X

Correo electrónico

Centro

Doctor

Descripción

Color primario

Color secundario

Fecha inicio

Fecha fin

Crear/Editar

Añadir/Editar paciente

X

Nombre

Apellidos

Edad

SIP

ZBS

Centro

Doctor

Crear/Editar

Añadir/Editar dispositivo

X

Coordinador

MAC

Tipo

Información técnica

Crear/Editar

Añadir/Editar zona básica de salud

X

Nombre

Centros

Centro Nombre

Añadir nuevo centro

Crear/Editar

Añadir/Editar sala

X

Nombre

Centro

MAC coordinador

Ip aplicación

MAC aplicación

Crear/Editar

Añadir/Editar centro

X

Nombre

Teléfono

Zona básica de salud

Salas

Sala Nombre

Crear/Editar

Añadir/Editar médico

X

Nombre

Apellidos

Email

Contraseña

Repetir contraseña

Permisos de administrador

☒

Si

ZBS

ZBS Nombre

Añadir nueva ZBS

Crear/Editar

Figura 11. Wireframe modales médicos

En la Figura 11 se pueden observar los principales diseños de los modelos necesarios para añadir y editar los distintos tipos de elementos.

1.2.2 Mockups

Se puede observar en cada una de las consiguientes figuras, la transformación del wireframe en mockup una vez aplicados los estilos pertinentes. En este caso, también se indica alguna información básica relevante que se mostrará como por ejemplo los datos de los pacientes.



Figura 12. Mockup inicio sesión médicos

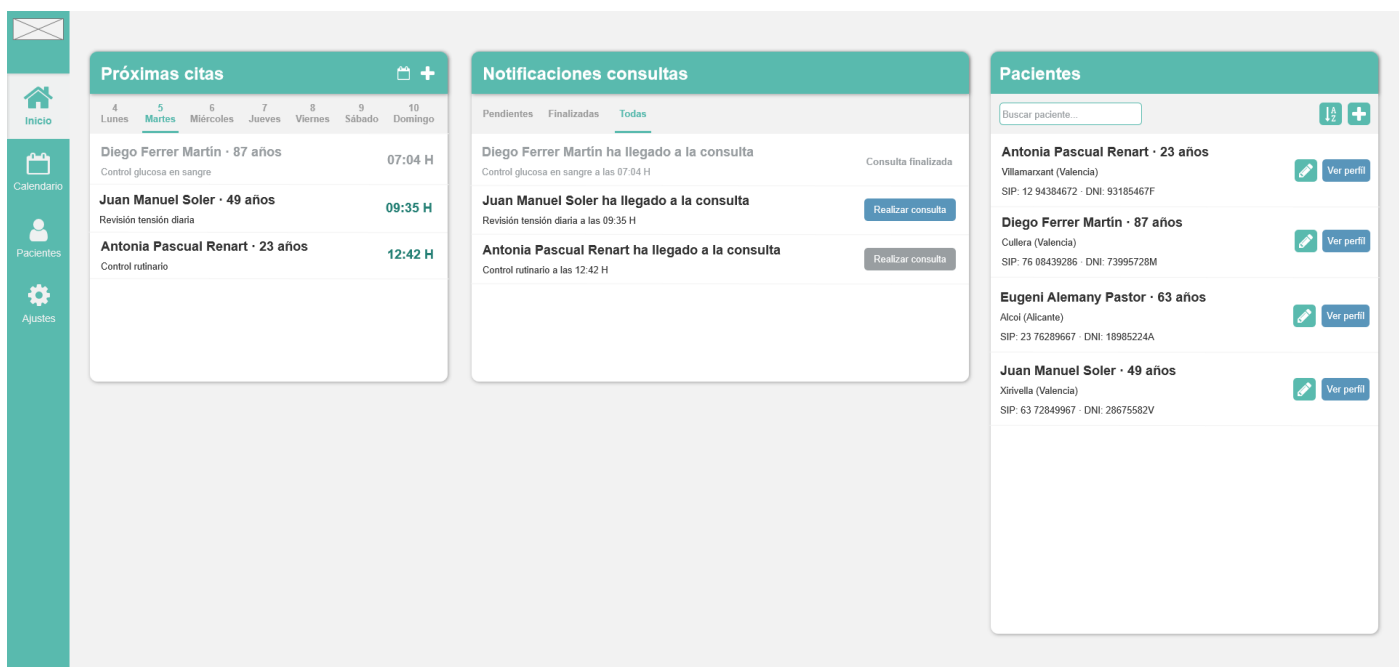


Figura 13. Mockup inicio médicos

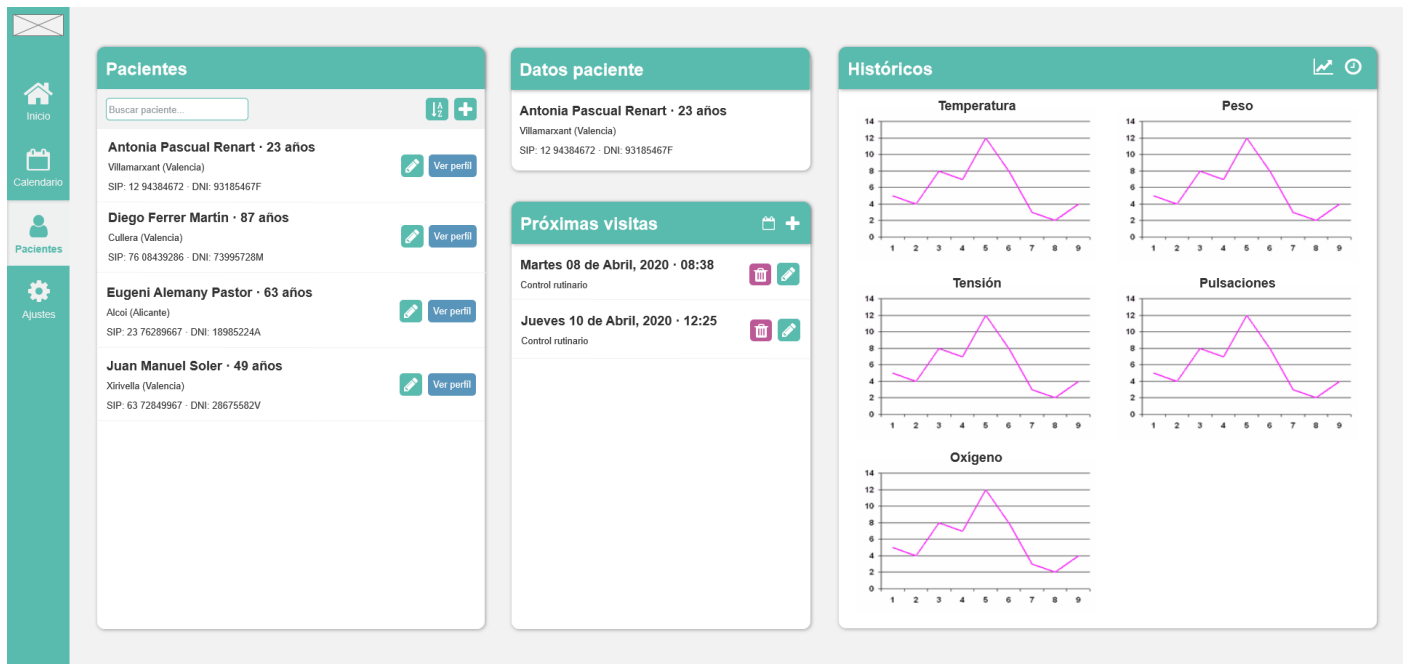


Figura 14. Mockup pacientes médico

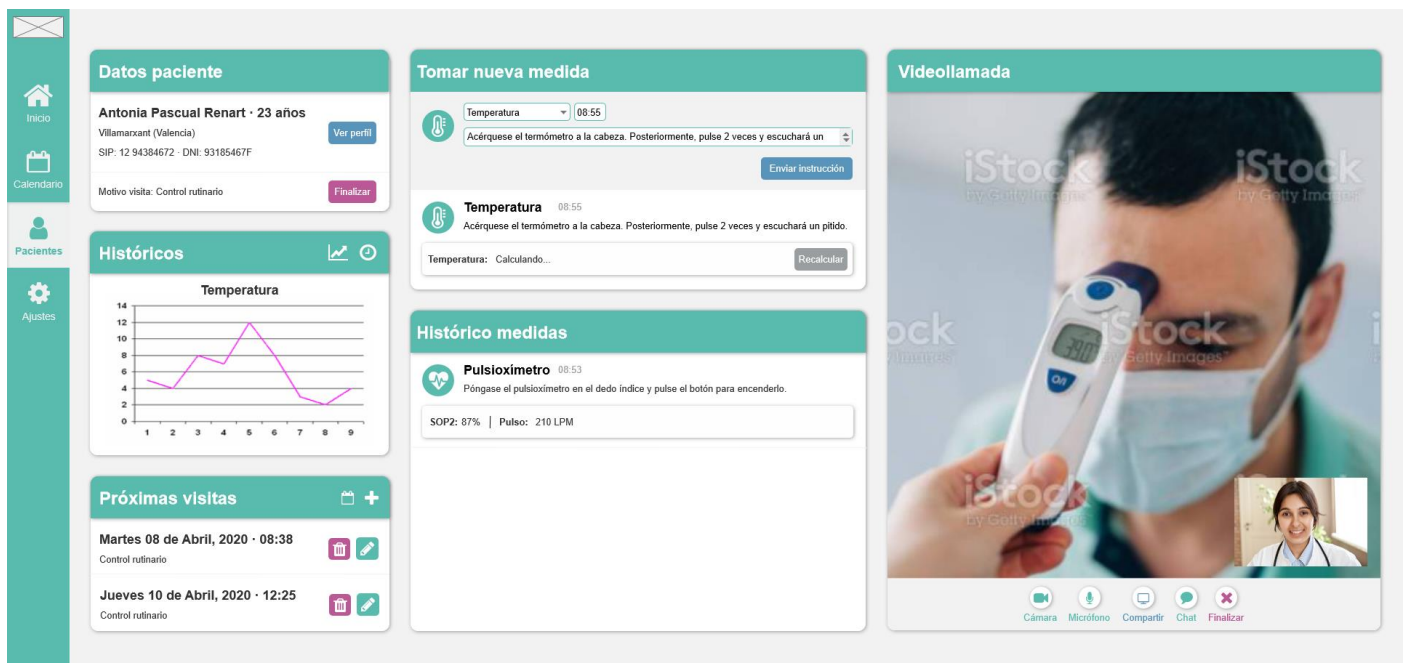


Figura 15. Mockup consulta médicos

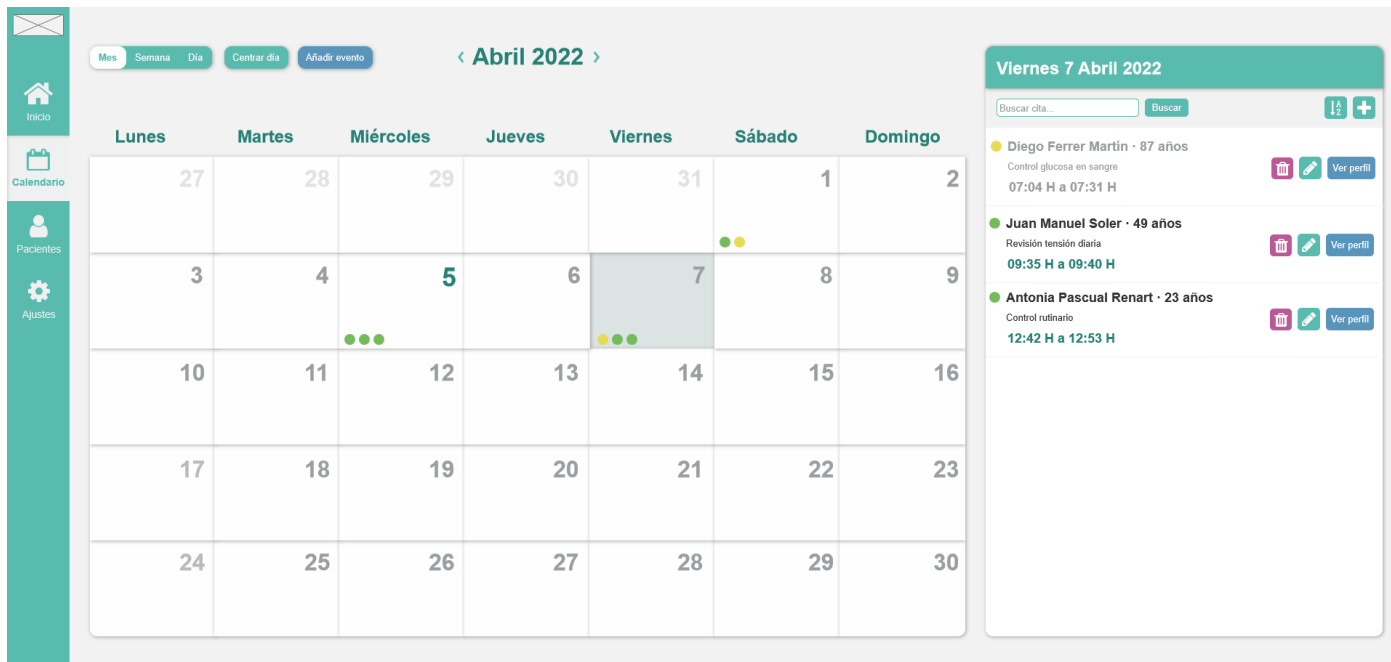


Figura 16. Mockup calendario médicos (v1)

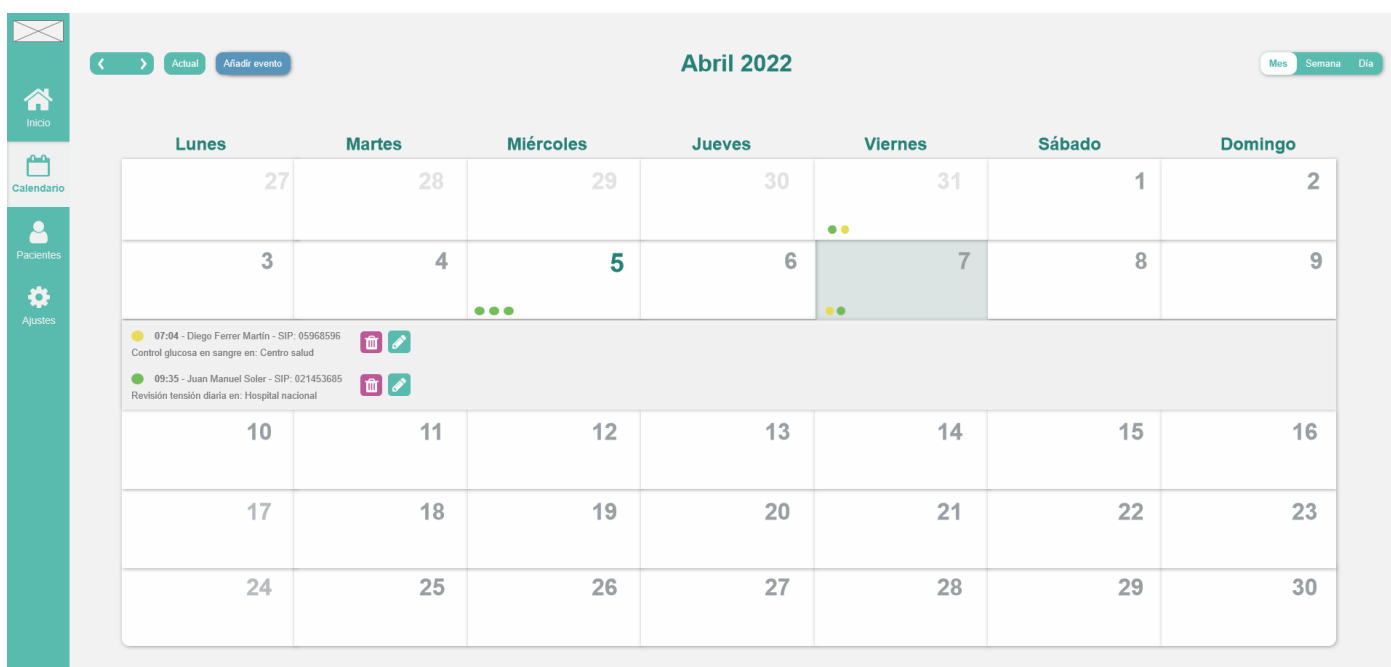


Figura 17. Mockup calendario médicos (v2)

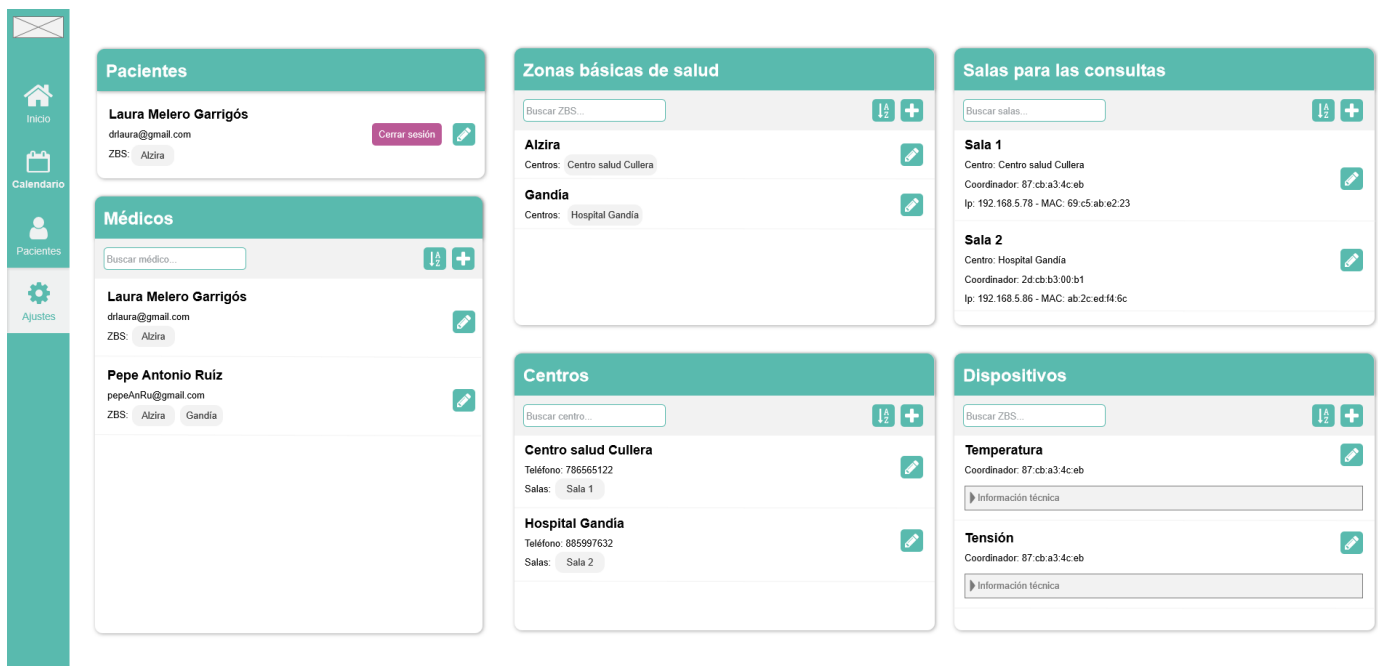


Figura 18. Mockup ajustes médicos

Añadir/Editar evento

Paciente

Seleccione un paciente

Centro

Seleccione un centro

Doctor

Laura Melero Garrigós

Descripción

Descripción cita

Color primario

Color secundario

Fecha inicio

DD/MM/YYYY HH:MM

Fecha fin

DD/MM/YYYY HH:MM

Crear/Editar

Añadir/Editar paciente

Nombre

Nombre paciente

Apellidos

Apellidos paciente

Edad

Edad paciente

SIP

SIP o identificador paciente

ZBS

Seleccione una zona

Centro

Seleccione un centro

Doctor

Seleccione un doctor

Crear/Editar

Añadir/Editar zona básica de salud

Nombre

Nombre zona

Centros

Centro Nombre

Crear/Editar

Añadir/Editar médico

Nombre

Nombre médico

Apellidos

Apellidos médico

Email

Correo electrónico

Contraseña

Contraseña

Repetir contraseña

Repetir contraseña

Permisos de administrador

☒ SI

ZBS

ZBS Nombre Añadir nueva ZBS

Crear/Editar

Añadir/Editar dispositivo

Coordinador

Seleccione un coordinador

MAC

Dirección física del dispositivo

Tipo

Seleccione un tipo de dispositivo

Información técnica

Crear/Editar

Añadir/Editar sala

Nombre

Nombre sala

Centro

Seleccione un centro

MAC coordinador

Dirección física

Ip aplicación

Dirección IP

MAC aplicación

Dirección física

Crear/Editar

Añadir/Editar centro

Nombre

Nombre centro

Teléfono

Número de teléfono

Zona básica de salud

Seleccione una ZBS

Salas

Sala Nombre

Crear/Editar

Figura 19. Mockup modales médicos

1.2.3 Resultado final

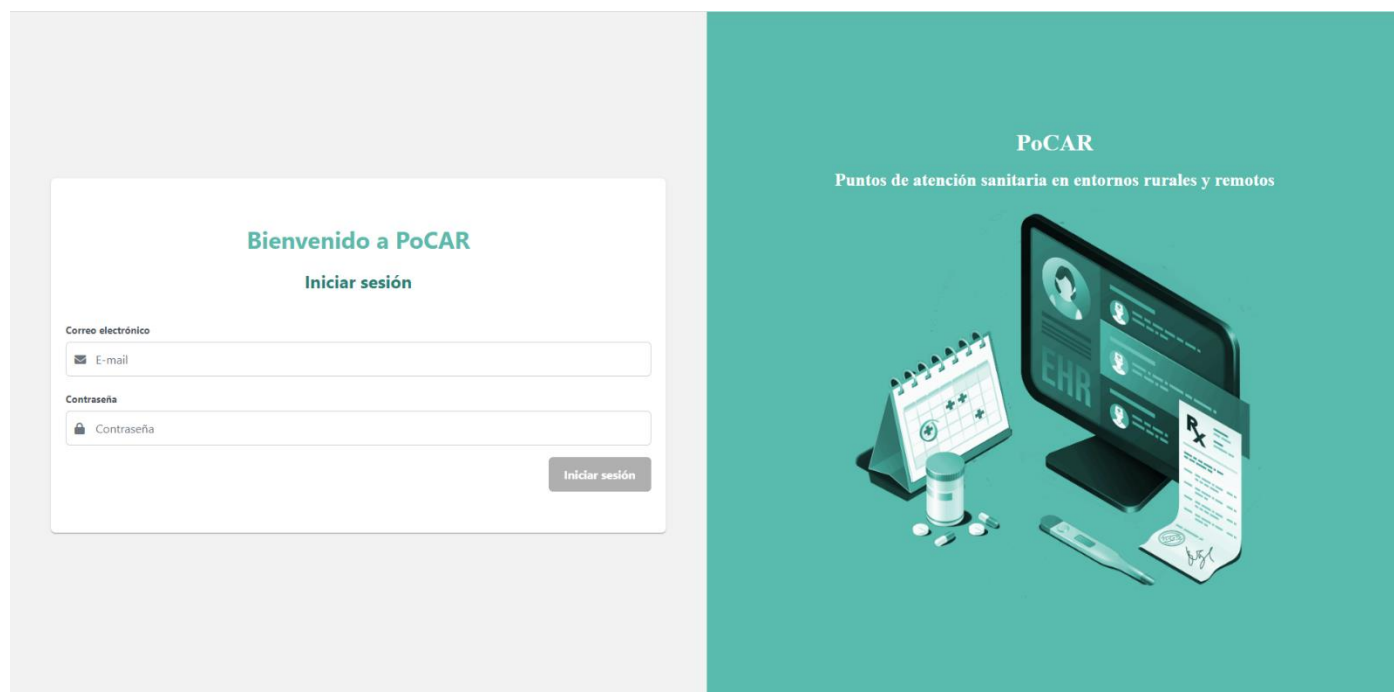


Figura 20. Resultado final inicio sesión médicos

Tal como se había explicado anteriormente, la primera página que se puede observar un pequeño login que pide el correo electrónico y la contraseña del paciente. En caso de que uno de los parámetros sea erróneo saldrá un mensaje de error notificándolo. En caso de que sea correcto se visualizará la Figura 21. En ella se visualizan los paneles que se han podido visualizar en los wireframes y mockups anteriores.

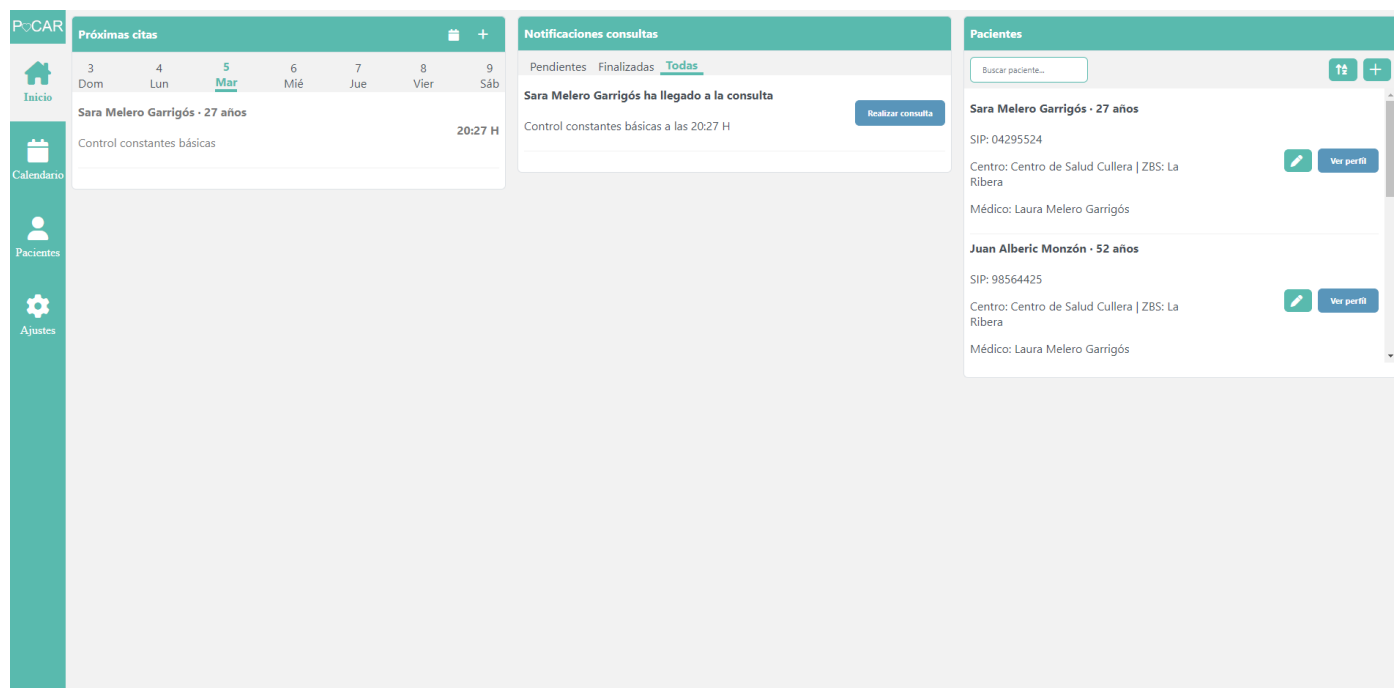


Figura 21. Resultado final inicio médicos (consulta por realizar)

En la Figura 21, a la izquierda del todo se ve el menú donde se puede ver que se muestra seleccionada en todo momento la página en la que se encuentra el médico. En el panel de *Próximas citas* se pueden ver 2 iconos (el del calendario que redirige a la página de este y el del + que permite crear citas para el día seleccionado), al hacer click en el segundo, se puede visualizar el modal de la Figura 22. En dicho modal se puede ver que el campo de Doctor no se

puede modificar ya que un médico solo puede crear eventos para él mismo. Adicionalmente, se ve como en los campos aparece por defecto el día seleccionado en el panel de *Próximas citas*.

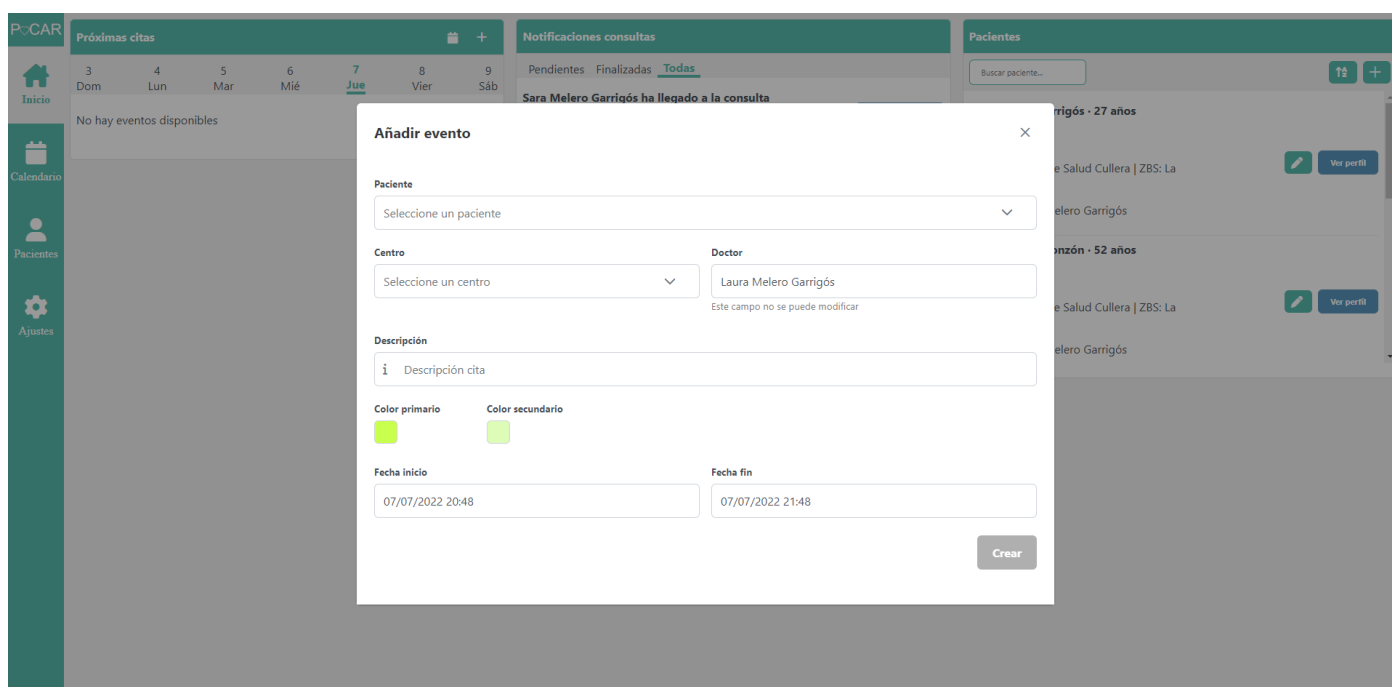


Figura 22. Resultado final inicio médicos (añadir evento)

El panel de pacientes tiene 2 tipos de filtros para facilitar la búsqueda de un paciente (se pueden ordenar alfabéticamente por el nombre y los apellidos o se puede buscar cualquier campo presente en el panel como el SIP, la edad, la ZBS, el centro o el médico). En este panel también se pueden crear o editar pacientes (los modales son los mismos que los de la Figura 24 y la Figura 25). En caso de hacer click en el botón de *Ver perfil* se redirigirá a la página de pacientes tal y como se puede ver en la Figura 26.

En el panel de Notificaciones consultas se puede observar que se clasifican en Pendientes, Finalizadas o Todas. En este panel aparecen todas las notificaciones del día concurrente. En este caso vemos como una paciente que tiene una cita para este día ha llegado a la consulta y lo ha notificado al médico. Cuando este quiera, podrá inicializar la consulta pulsando en el botón de realizar consulta, que llevará a la Figura 31.

En la página de pacientes, lo primero que se observa es el panel de *Pacientes*, que ofrece el listado de pacientes. Este panel es el mismo que usado en la página de inicio (Figura 21). Cuando se accede desde el menú lateral no habrá ningún paciente seleccionado tal como se puede ver en la Figura 23, pero si se accede desde el botón de Ver perfil, el resultado será el presentado en la Figura 26. Hay que destacar que en el caso de la Figura 23, tanto los botones de añadir una cita (panel *Próximas citas*) como los de los filtros (panel *Históricos*) están desactivados. Los modales de Editar paciente y Crear paciente, se pueden visualizar en la Figura 25 y Figura 24. Como detalle a destacar, es que una vez creado el paciente, no se puede editar su SIP ya que es su identificador único.

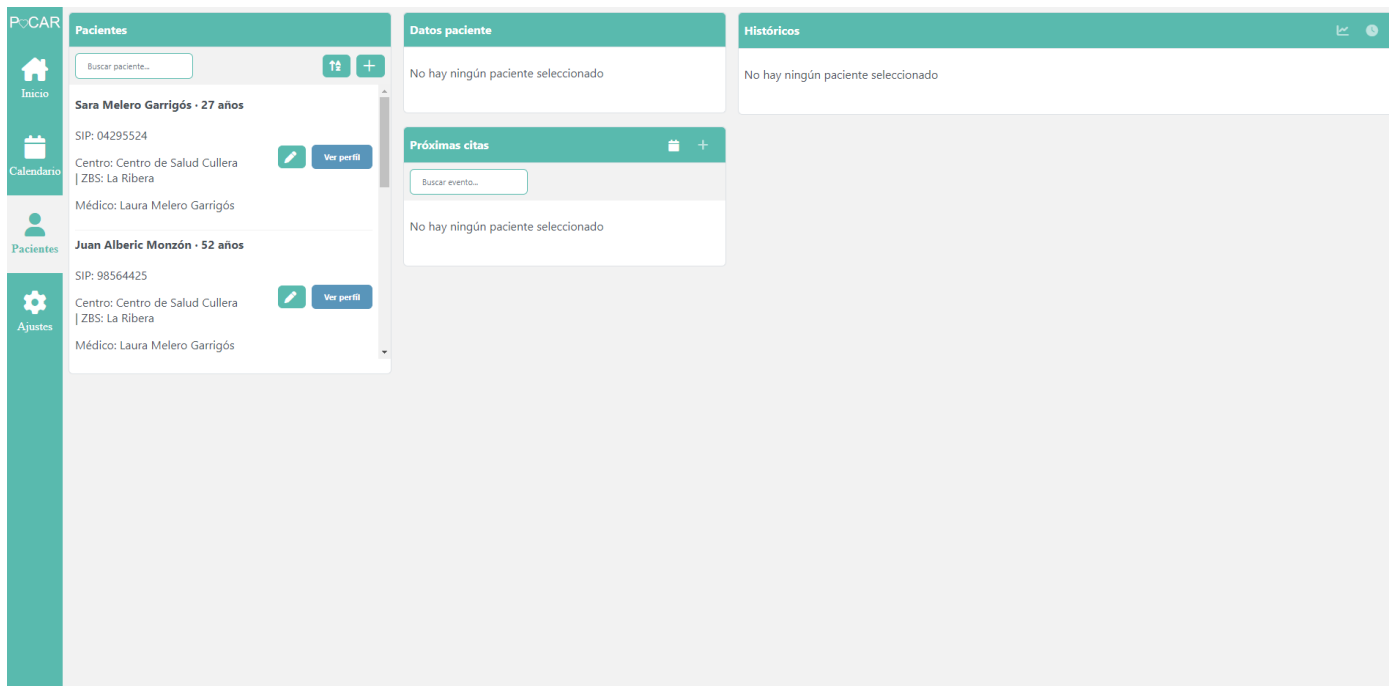


Figura 23. Resultado final pacientes médicos (paciente no seleccionado)

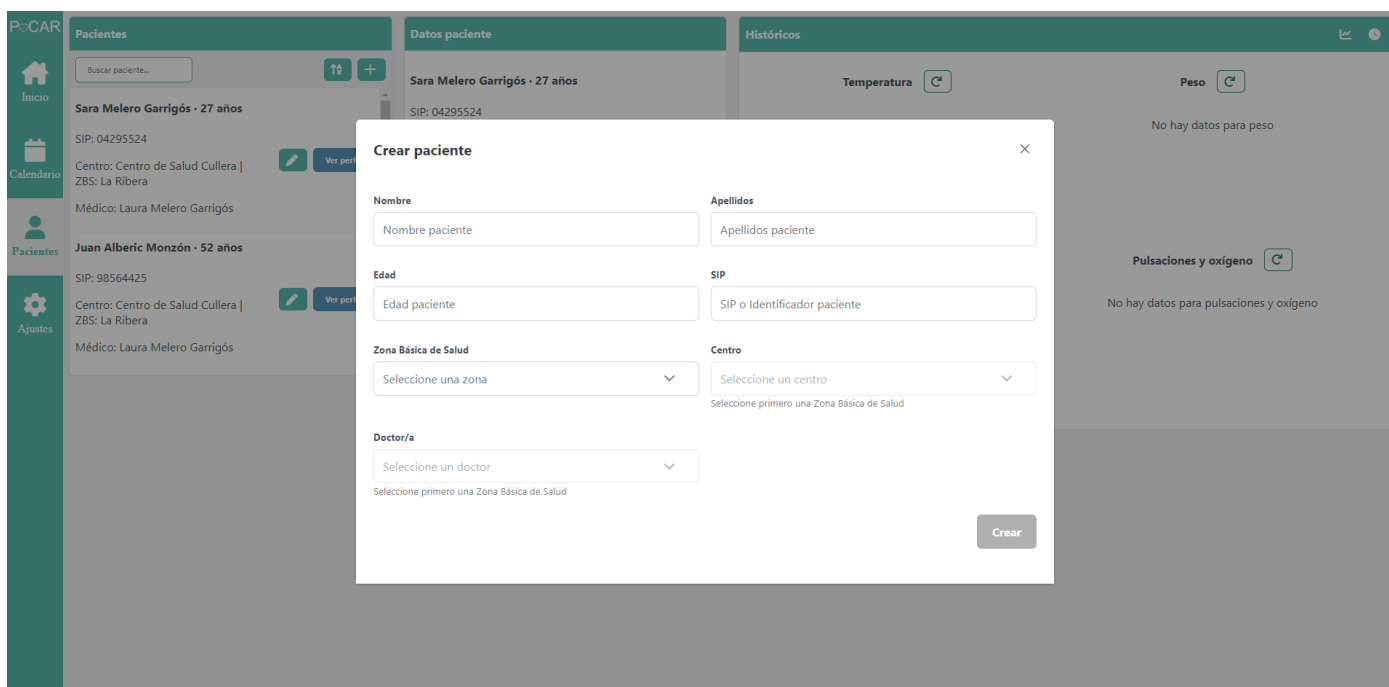


Figura 24. Resultado final pacientes médicos (crear paciente)

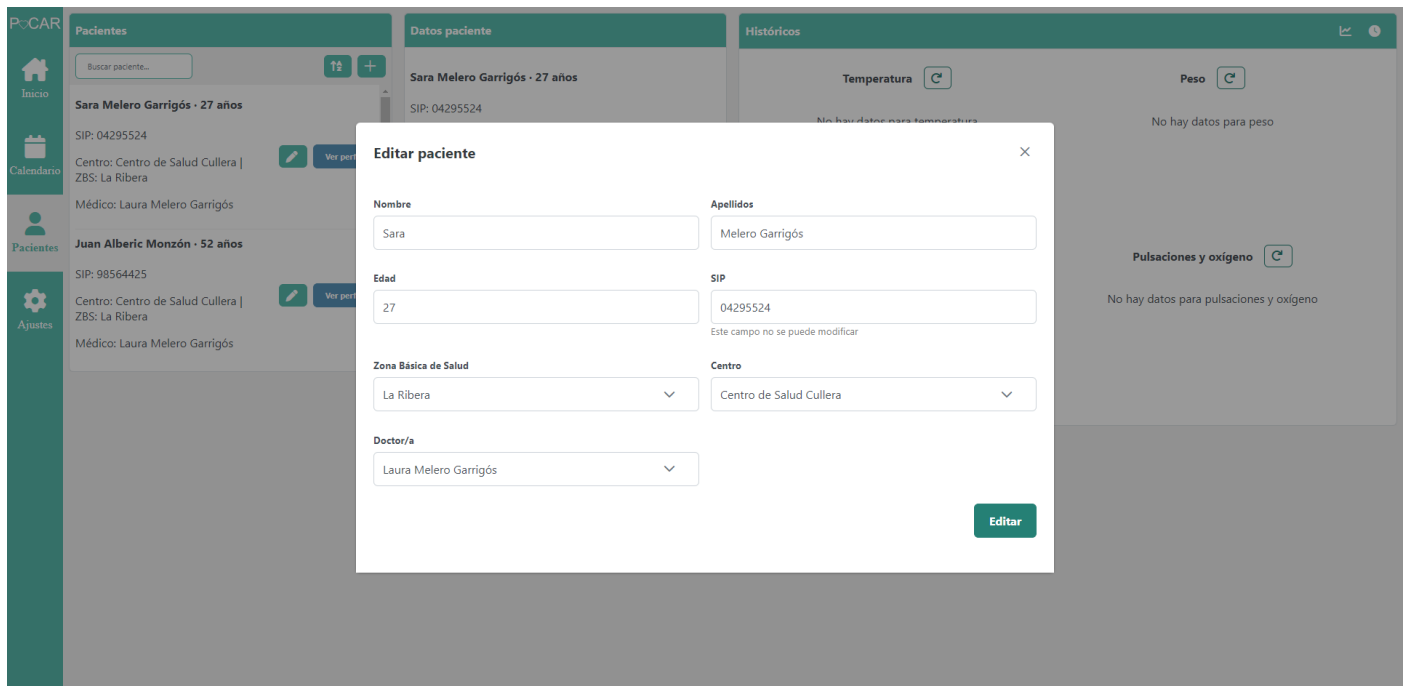


Figura 25. Resultado final pacientes médicos (editar paciente)

Al tener un paciente seleccionado, se podrán ver todos los datos de dicho paciente (panel Datos paciente) así como sus citas (*Próximas citas*) o sus datos personales. Se pueden crear y editar eventos de igual manera que desde la página de inicio mediante el panel de *Próximas citas*, con la única condición de que tal como se observa en la Figura 27 y en la Figura 28 el usuario también está puesto por defecto (de esta forma se minimiza la posibilidad de errores).

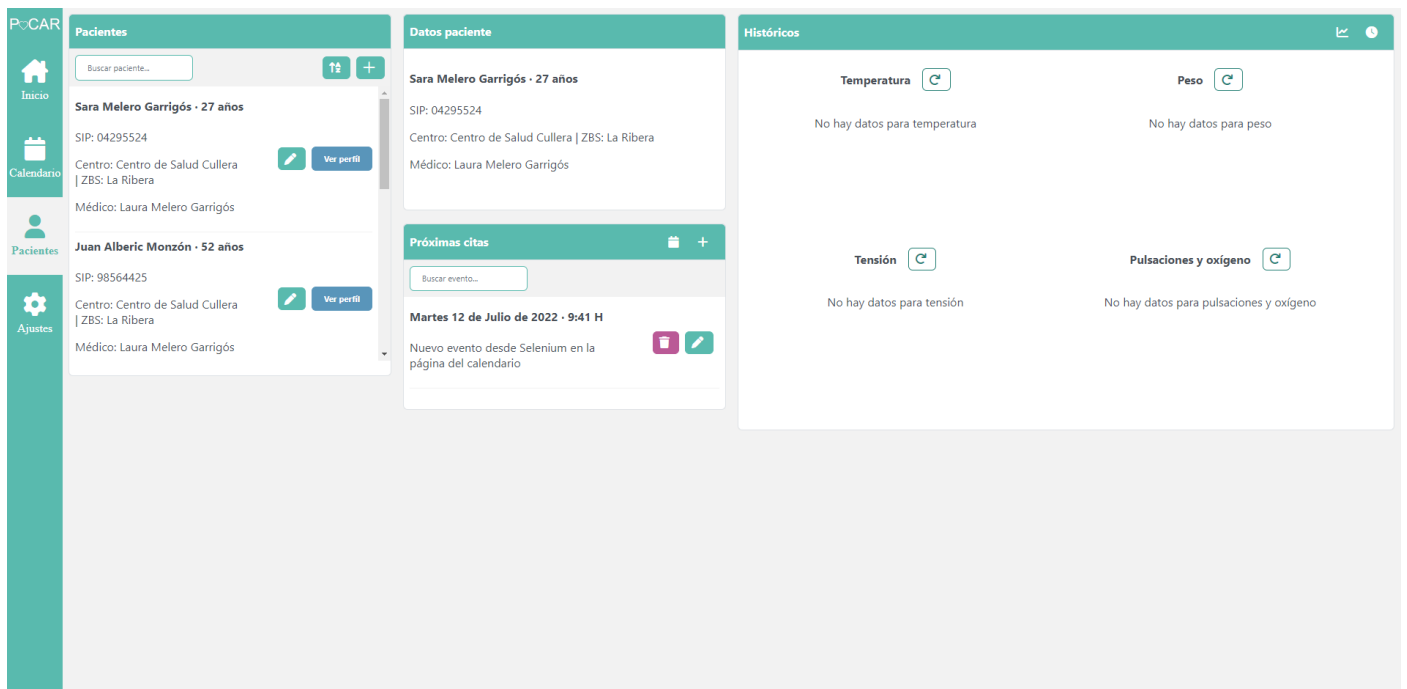


Figura 26. Resultado final pacientes médicos (paciente seleccionado)

Pacientes

Buscar paciente...

Sara Melero Garrigós - 27 años
 SIP: 04295524
 Centro: Centro de Salud Cullera | ZBS: La Ribera
 Médico: Laura Melero Garrigós

Juan Alberic Monzón - 52 años
 SIP: 98564425
 Centro: Centro de Salud Cullera | ZBS: La Ribera
 Médico: Laura Melero Garrigós

Añadir evento

Paciente
 Sara Melero Garrigós - 04295524
 Este campo no se puede modificar

Centro
 Centro de Salud Cullera

Doctor
 Laura Melero Garrigós
 Este campo no se puede modificar

Descripción
 Descripción cita

Color primario
 Color secundario

Fecha inicio
 DD/MM/YYYY HH:MM

Fecha fin
 DD/MM/YYYY HH:MM

Crear

Figura 27. Resultado final pacientes médicos (añadir evento)

Pacientes

Buscar paciente...

Sara Melero Garrigós - 27 años
 SIP: 04295524
 Centro: Centro de Salud Cullera | ZBS: La Ribera
 Médico: Laura Melero Garrigós

Juan Alberic Monzón - 52 años
 SIP: 98564425
 Centro: Centro de Salud Cullera | ZBS: La Ribera
 Médico: Laura Melero Garrigós

Editar evento

Paciente
 Sara Melero Garrigós - 04295524
 Este campo no se puede modificar

Centro
 Centro de Salud Cullera

Doctor
 Laura Melero Garrigós
 Este campo no se puede modificar

Descripción
 Nuevo evento desde Selenium en la página del calendario

Color primario
 Color secundario

Fecha inicio
 07/12/2022 09:41

Fecha fin
 07/12/2022 10:41

Modificar

Figura 28. Resultado final pacientes médicos (editar evento)

Por último se pueden visualizar los datos de las mediciones tomadas al paciente mediante el panel de *Históricos*. Tal como se ha mencionado anteriormente incluye dos filtros. En de la Figura 29 se puede modificar el intervalo temporal en que se quieren visualizar los datos (los ejes se modifican para facilitar la lectura), la opción de *Personalizado* muestra dos inputs que permiten introducir dos fechas entre las cuales se quiere mostrar los datos. En el de la Figura 30 se pueden seleccionar los gráficos que se quiera mostrar y los que no, esto es muy útil para la sección de consulta ya que hay menos espacio.

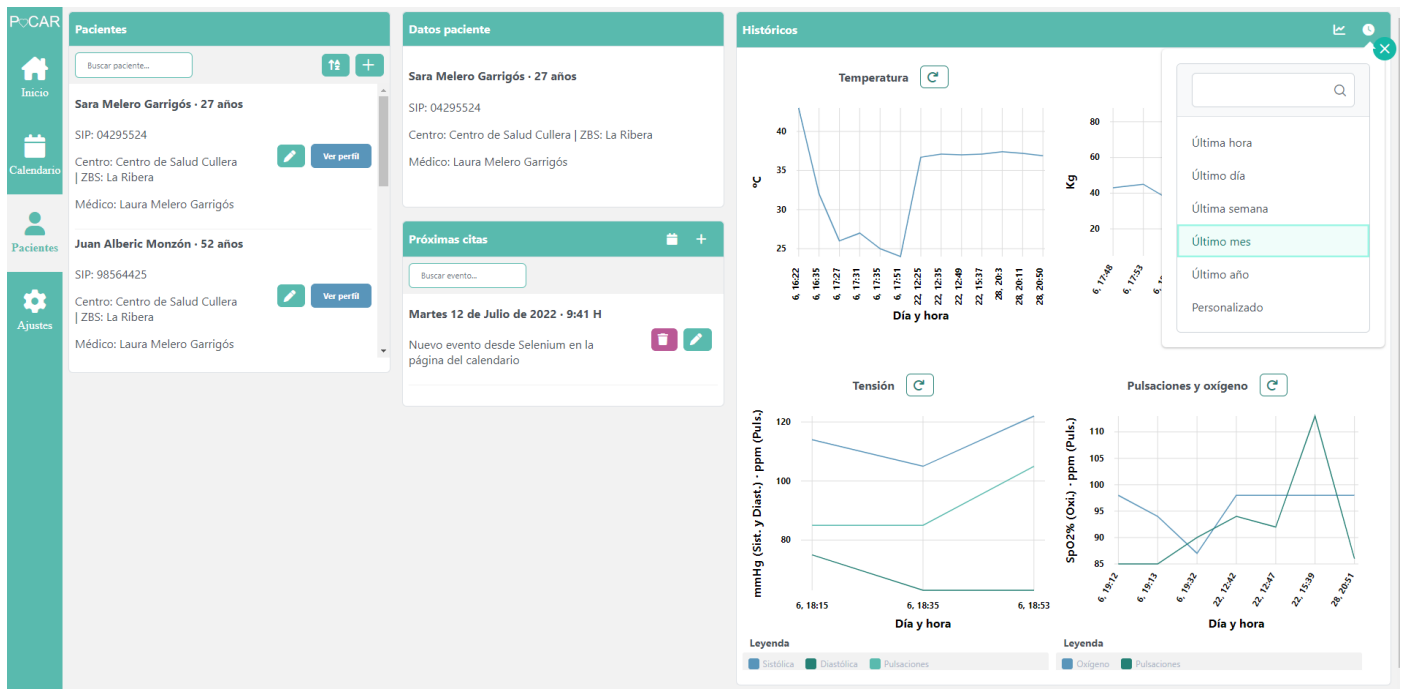


Figura 29. Resultado final pacientes médicos (filtro tiempo históricos)

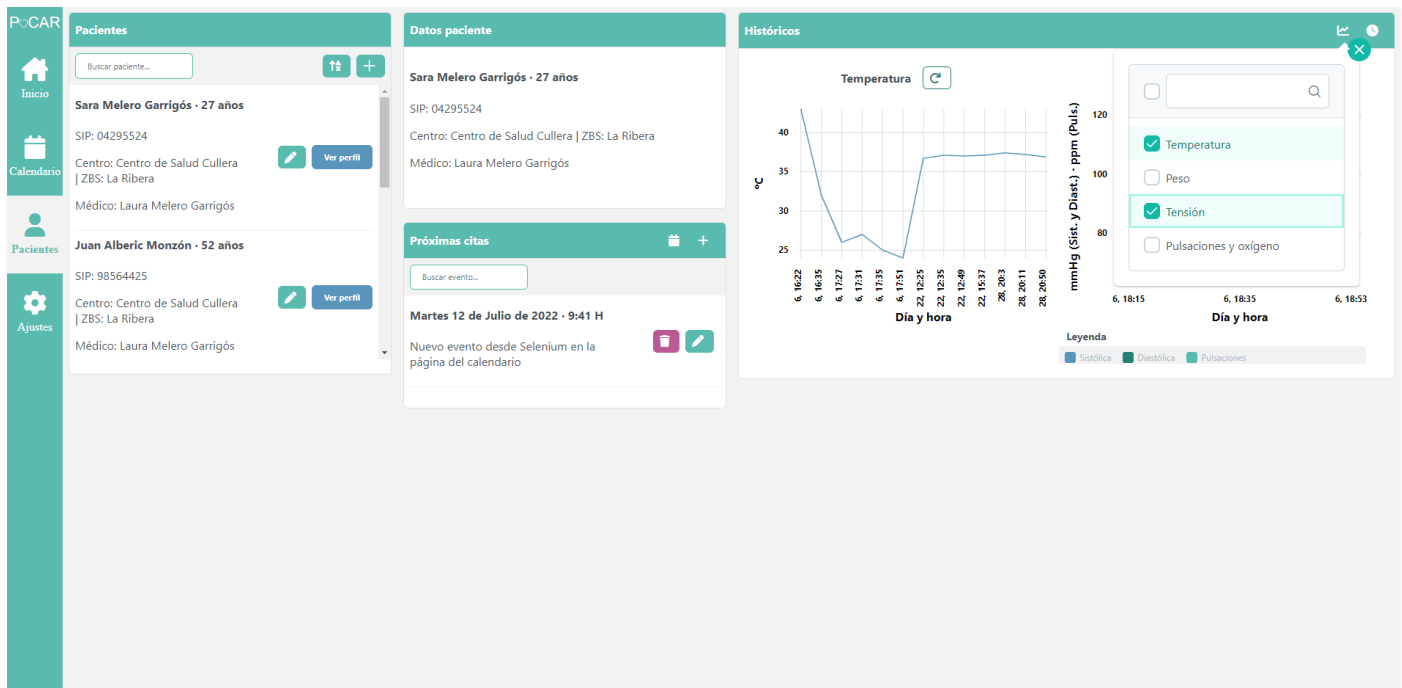


Figura 30. Resultado final pacientes médicos (filtro tipos gráficos)

Tal como se había mencionado anteriormente, al llegar una notificación de consulta, desde el panel de inicio Figura 21 se puede empezar la consulta pulsando el botón de *Realizar Consulta*. Este botón estará siempre disponible en dos casos: la consulta no haya sido finalizada (mediante el botón de *Finalizar visita* de la página de la consulta) y la sala donde el paciente ha notificado la llegada no esté siendo usada en ese momento para otra consulta. Si se cumplen ambas condiciones, se llevará a la página de consulta (Figura 31).

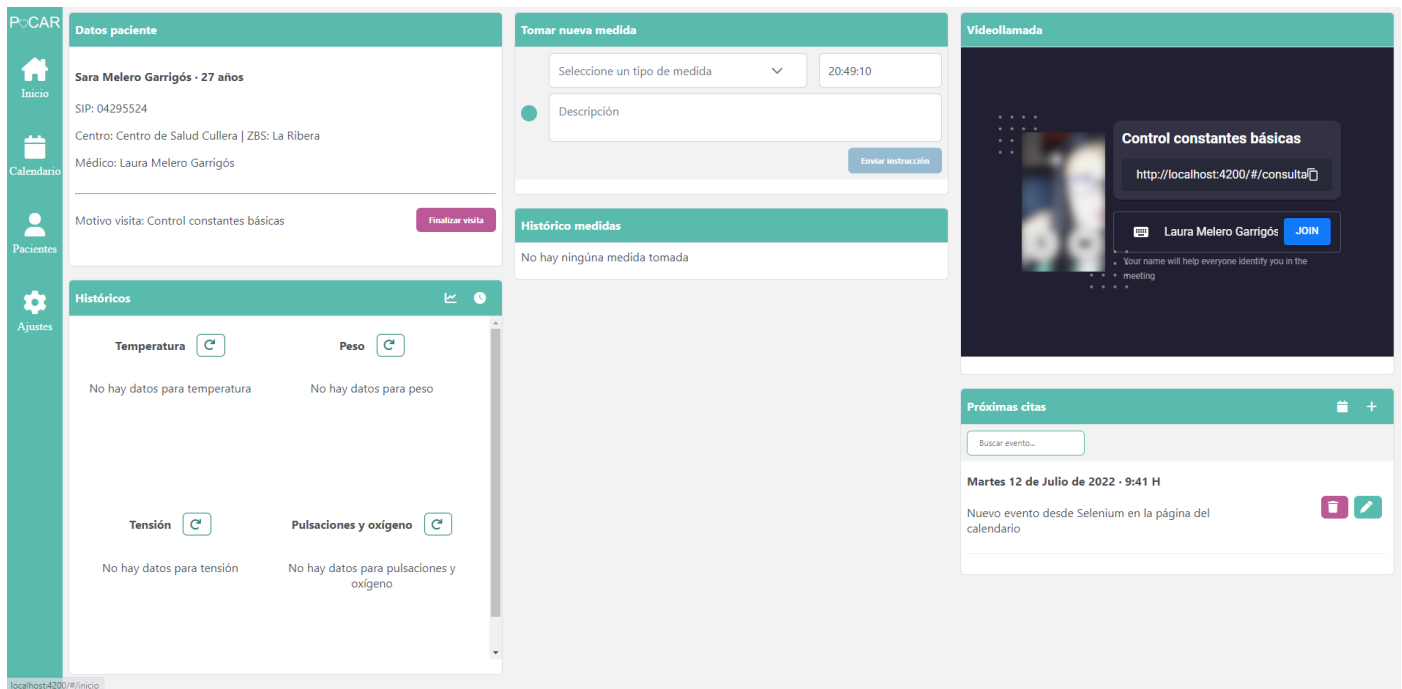


Figura 31. Resultado final consulta médicos (consulta general)

En esta página lo primero que se puede observar son los datos del paciente (es el mismo panel de la Figura 26) pero con una pequeña sección diciendo el motivo de la visita y un botón de *Finalizar visita*. Cuando se pulse este botón se envía una notificación al dispositivo del paciente para que cierre la llamada automáticamente y se cambia el estado de la notificación a end, tal como se puede ver en la Figura 32.

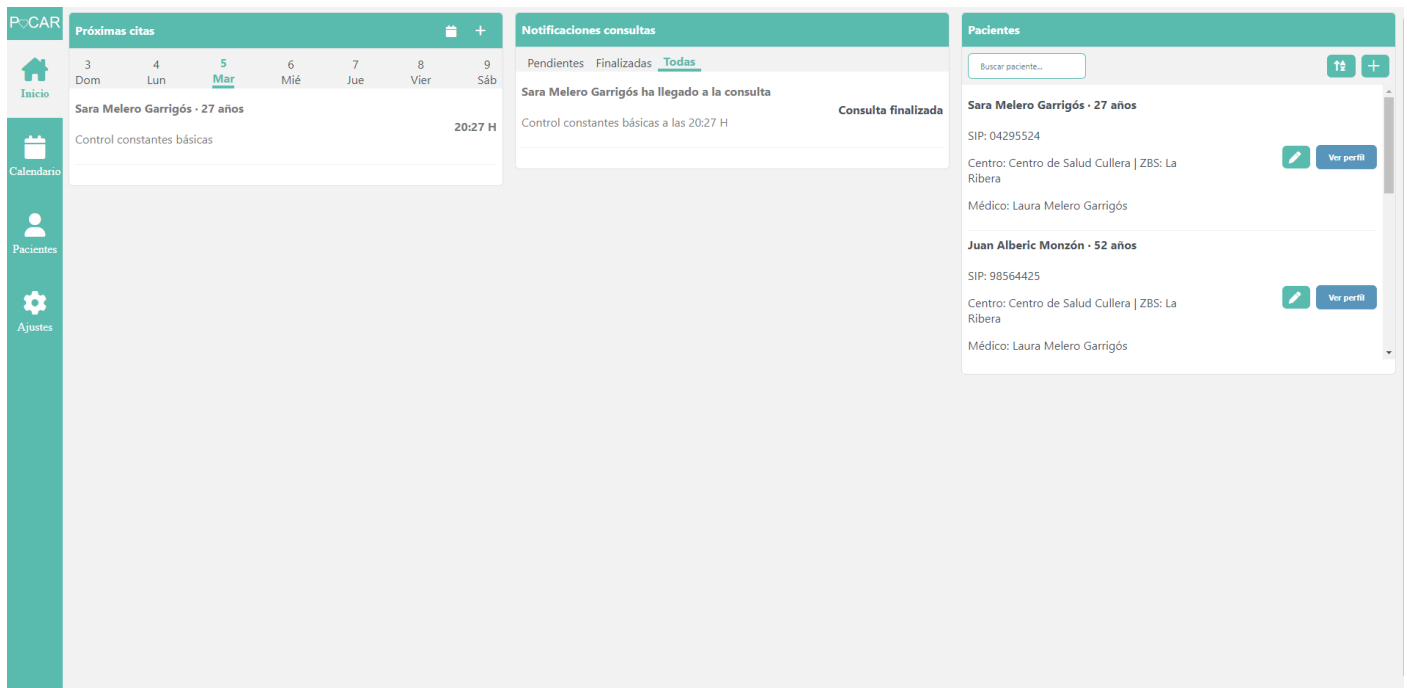


Figura 32. Resultado final inicio médicos (consulta finalizada)

En la Figura 31 también se ha incluido el panel de Históricos y de Próximas citas (presentes también en la página de pacientes). Hay que destacar que la organización de la página no es exactamente igual a la de los wireframes o mockups (tiene los mismos paneles pero ordenados de forma distinta) ya que se identificó falta de espacio que impedía ver correctamente los gráficos o eventos.

Dos de los paneles más importantes son los de *Tomar nueva medida* e *Histórico medidas*. Cuando se selecciona un tipo de medida se pone una descripción por defecto que explica el funcionamiento de dicho dispositivo, aunque el

médico lo puede cambiar si lo prefiere. Al pulsar en Enviar instrucción se envía un mensaje tanto a la aplicación del paciente como al coordinador y se espera una respuesta. Cuando llega una respuesta, si es válida, la añade al histórico de medidas tal como se observa en la Figura 33 y la . Se observa que la medida es añadida de manera automática en el panel de históricos. En caso de que una medida no se tome correctamente (el coordinador no es capaz de conectarse por cualquier motivo al dispositivo) o pasa 1 minuto y medio sin respuesta, se notifica para que el médico pueda elegir si reintentar el proceso tal como se puede ver en la Figura 35. Si se pulsa en *Recalcular*, el sistema enviara una nueva notificación para volver a intentar obtener el resultado.

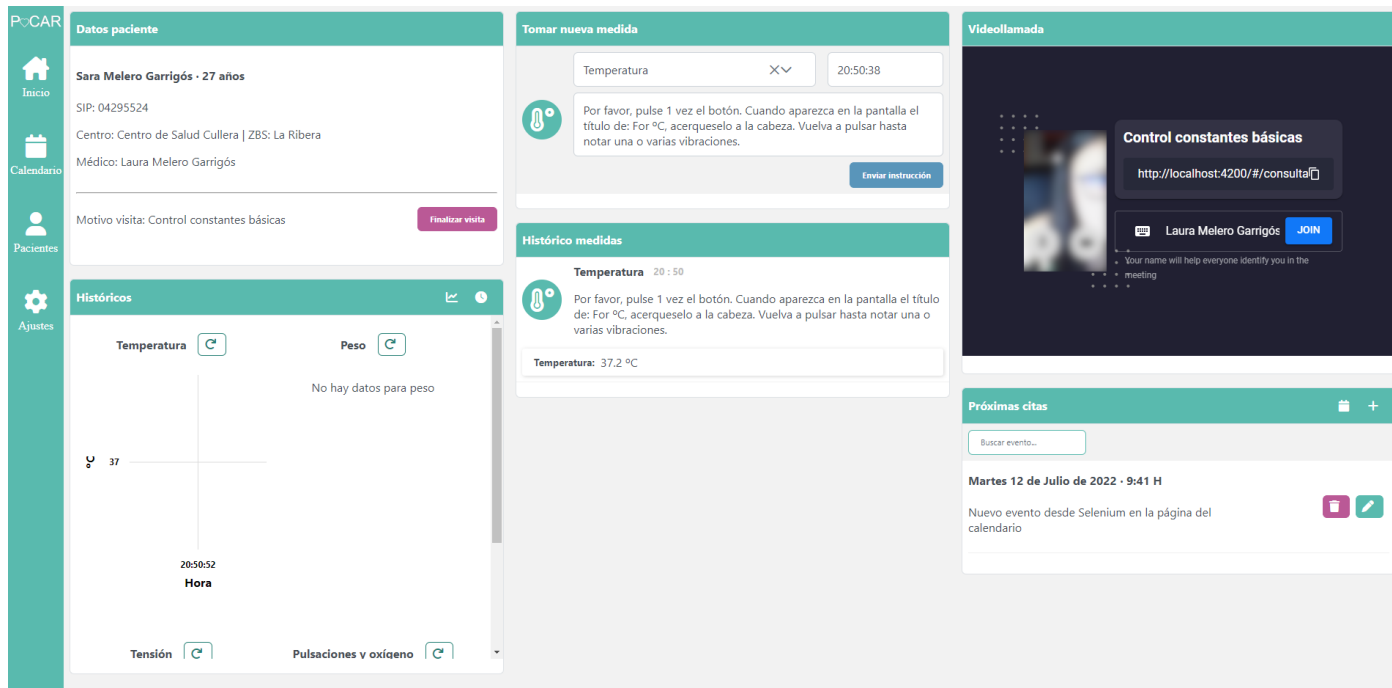


Figura 33. Resultado final consulta médicos (toma medición correcta)

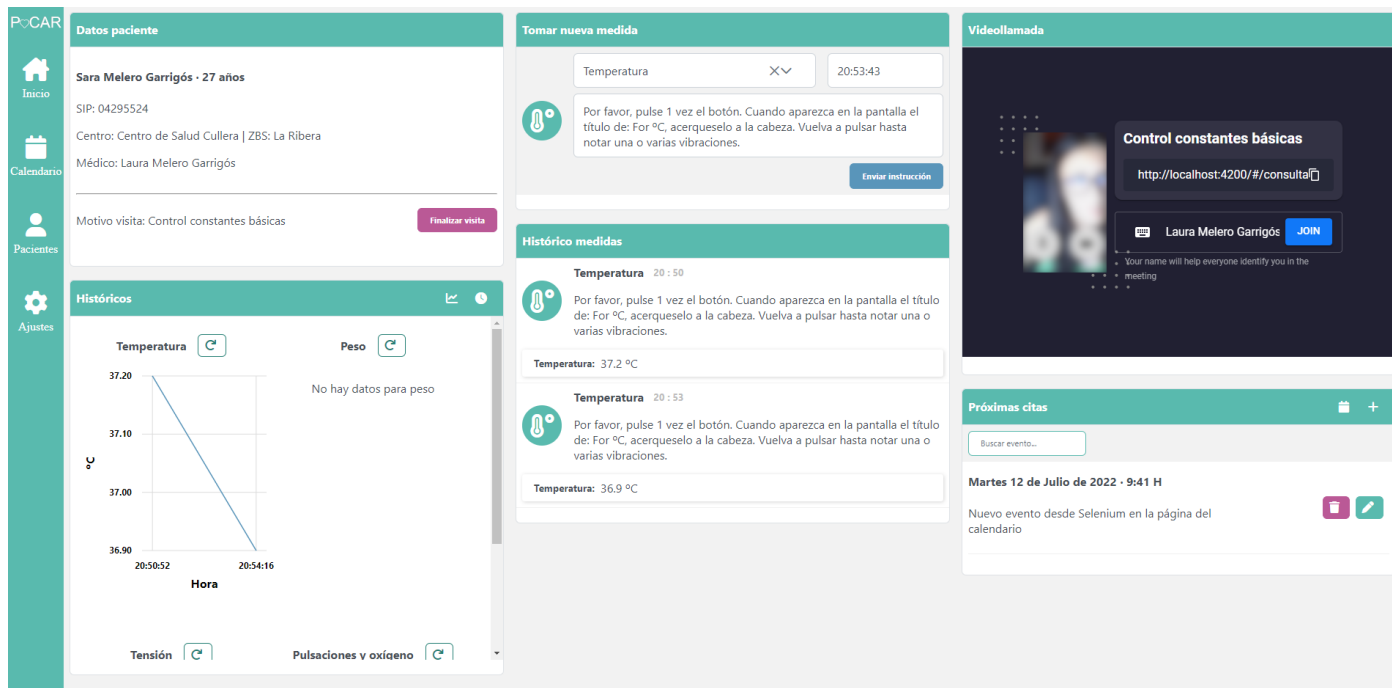


Figura 34. Resultado final consulta médicos (toma medición correcta 2)

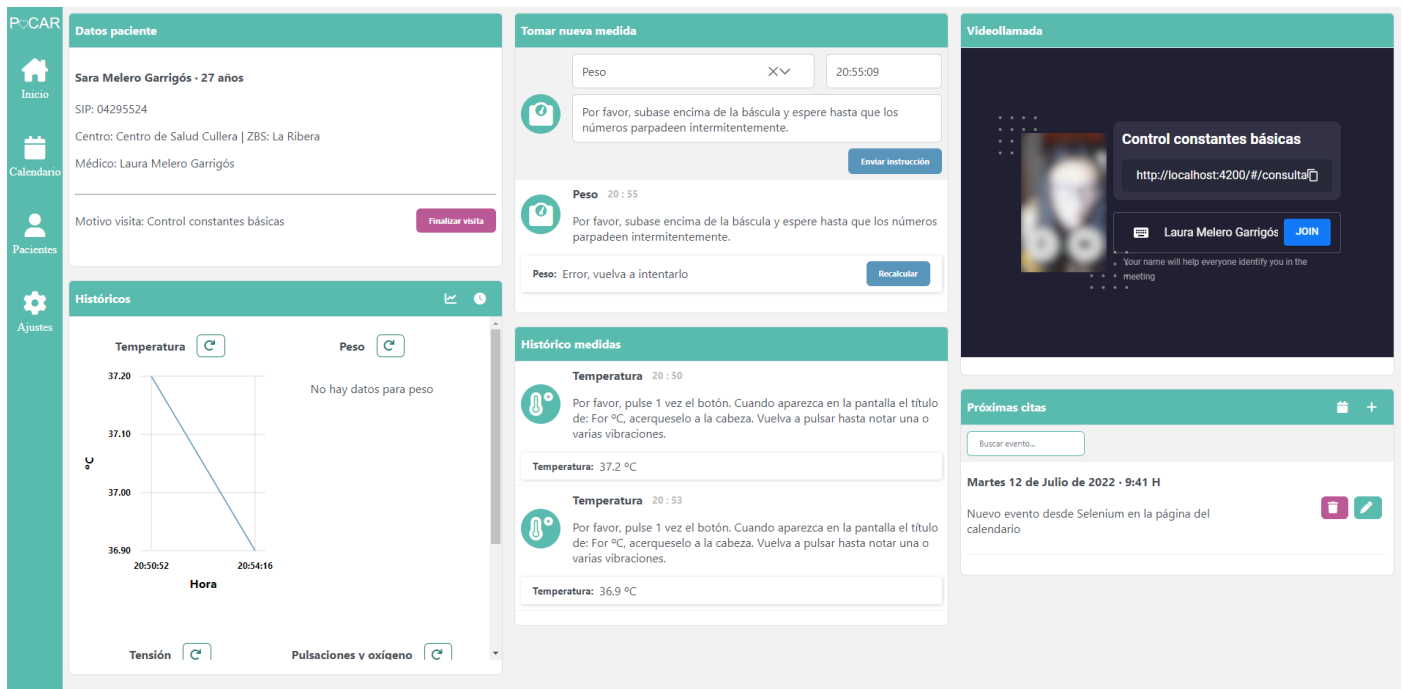


Figura 35. Resultado final consulta médicos (toma medición errónea)

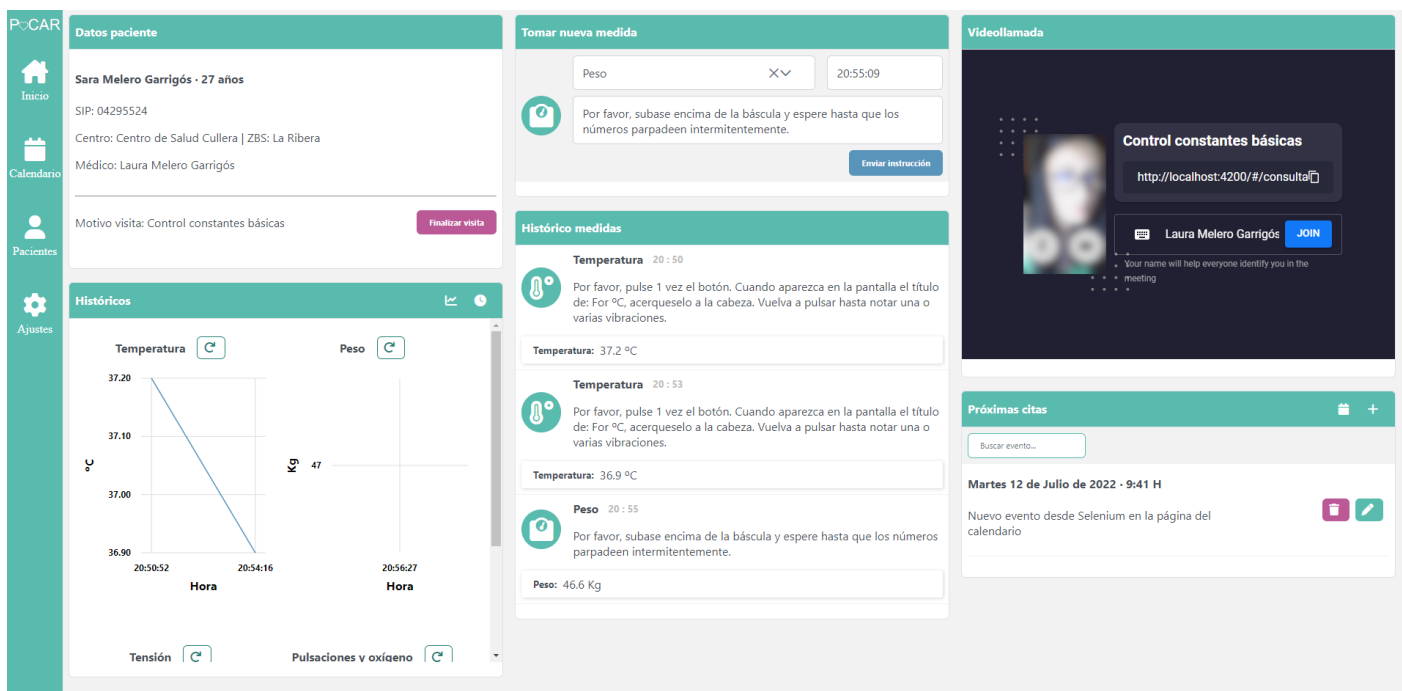


Figura 36. Resultado final consulta médicos (reintento toma medición)

Por último, se puede visualizar el panel de la videollamada, este es uno de los más importantes ya que era un requisito indispensable por parte de los médicos. Tal como se puede ver en las figuras anteriores, como motivo de llamada va a aparecer el motivo de la visita (en este ejemplo, Control constantes básicas). También aparece por defecto el nombre y apellidos del médico (al igual que en el caso del paciente). A diferencia de la aplicación del paciente, tal como se puede observar en la Figura 37, el médico tiene control total sobre la videollamada, él puede controlar tanto el inicio y cierre, como el micrófono o la cámara tanto suya como la del paciente. Esto es importante para evitar que el paciente interaccione con cosas que puedan generar problemas, tal como se ha explicado anteriormente. La videollamada (en los tres puntos) ofrece opciones como la de la pizarra digital, compartir pantalla...

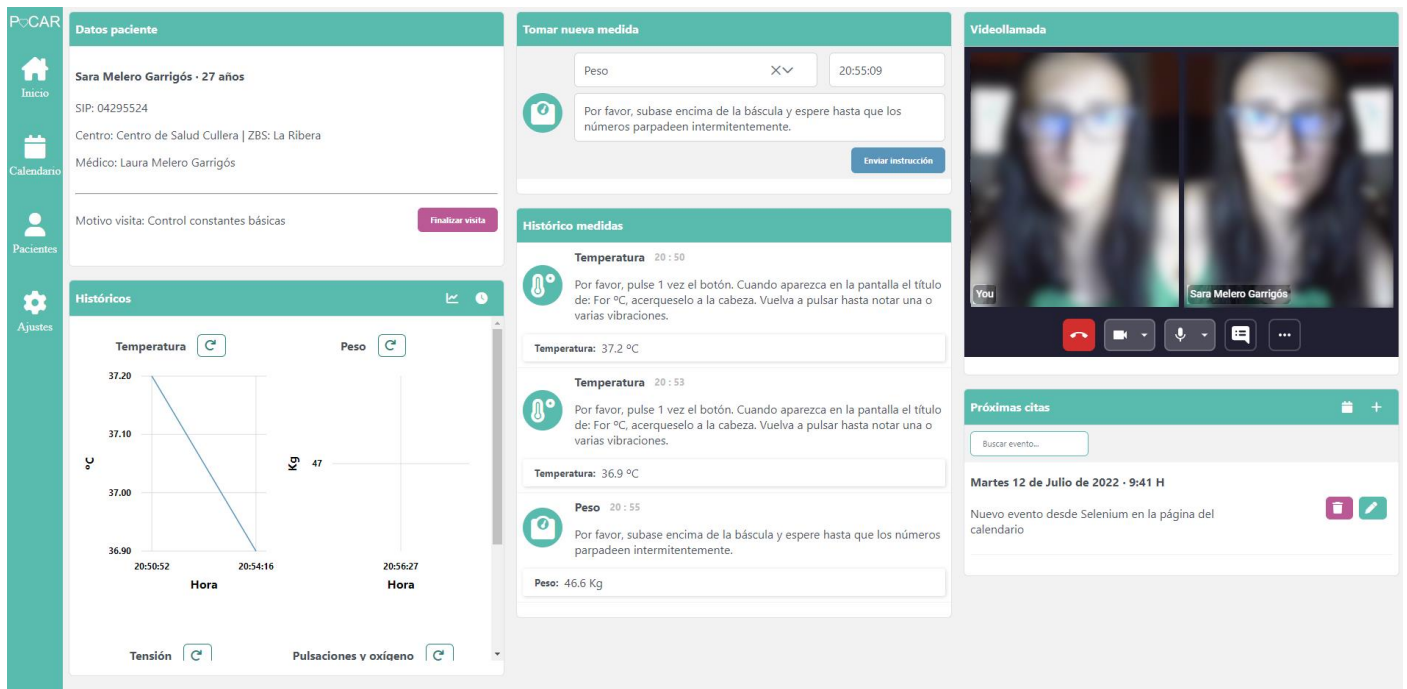


Figura 37. Resultado final consulta médicos (videollamada)

La última sección de la web es la de los ajustes (Figura 38) que permite gestionar todo lo necesario, inclusive los dispositivos. Destacar que todas opciones están habilitadas siempre y en cuanto, el médico tenga permisos de administrador, si no, no le dejará acceder a los datos generales. Esto se puede ver en la Figura 39.

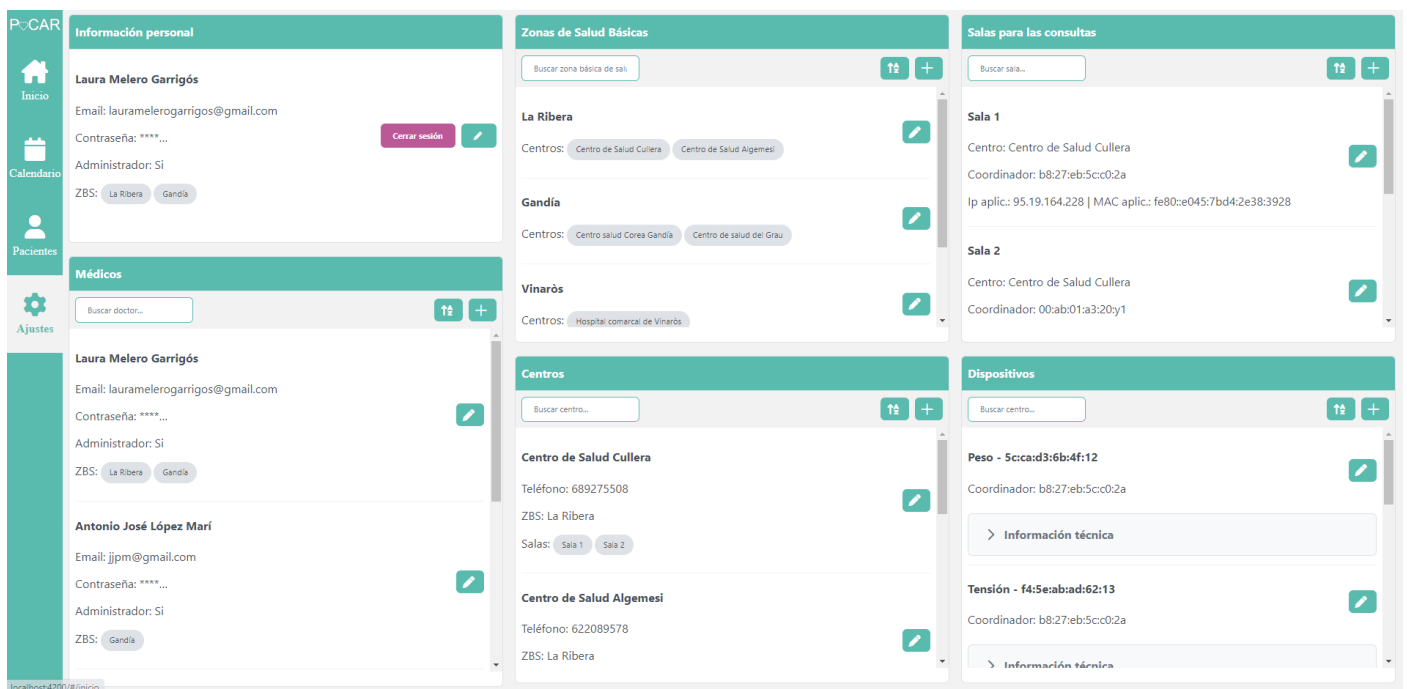


Figura 38. Resultado final ajustes médicos

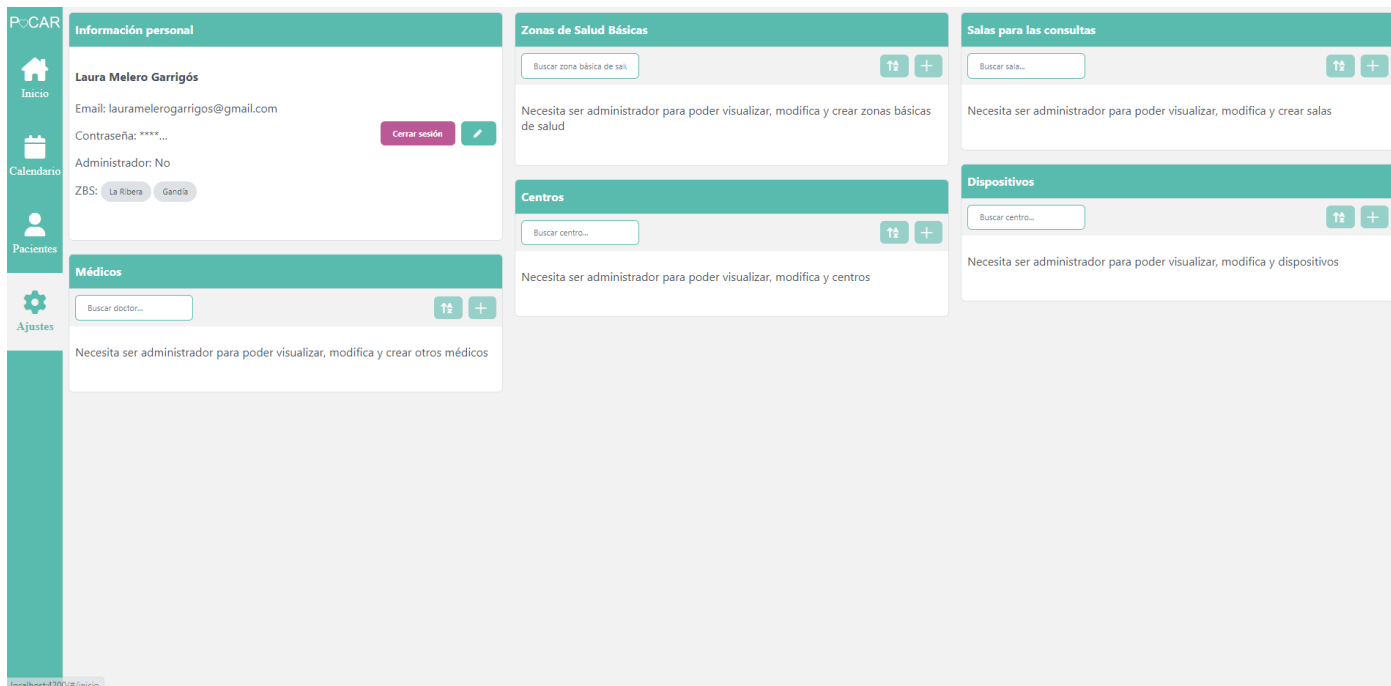


Figura 39. Resultado final ajustes médicos (médico sin permisos de administrador)

En esta página se puede editar un médico. En caso de que no se tengan permisos de administrador no se pueden autoasignar, es decir, que otro usuario con dichos permisos ha de otorgarlos. En este caso, se puede ver que el email no se puede cambiar ya que es lo que se utiliza como identificador.

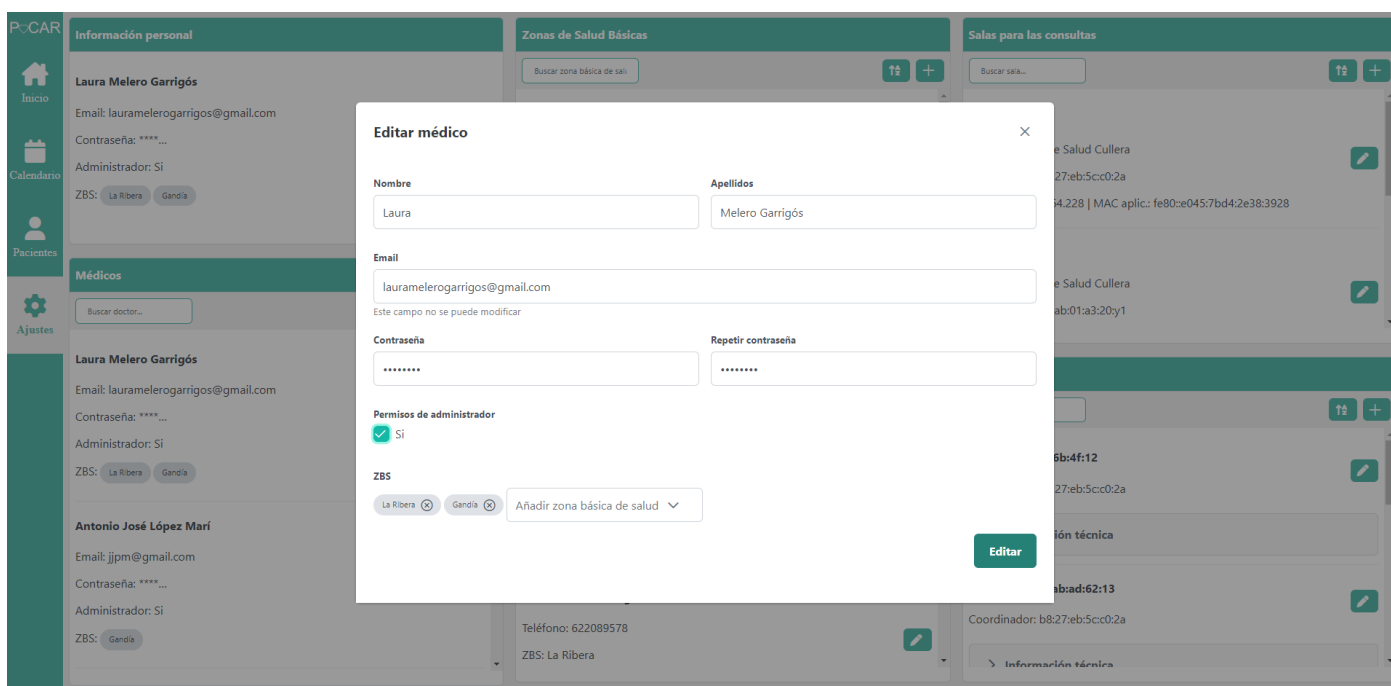


Figura 40. Resultado final ajustes médicos (editar médico)

El proceso de crear un médico (es igual, siempre teniendo en cuenta que el médico que procure hacerlo tenga los permisos de administrador pertinentes.

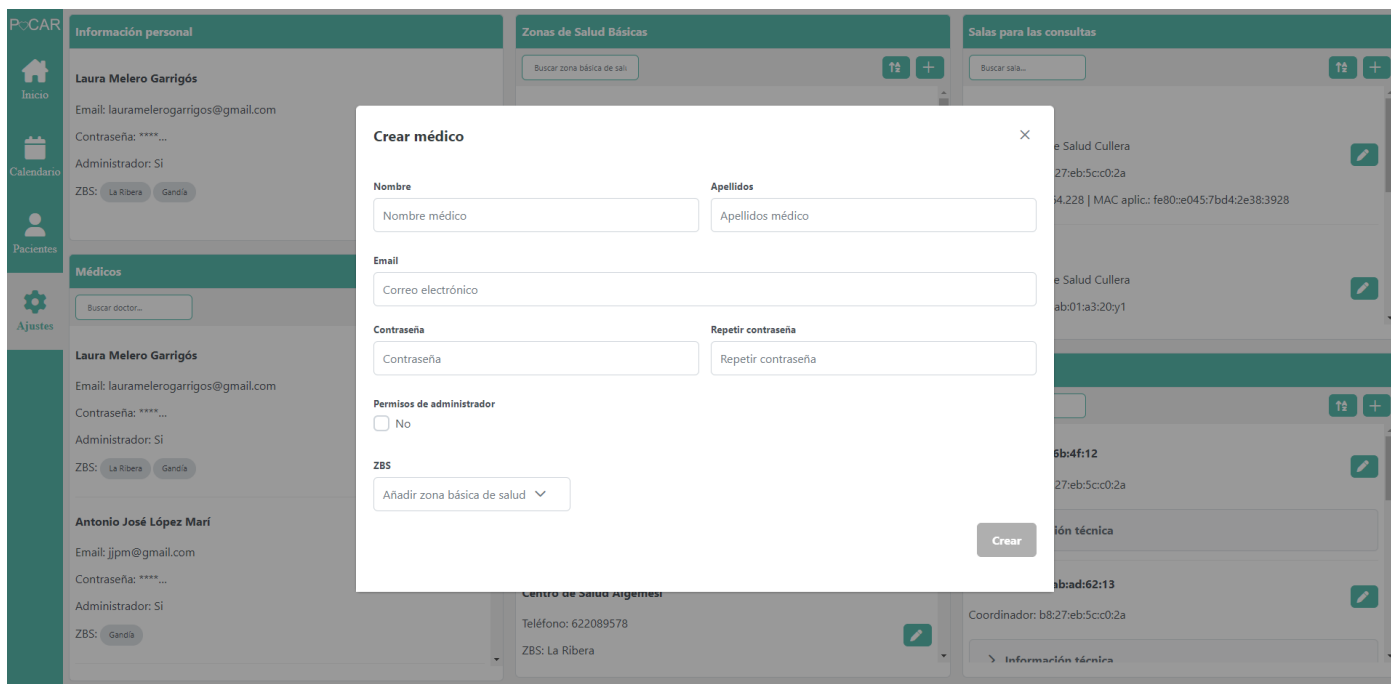


Figura 41. Resultado final ajustes médicos (crear médico)

De igual manera, se pueden editar Figura 42 y crear zonas básicas de salud. En este caso, aunque se puedan ver los centros que pertenecen a dichas zonas, no se pueden editar, se ha de hacer desde el propio panel de *Centros*.

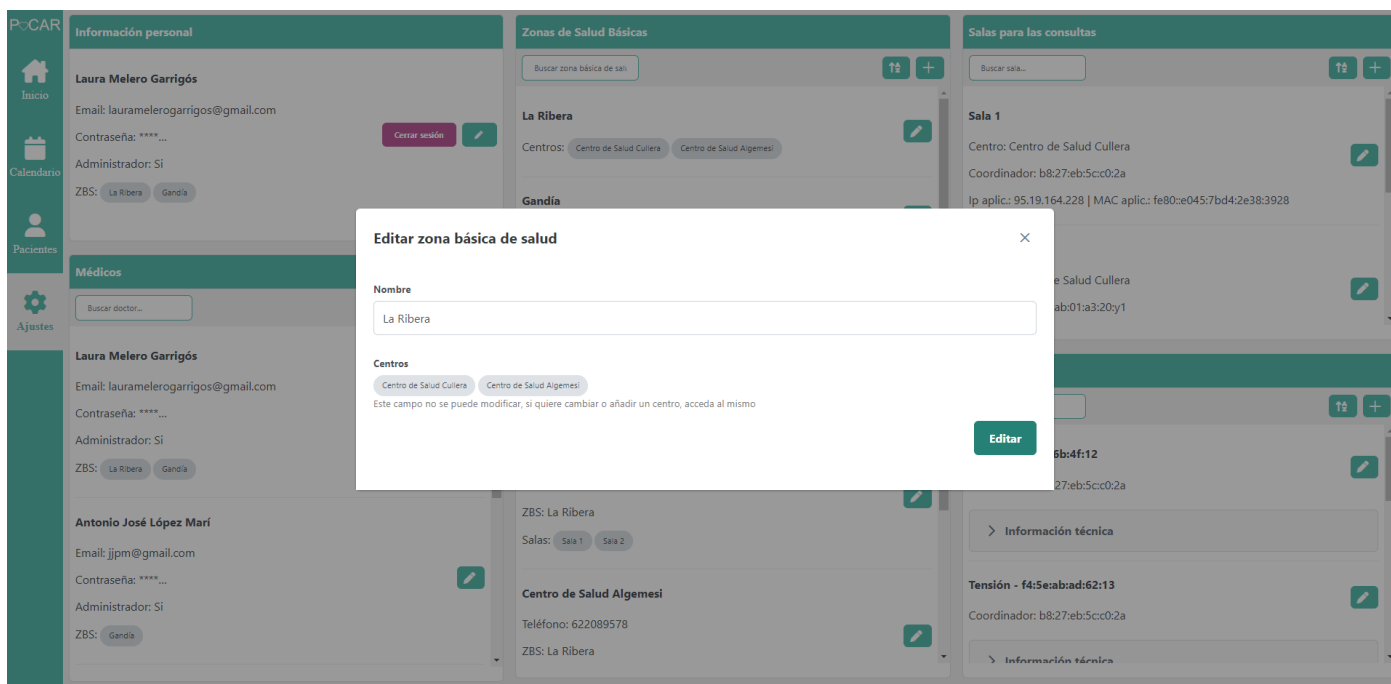


Figura 42. Resultado final ajustes médicos (editar ZBS)

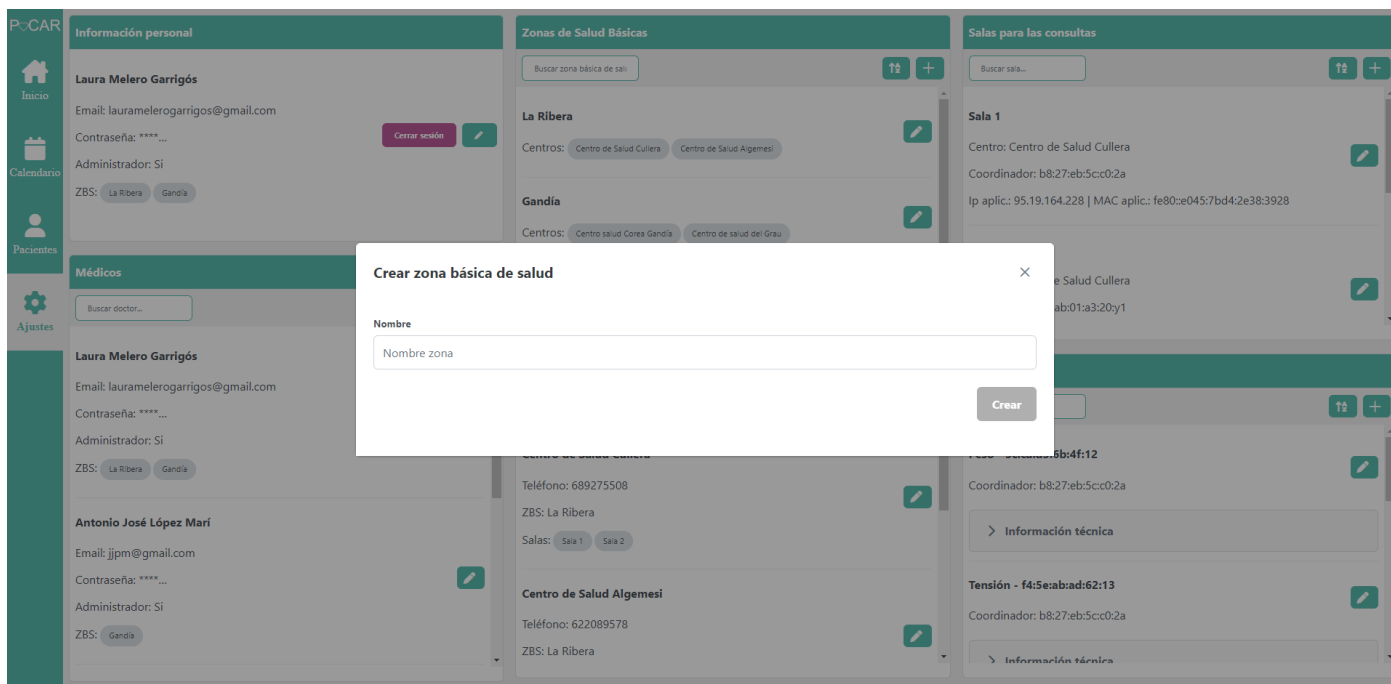


Figura 43. Resultado final ajustes médicos (crear ZBS)

En la Figura 44 y en la Figura 45 se puede ver el proceso de crear y editar un centro que se puede mostrar desde el panel de *Centros*. Cada uno de los centros pertenecerá a una zona sanitaria (al añadirse o crearse un centro nuevo se actualiza también el panel de Zonas de Salud Básicas). Al igual que en el caso anterior las *Salas* se mostrarán como información adicional pero para modificarlas habrá que acceder a cada una de ellas o crear una nueva. El número de teléfono se pone como soporte por si ocurriese algo en el sistema de los pacientes y no hubiese personal en ese momento, saber a dónde dirigirse.

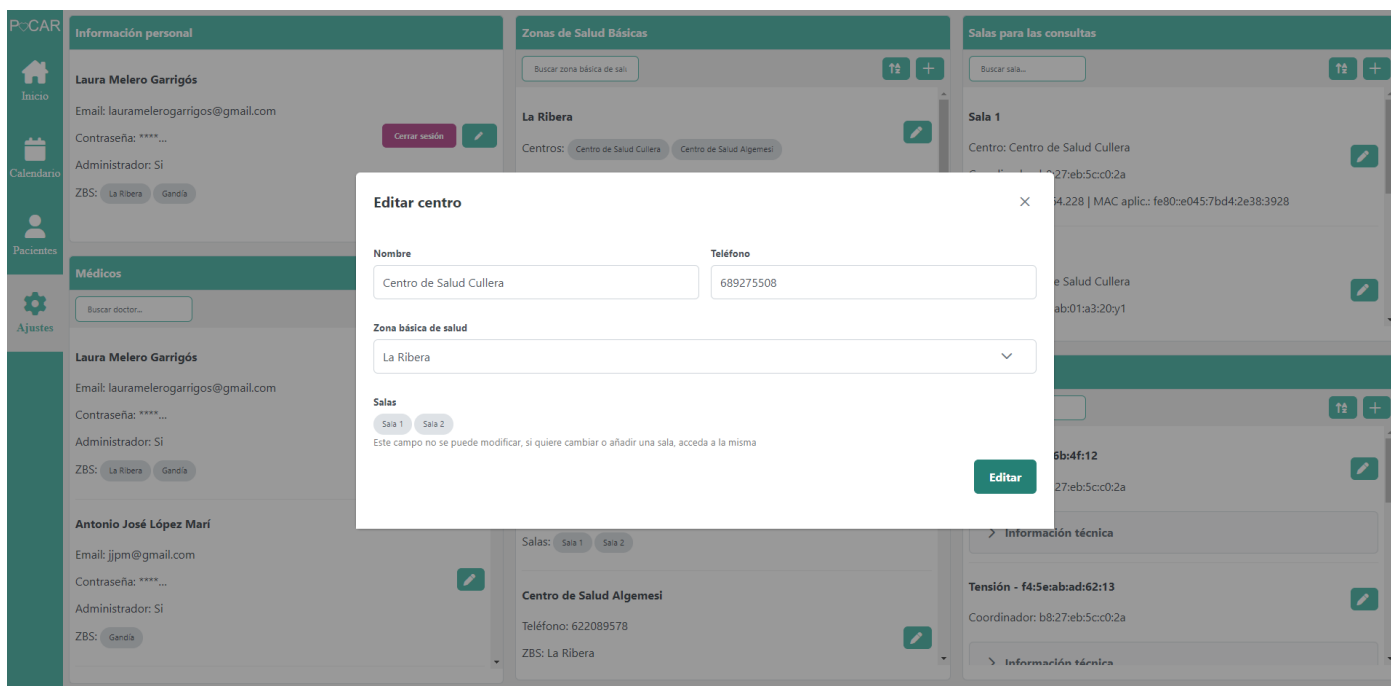


Figura 44. Resultado final ajustes médicos (editar centro)

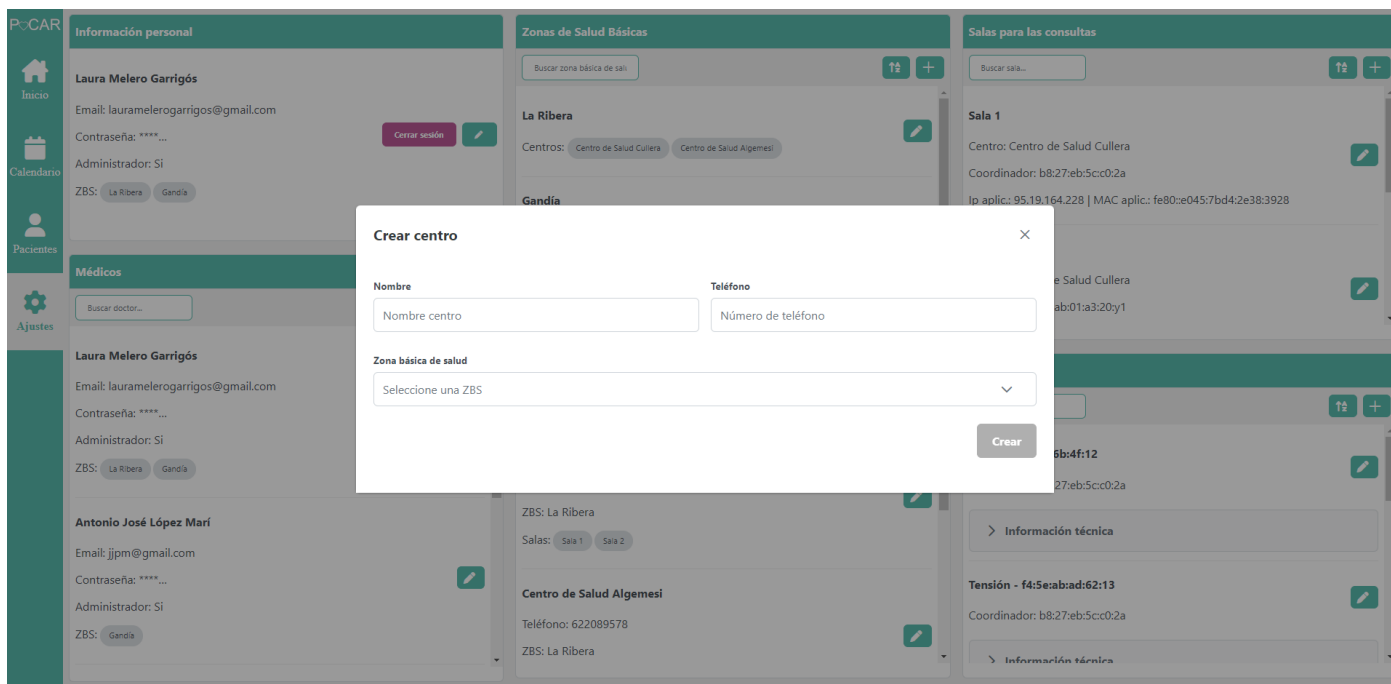


Figura 45. Resultado final ajustes médicos (crear centro)

Gracias al panel de *Salas para las consultas* se pueden crear (Figura 47) y editar salas (Figura 46). La información necesaria para crear una es el centro al que pertenece, la MAC del coordinador (para poderse comunicar con él durante una consulta) así como la IP del dispositivo de que contiene la aplicación del paciente (se recomienda una IP estática, para que no haya problemas) y su respectiva MAC. Destacar que nunca de las MAC ni la IP pueden estar repetidas, cosa que se consulta cada vez que se quiere hacer alguna modificación.

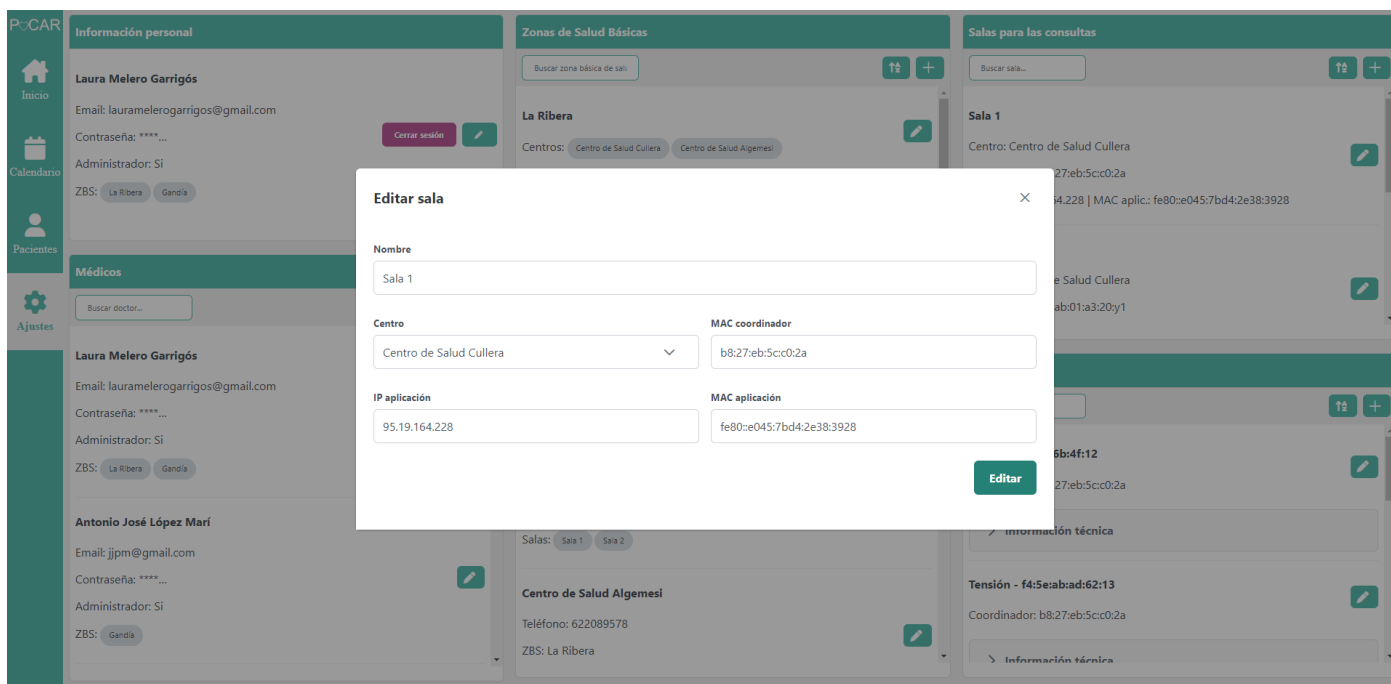


Figura 46. Resultado final ajustes médicos (editar sala)

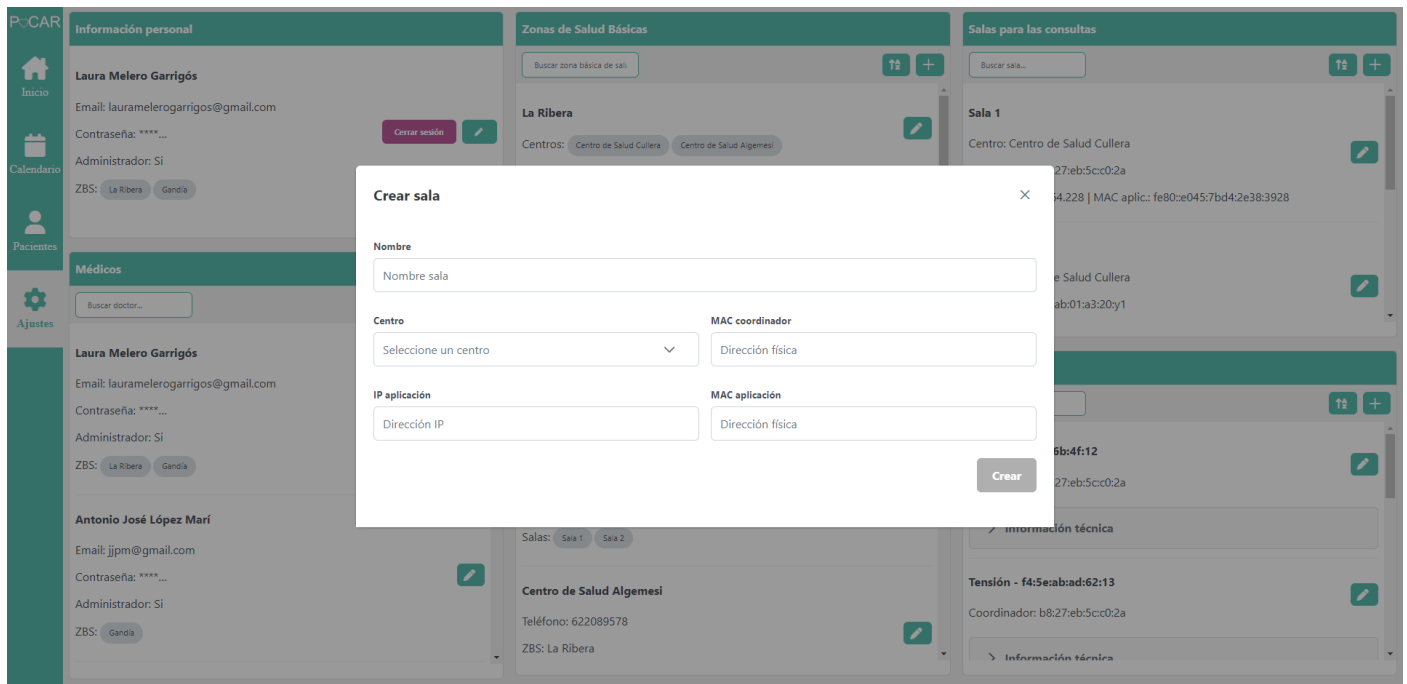


Figura 47. Resultado final ajustes médicos (crear sala)

Por último, el panel de Dispositivos permite no solo visualizarlos (al igual que todos los paneles), sino además crear nuevos (Figura 49) y editarlos (Figura 48). En este caso la información es más delicada, por eso está dentro de un desplegable que pone Información técnica. Al intentar editar o crear un dispositivo, hay unos valores predeterminados en función del tipo de dispositivo que sea y se recomienda no modificar salvo tener conocimientos técnicos. La MAC del coordinador no puede repetirse y tampoco puede haber dos dispositivos del mismo tipo asignados a un mismo coordinador (cosas que se verifican cada vez).

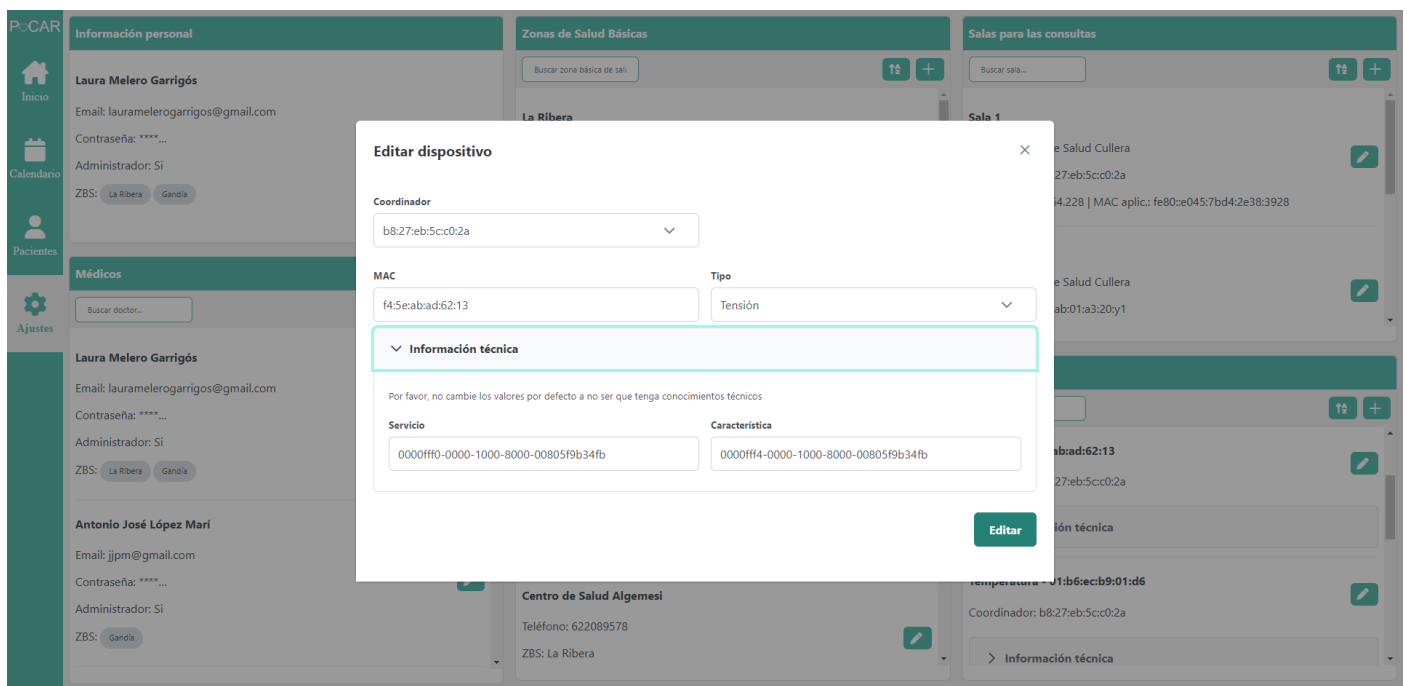


Figura 48. Resultado final ajustes médicos (editar dispositivo)

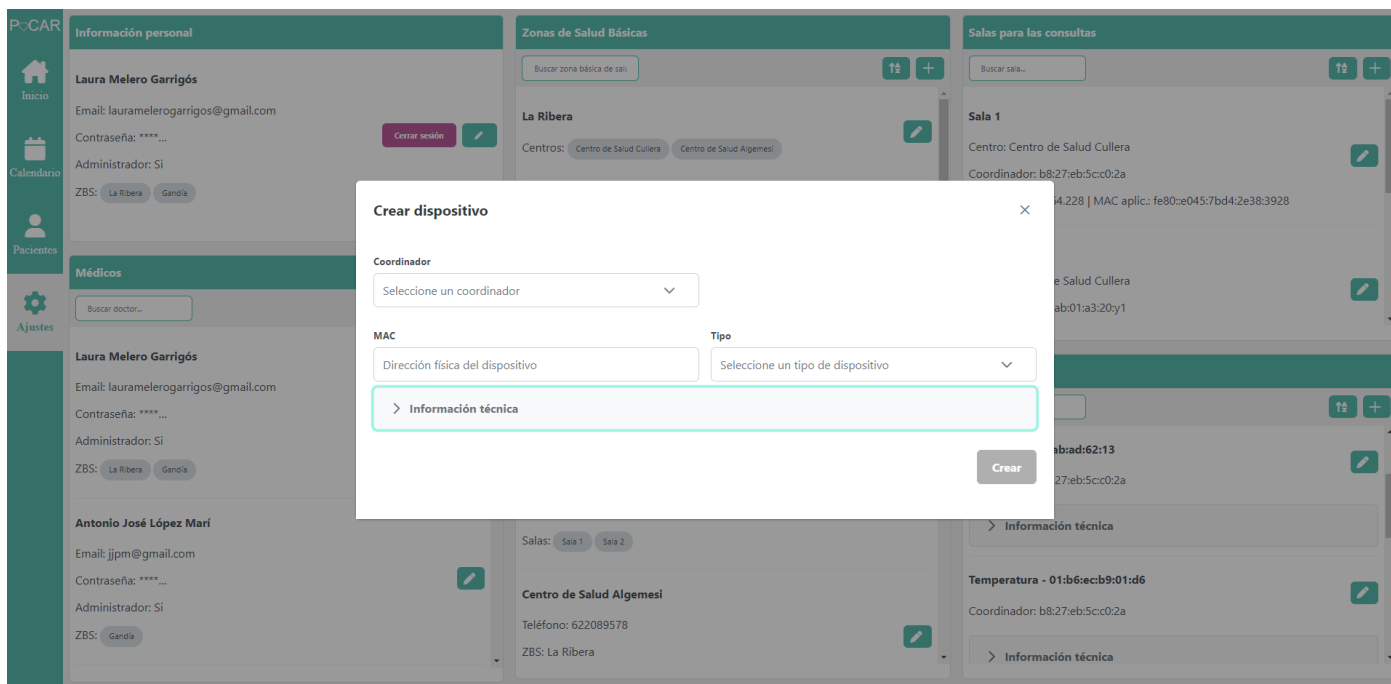


Figura 49. Resultado final ajustes médicos (crear dispositivo)

Destacar, que a pesar de ser una aplicación web pensada para ser utilizada en ordenadores, es completamente responsive. Algunas muestras de ello se pueden ver en las siguientes figuras.

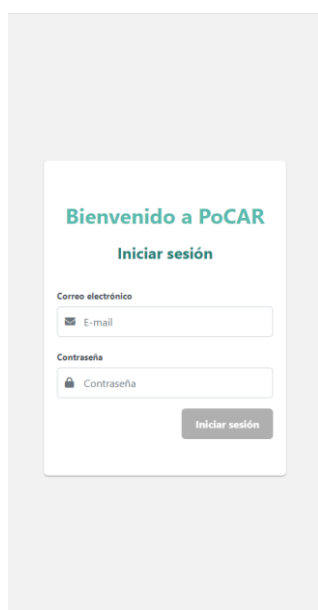


Figura 50. Resultado final móvil inicio sesión médicos

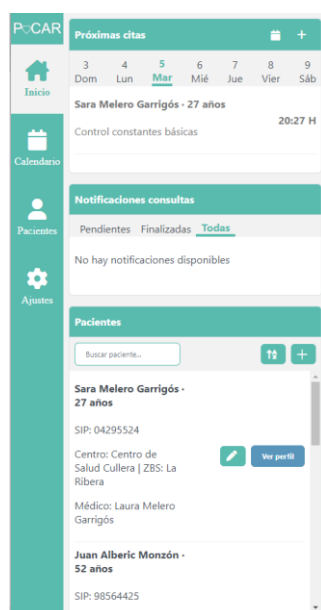


Figura 51. Resultado final móvil inicio médicos

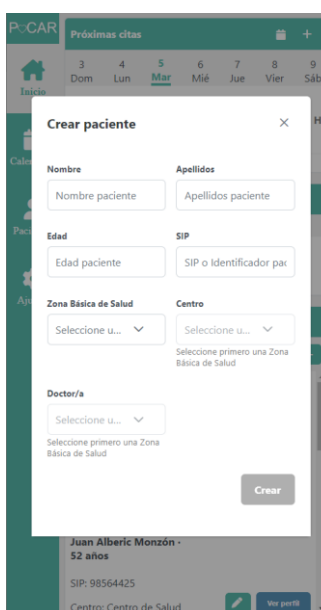


Figura 52. Resultado final móvil inicio médicos (crear paciente)

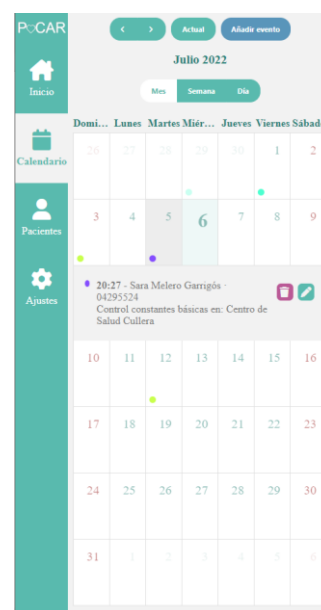


Figura 53. Resultado final móvil calendario médicos

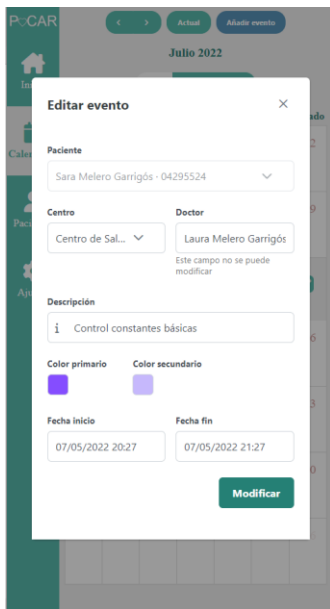


Figura 54. Resultado final móvil calendario médicos (editar evento)

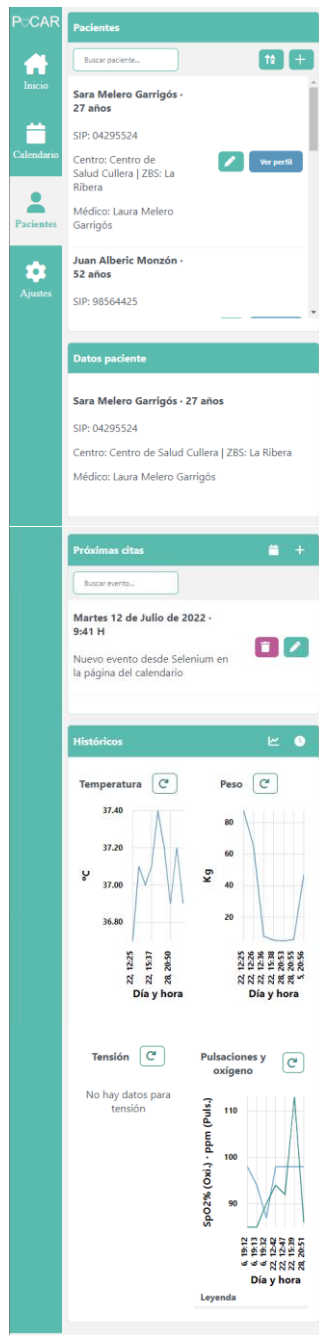


Figura 55. Resultado final móvil pacientes médicos

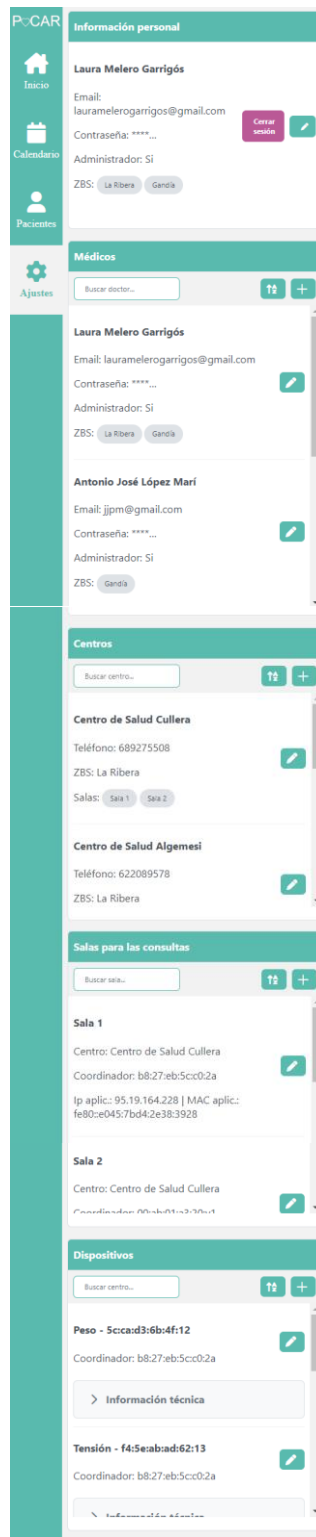


Figura 56. Resultado final móvil ajustes médicos

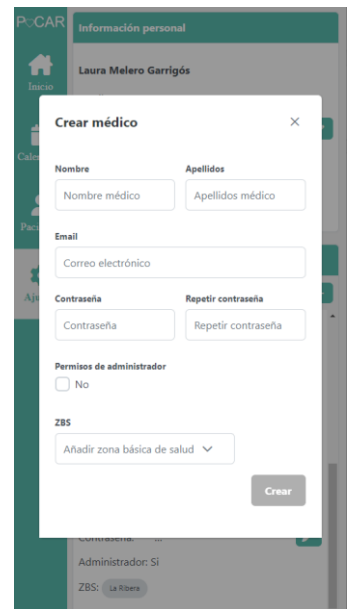


Figura 57. Resultado final móvil ajustes médicos (crear médico)

1.3 PWA pacientes

1.3.1 Wireframes

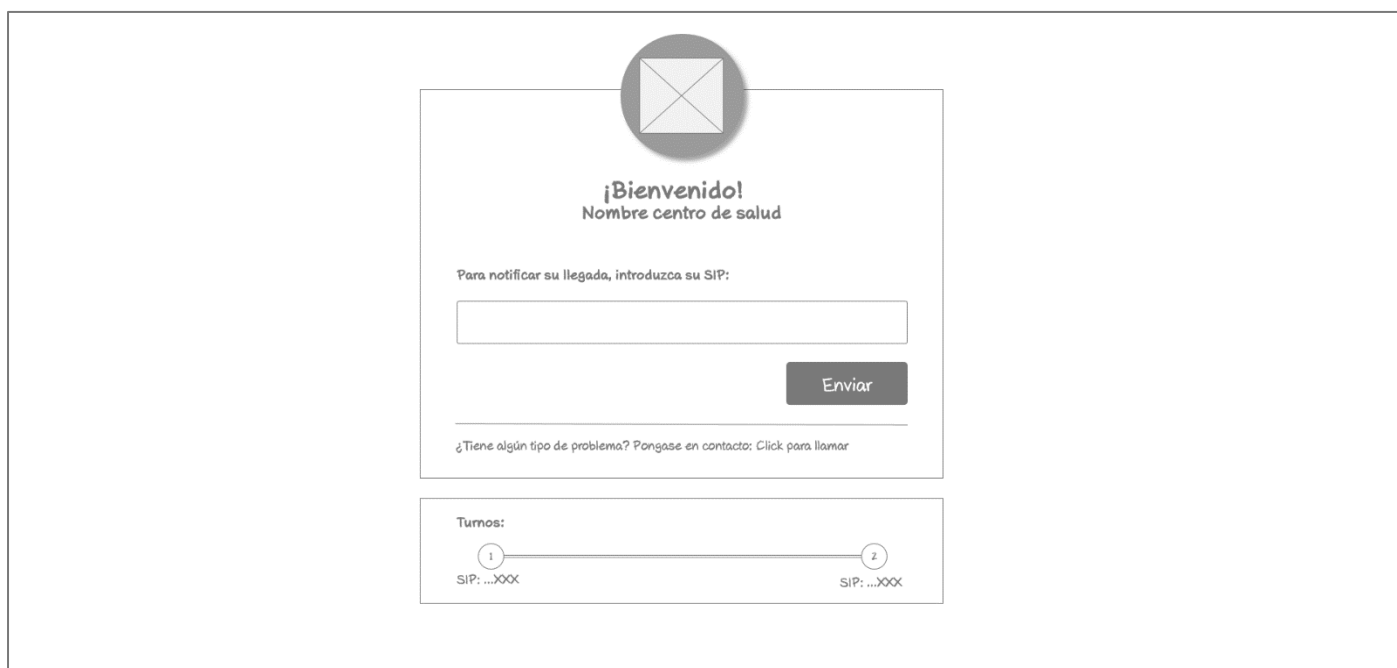


Figura 58. Wireframe inicio sesión pacientes

Tal como muestra la Figura 58, lo primero que verá el paciente será un pequeño login, donde se le pedirá que introduzca su número de SIP. Una vez introducido se enviará una notificación al médico para avisarle de su llegada. En el pequeño panel inferior, se muestra un listado de turnos, ordenado por hora para que cada paciente sepa cuantos otros hay delante de él. Este listado de turnos se almacenará para que aunque se reinicie el sistema o algún caso semejante, no se pierdan los datos.

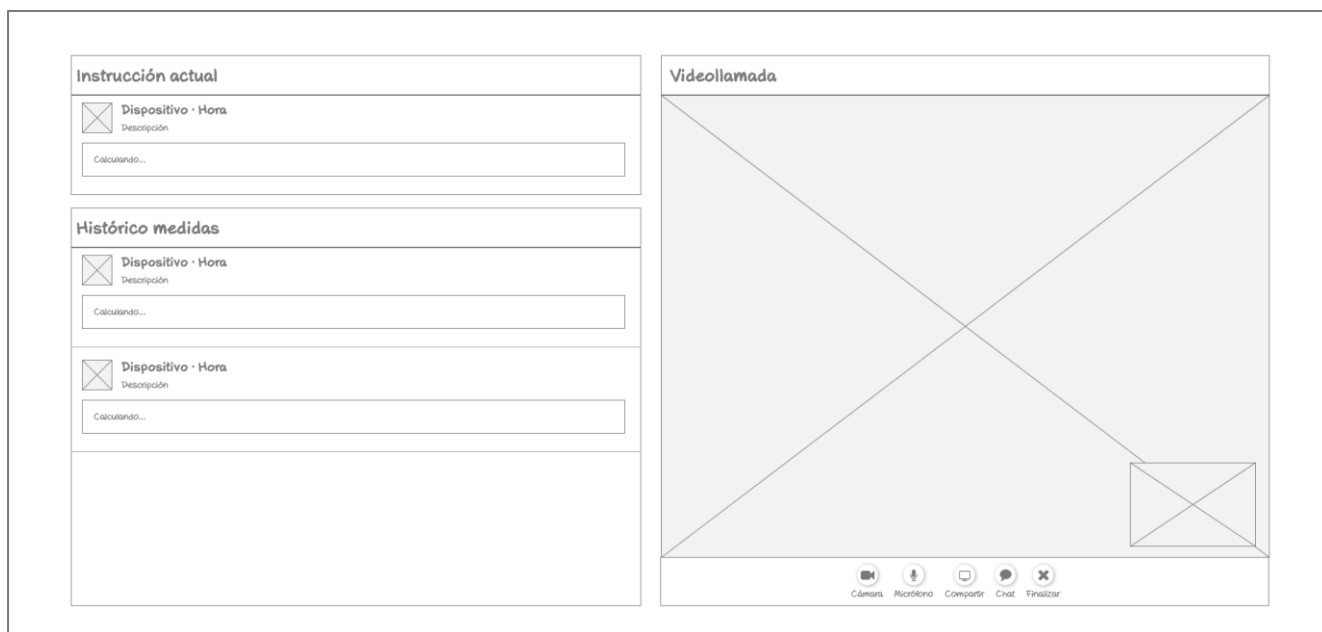


Figura 59. Wireframe consulta pacientes

Cuando un médico decide iniciar la consulta desde su aplicación, la aplicación del médico le redirige directamente a la página de consulta (se puede visualizar en la Figura 59). En esta página se observa que el paciente va a tener las mínimas interacciones posibles, para facilitarle el proceso. Incluso, se pretende bloquear todo lo posible las configuraciones de la videollamada para que por ejemplo "no quiten el volumen sin querer".

1.3.2 Mockups

Se observa que los mockups de la PWA del paciente no distan mucho en cuanto a estética con los del médico; esto es, porque se quiere mantener una misma estética en ambas aplicaciones. La diferencia que si se detecta es que la PWA del paciente tiene muchas menos funcionalidades y por tanto dificultad. Esto es beneficioso pensando sobretodo en un paciente de avanzada edad, con unos conocimientos tecnológicos bajos o con algún tipo de dificultad o discapacidad.

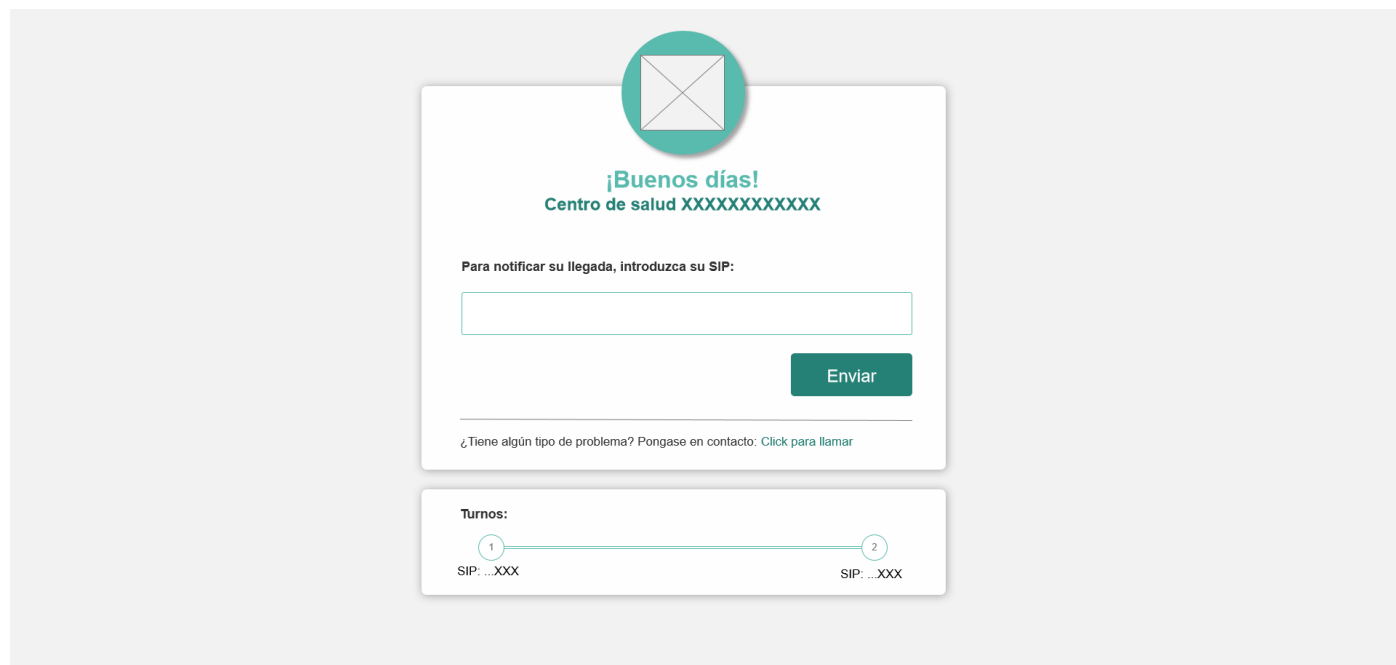


Figura 60. Mockup inicio sesión pacientes

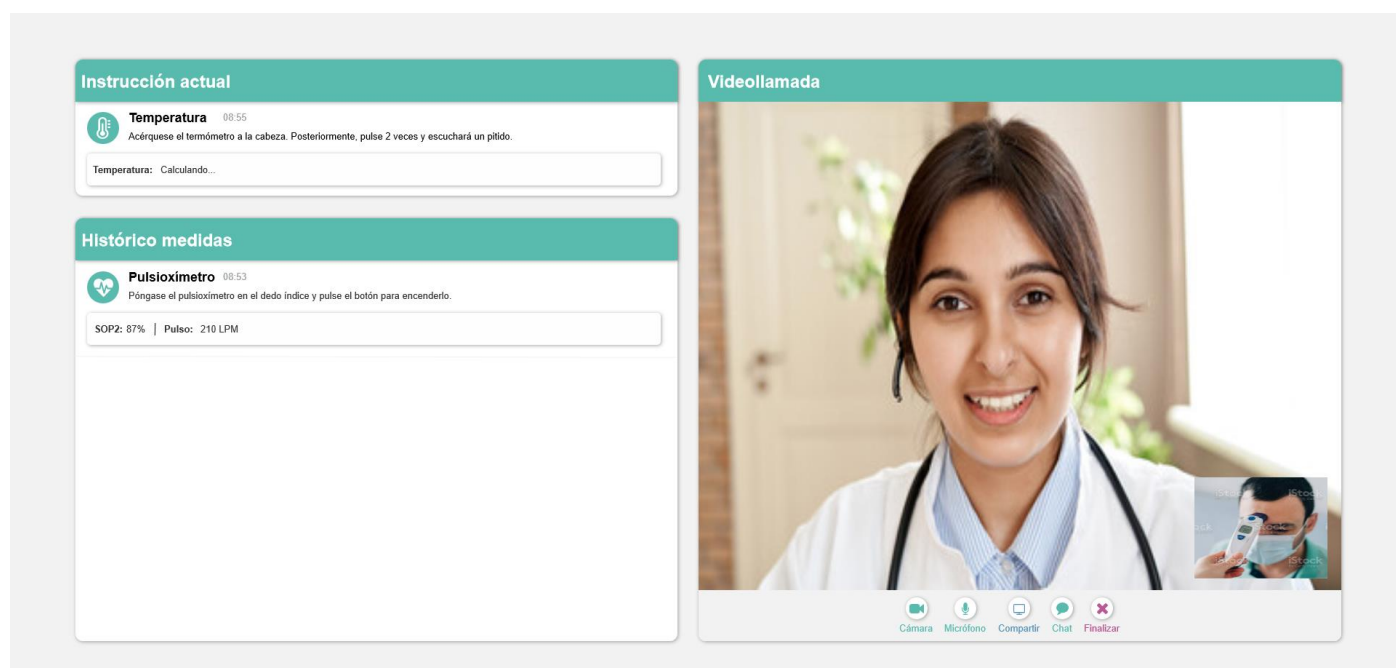


Figura 61. Mockup consulta pacientes

1.3.3 Resultado final

Al llegar un paciente a la consulta (Figura 62), lo primero que deberá hacer es introducir su SIP para indicar al médico que ha llegado a la sala. Una vez esto ocurre, se hacen múltiples comprobaciones como que el SIP sea correcto o que el médico esté conectado a la plataforma (si no, no le llegaría la notificación).

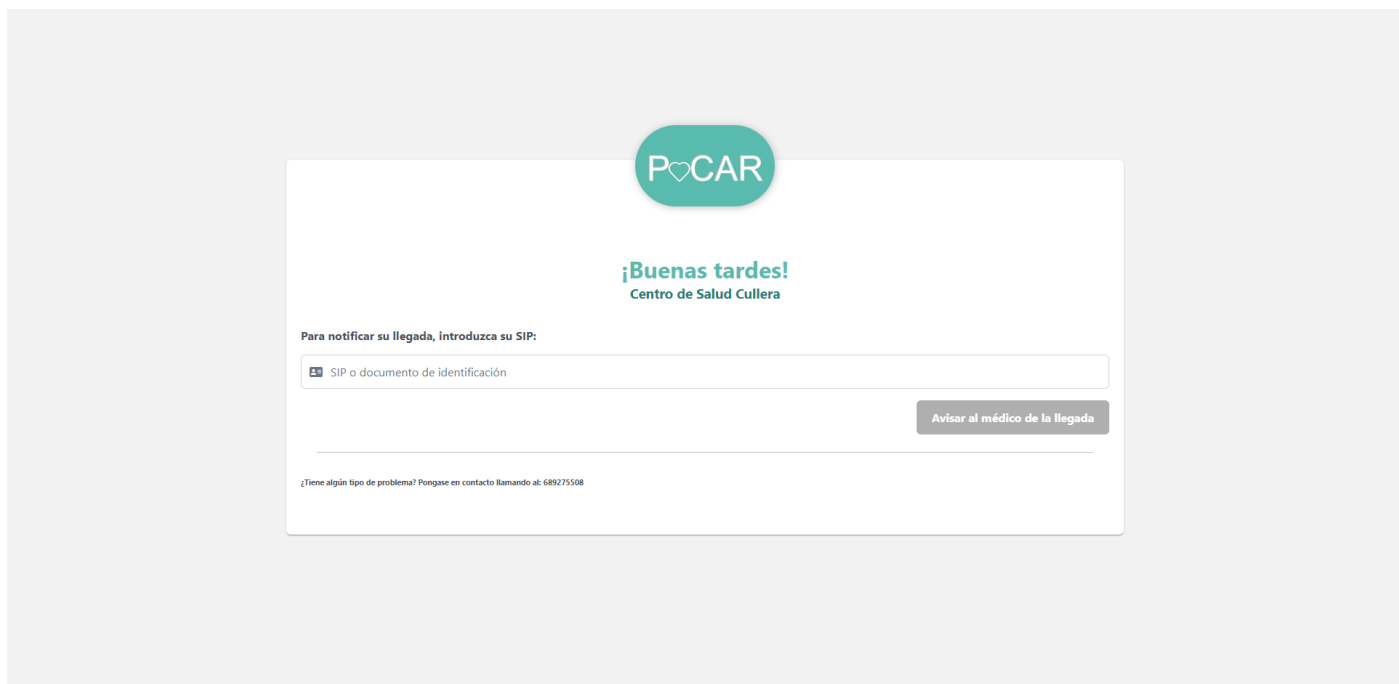


Figura 62. Resultado final inicio sesión pacientes

Una vez el proceso sea correcto se añade al paciente a la lista de espera, tal como se puede ver en la Figura 63. El paciente deberá esperar a que el médico inicie la consulta virtual (al hacerlo, se le redirige directamente a la Figura 64). La idea es hacer que el paciente tenga que hacer la mínima gestión posible para evitarle dificultades con el sistema.

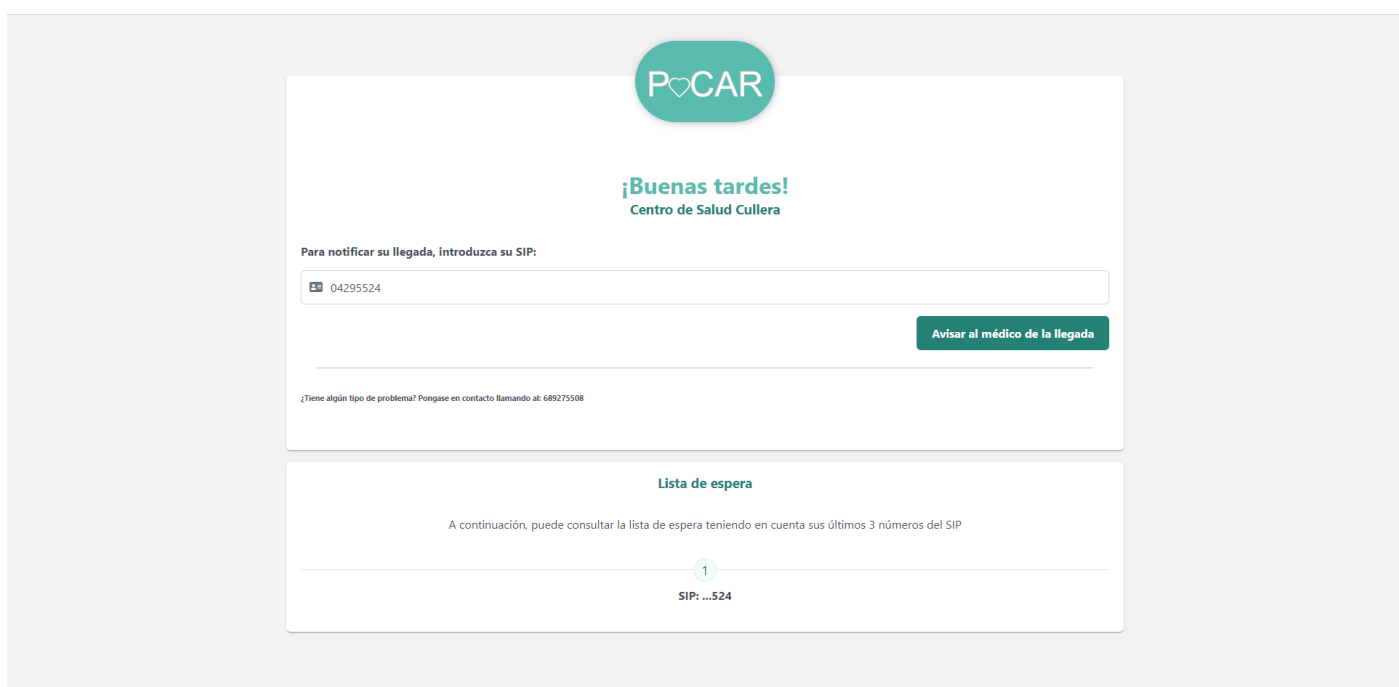


Figura 63. Resultado final inicio sesión pacientes (lista espera)

Si se compara la pantalla de consulta del paciente con la del médico (Figura 31) los principales cambios son que el paciente no tiene prácticamente ningún tipo de interacción (el de iniciar la llamada que era inevitable) pero si quiera puede por ejemplo en el panel de videollamada quitar o subir el volumen o encender y apagar la cámara (cosa que si se veía en la del médico).

También se observa que los paneles son mucho más grandes que los de los médicos, sobre todo para facilitar la usabilidad para personas que puedan tener problemas de vista o cualquier cosa.

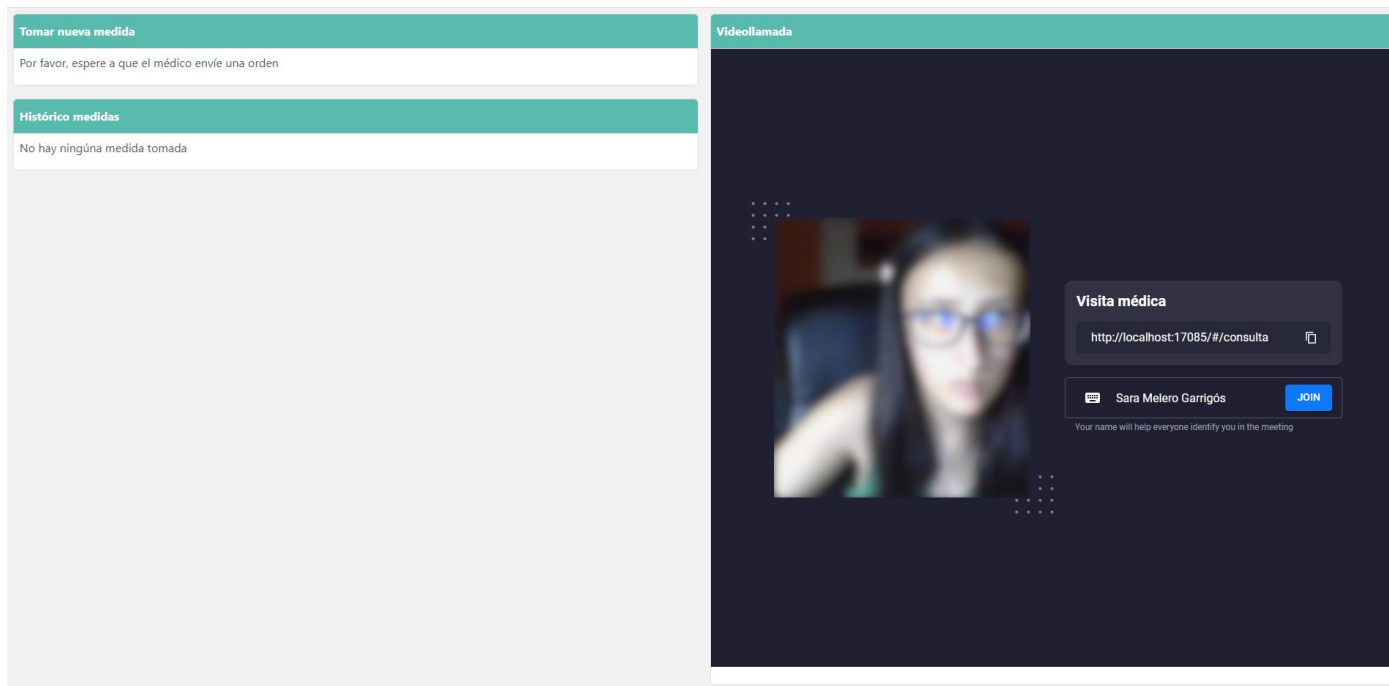


Figura 64. Resultado final consulta pacientes

Cuando el médico envía una petición de obtención de medida, le llega de igual manera al coordinador y al paciente, para que en todo momento tenga una guía de pasos a seguir (sobre todo por si la conexión a internet fuera inestable como para realizar una videollamada).

A continuación, se puede ver el proceso realizado de forma simultánea al del médico. La Figura 65 equivale a la Figura 33, la Figura 66 a la Figura 34, la Figura 67 a la Figura 35 y la Figura 68 a la Figura 36. En todo momento, se avisa al paciente de que hacer, cuando se obtiene una medida correcta se añade al *Histórico de medidas*.

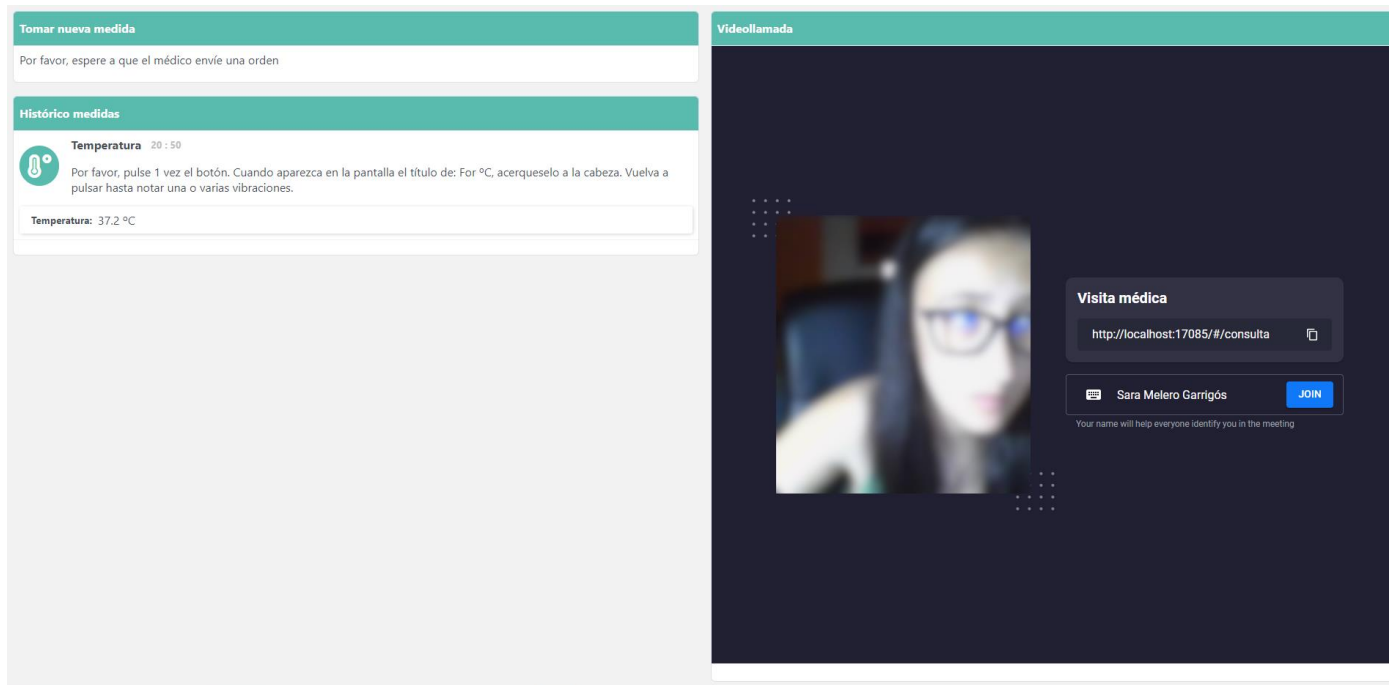


Figura 65. Resultado final consulta pacientes (toma medición correcta)

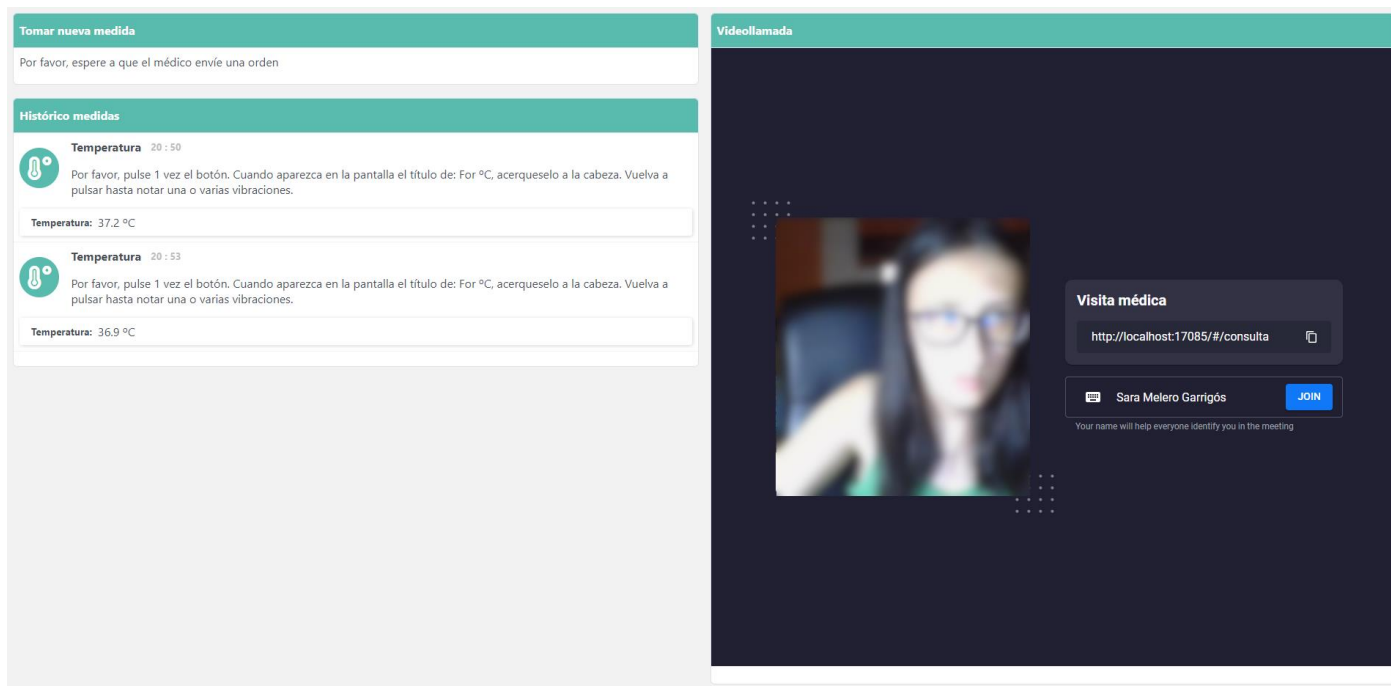


Figura 66. Resultado final consulta pacientes (toma medición correcta 2)

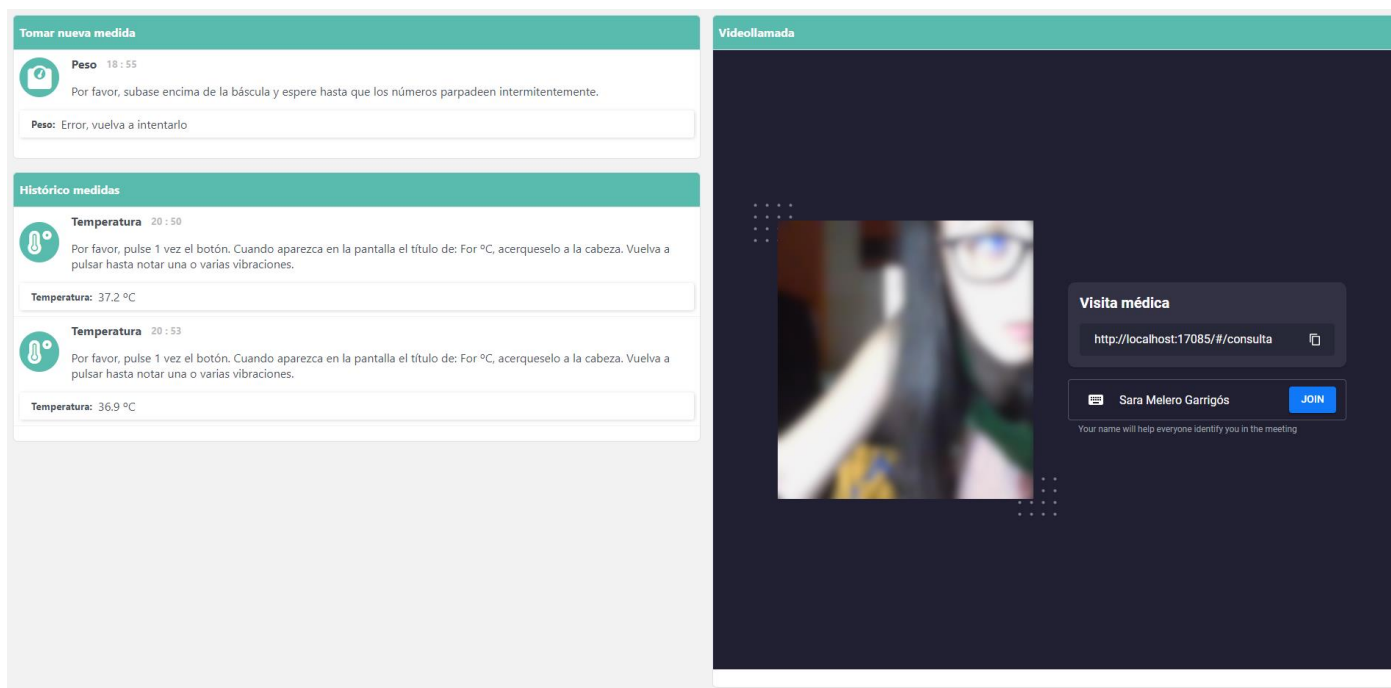


Figura 67. Resultado final consulta pacientes (toma medición errónea)

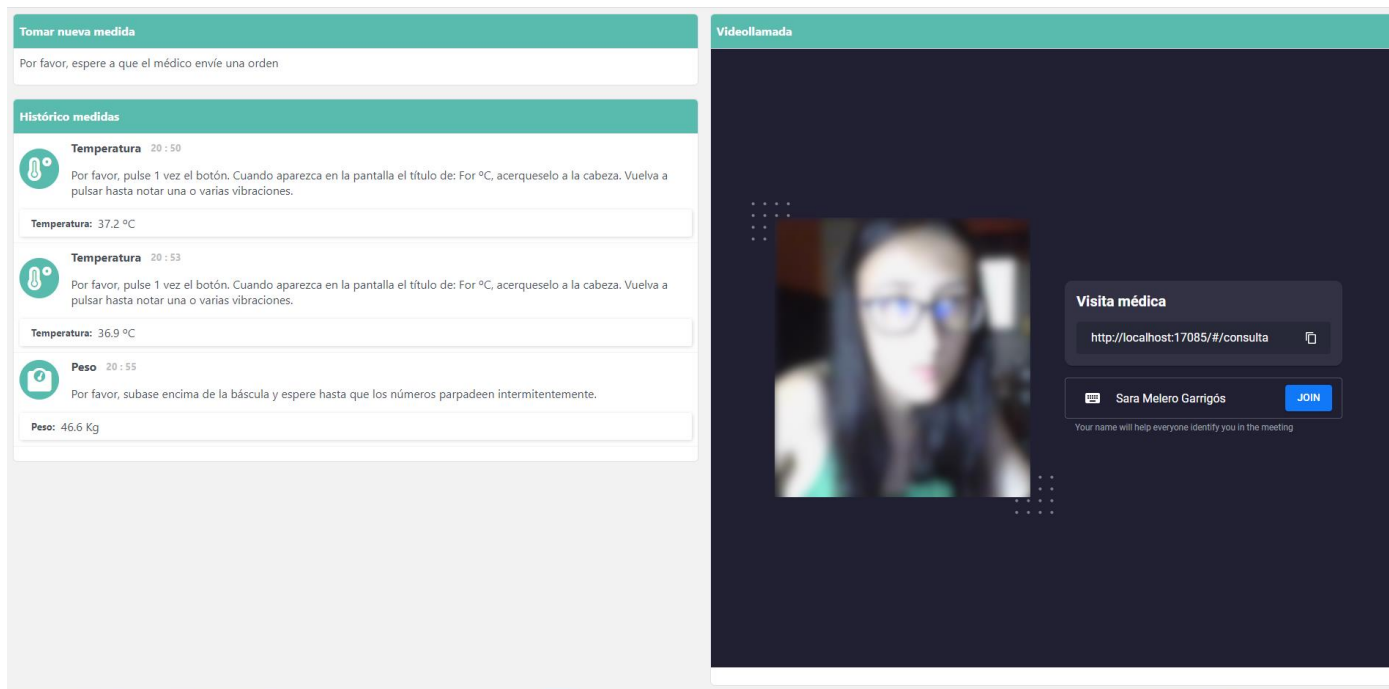


Figura 68. Resultado final consulta pacientes (reintento toma medición)

En el caso de la videollamada, tal como se puede ver hasta ahora y en la Figura 69 el paciente no puede salir de la llamada, ni cambiar la cámara ni nada semejante. En caso de necesitar hacer algún ajuste, lo podrá hacer el médico de manera remota.

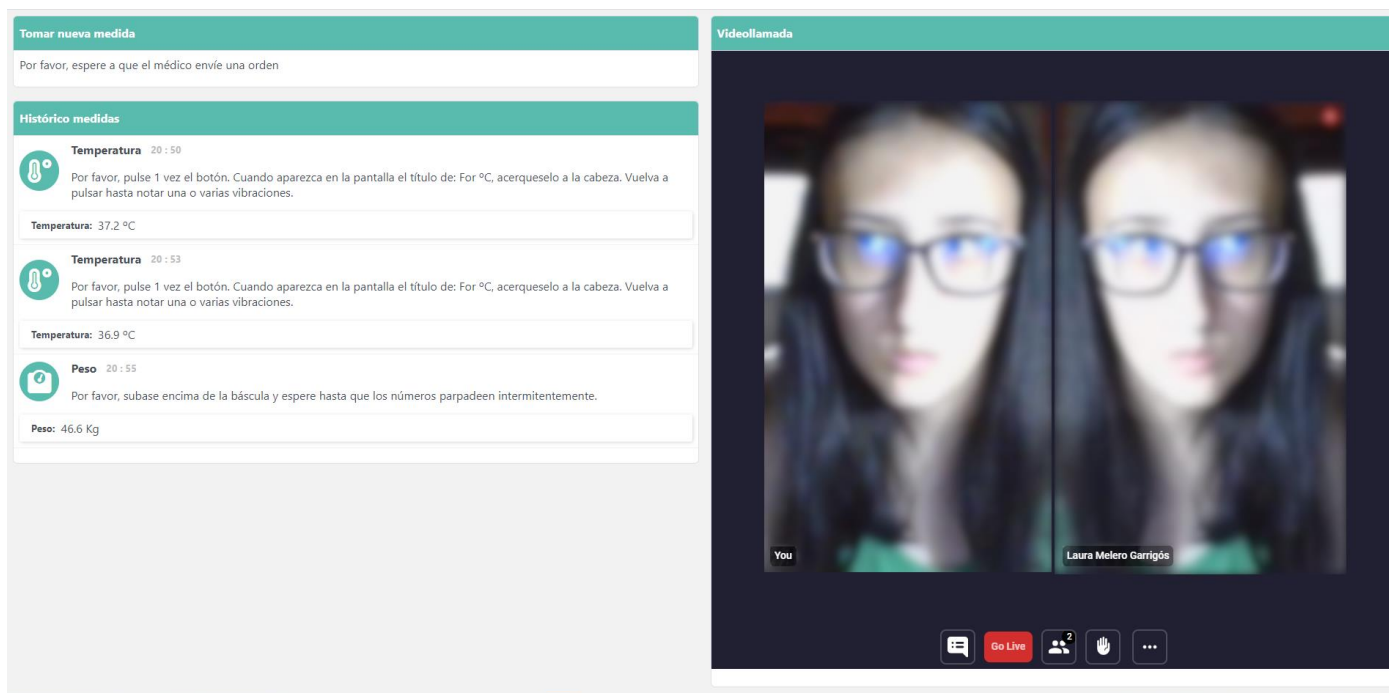


Figura 69. Resultado final consulta pacientes (videollamada)

Destacar, que al igual que la página del médico, esta está preparada para ser responsive, aunque por el bien de la experiencia de usuario no se recomienda utilizar en dispositivos pequeños como un móvil, sino en por ejemplo una tableta (al ser táctil es más simple de usar) en paisado.

2 Test Selenium

2.1 Login

Se hacen 2 tests diferentes, uno que ponga los datos del login incorrectos (email y/o contraseña) y otro correctos. Se identificará en ambos casos el error viendo si se genera un mensaje de error en la web o no. En ambos casos, el testeo ha sido un éxito tal como se puede comprobar en la Figura 70.

```
>>>
==== RESTART: C:\Users\Laura\Desktop\Universitat\TFG\tfg\Tests\testLogin.py ====
La prueba de éxito ha sido un éxito, ha iniciado sesión
.La prueba de error ha sido un éxito, ha dado error
.
-----
Ran 2 tests in 10.636s

OK
```

Figura 70. Resultado testeo Login Selenium Python.

2.2 Eventos

Se hacen pruebas de crear, editar y eliminar eventos desde distintas ubicaciones de la página web, más concretamente las pruebas realizadas son:

- Añadir un evento desde la página de inicio (home).
- Añadir un evento desde la página del calendario (calendar).
- Editar un evento desde el calendario.
- Añadir un evento en un día no concurrente desde la página de inicio y después borrarlo desde el calendario.

Todos estos casos, se resuelven con éxito, tal como se puede observar en la Figura 71.

```
>>>
==== RESTART: C:\Users\Laura\Desktop\Universitat\TFG\tfg\Tests\testEvents.py ===
Se ha creado el evento para ser eliminado con éxito
La prueba ha sido un éxito, se ha borrado correctamente
.La prueba ha sido un éxito, el elemento se ha modificado correctamente
.La prueba ha sido un éxito, se ha creado el evento desde la página del calendario
.La prueba ha sido un éxito, se ha creado el evento desde el home
.
-----
Ran 4 tests in 23.111s

OK
```

Figura 71. Resultado testeo Eventos Selenium Python

Esto también se puede visualizar desde la página web, donde vemos que se ha creado un evento para el día de hoy (03/07/2022) y otro para el día 12 de julio; que se ha modificado la descripción del evento del día 1 de julio y que el evento creado para el día 5 de julio no aparece ya que ha sido borrado con éxito.

3	4	5	6	7	8	9
 9:41 - Juan Alberic Monzón - 98564425 Nuevo evento desde Selenium en la página de inicio en: Centro de Salud Cullera  						

Figura 72. Resultado testeo Eventos Selenium Web 1

10	11	12	13	14	15	16
 9:41 - Sara Melero Garrigós - 04295524 Nuevo evento desde Selenium en la página del calendario en: Centro de Salud Cullera  						

Figura 73. Resultado testeo Eventos Selenium Web 2



26	27	28	29	30	1	2
  19:34 - Sara Melero Garrigós - 04295524 Evento modificado desde Selenium en: Centro de Salud Cullera  						

Figura 74. Resultado testeo Eventos Selenium Web 3

2.3 Pacientes

Se realizan múltiples pruebas desde el home (ya que valen también para la sección de pacientes, ya que así como en el caso de los eventos sí que había discrepancias en función de donde estuviera localizado el componente, en este caso, en ambos lugares, es igual). Las pruebas son:

- Crear un paciente.
- Editar un paciente.

Al intentar crear un paciente se detecta que hay un error ya que el listado de doctores no aparece ni se actualiza al cambiar de ZBS. Al observar el código se detecta que se había comentado la línea de código que realizaba dicha función (posiblemente durante algún proceso de pruebas durante el desarrollo).

Crear paciente

Nombre:

Apellidos:

Edad:

SIP:

Zona Básica de Salud:

Centro:

Doctor/a:

Figura 75. Error Pacientes detectado con Selenium

Crear paciente

Nombre:

Apellidos:

Edad:

SIP:

Zona Básica de Salud:

Centro:

Doctor/a:

Figura 76. Error Pacientes solucionado

Al intentar editar un paciente, se observaba que por defecto el centro no aparecía correctamente en algunos casos. Después de investigarlo, se corrigió ya que era problema de paso de parámetros incorrectos (se enviaba la id del médico para buscar los centros y los doctores en lugar de la ZBS). Tras solventar dichos problemas y volver a realizar las pruebas, en la Figura 77 se observa el correcto resultado.

```
>>>
=== RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testPatients.py ==
La prueba ha sido un éxito, el paciente se ha editado correctamente
.La prueba ha sido un éxito, se ha creado un paciente correctamente
.
-----
Ran 2 tests in 14.422s

OK
```

Figura 77. Resultado testeo Pacientes Selenium Python

El sistema de pruebas modifica únicamente los campos que le habíamos dicho, es decir, los apellidos, la edad, el SIP, la ZBS, el centro y el doctor.

Editar paciente

Nombre:

Apellidos:

Edad:

SIP:

Este campo no se puede modificar

Zona Básica de Salud:

Centro:

Doctor/a:

Figura 78. Previo testeo Pacientes Selenium Web

Editar paciente

Nombre:

Apellidos:

Edad:

SIP:

Este campo no se puede modificar

Zona Básica de Salud:

Centro:

Doctor/a:

Figura 79. Resultado testeo Pacientes Selenium Web 1

En el caso de añadir un paciente, también se observa que el proceso se realiza correctamente.

Selenium Programa Test · 18 años

SIP: 111111111

Centro: Centro de Salud Cullera | ZBS: La Ribera

Médico: Laura Melero Garrigós


 [Ver perfil](#)

Figura 80. Resultado testeo Pacientes Selenium Web 2

2.4 Médicos

En este caso, se van a desarrollar las pruebas en dos archivos diferentes, ya que si se quiere comprobar si el cambio de permisos de administrador es correcto (cambia de true a false y luego vuelve a intentar cambiar a true); al quitarse los permisos al administrador, no se podrán realizar el resto de las pruebas ya que los elementos no existen.

2.4.1 Médico actual

Se observa que todas las pruebas se realizan con éxito.

```
>>>
= RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testCurrentDoctor.py
La prueba ha sido un éxito, se ha modificado el estado del doctor correctamente.
Pasa de isAdmin = true a isAdmin = false
La prueba ha sido un éxito, no se ha cambiado el estado del doctor porque no es
administrador
.La prueba ha sido un éxito, se ha cambiado la contraseña correctamente
.La prueba ha sido un éxito, se ha generado un error al ser las 2 contraseñas di
ferentes
.
-----
Ran 3 tests in 18.119s

OK
```

Figura 81. Resultado testeo Médicos Selenium Python 1

El resultado también se puede visualizar a través de la Figura 82 obtenida desde la página web:

Información personal

Laura Melero Garrigós

Email:
lauramelerogarrigos@gmail.com

Contraseña: ****...

Administrador: No

ZBS: La Ribera Gandía


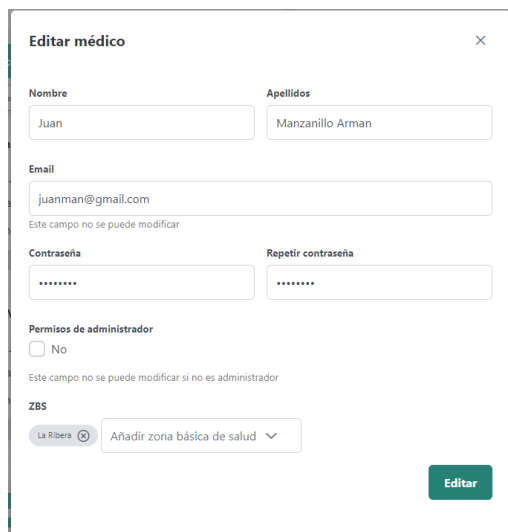
[Cerrar sesión](#) 

Figura 82. Resultado testeo Médicos Selenium Web 1

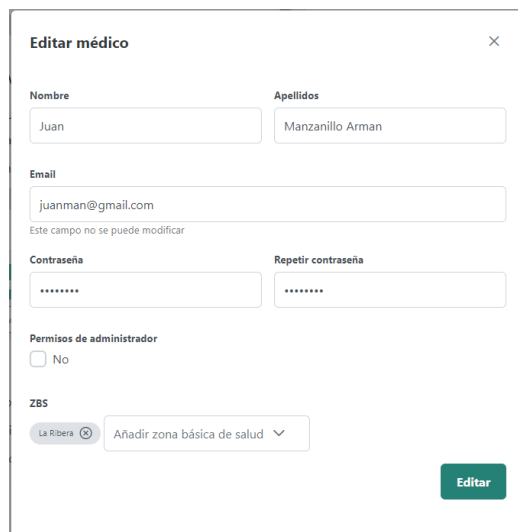
2.4.2 Todos los médicos

Al intentar modificar los permisos de administrador de un médico, se descubre un problema. En la sección de información del médico actual los permisos de administrador se pueden asignar o no de manera correcta, pero en el caso de ser administrador y querer cambiar los permisos de administrador de otro médico no lo permite puesto que se deshabilita el checkbox al no ser administrador ese médico. El problema se solventa comprobando si el médico que tiene la sesión iniciada es quien tiene los permisos y no cada médico en particular.



The screenshot shows a web form titled "Editar médico". It contains fields for "Nombre" (Juan), "Apellidos" (Manzanillo Arman), "Email" (juanman@gmail.com), "Contraseña" (masked with asterisks), and "Repetir contraseña" (masked with asterisks). Below these is a checkbox labeled "Permisos de administrador" which is disabled, with the text "Este campo no se puede modificar si no es administrador" underneath. At the bottom, there is a "ZBS" section with a button "La Ribera" and a dropdown menu "Añadir zona básica de salud". An "Editar" button is at the bottom right.

Figura 83. Error Médicos detectado con Selenium



This screenshot is identical to the previous one, but the "Permisos de administrador" checkbox is now enabled, and the text "Este campo no se puede modificar" is shown below it instead of the previous message.

Figura 84. Error Médicos solucionado

Una vez solucionado el problema, al hacer las pruebas, se observa que los resultados son correctos mediante la Figura 85.

```
>>>
=== RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testDoctors.py ===
La prueba ha sido un éxito, se ha modificado el médico
.La prueba ha sido un éxito, se ha creado el médico
.
-----
Ran 2 tests in 15.166s

OK
```

Figura 85. Resultado testeo Médicos Selenium Python 2

Todos y cada uno de los casos también se pueden verificar a través de la web, donde se observa que los datos se modifican y crean correctamente.

Figura 86. Previo testeo Médicos Selenium Web

Figura 87. Resultado testeo Médicos Selenium Web 2

Dr. Selenium Software Test

Email: selenium@test.com

Contraseña: ****...

Administrador: Si

ZBS: La Ribera



Figura 88. Resultado testeo Médicos Selenium Web 3

2.5 Zonas de salud básicas (ZBS)

Se realizarán dos pruebas básicas para confirmar el correcto funcionamiento de las ZBS. Las pruebas consisten en conseguir crear y editar una ZBS con éxito. Los resultados al lanzar los tests son correctos, tal como se puede ver en la Figura 89.

```
>>>
===== RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testBHZ.py =====
La prueba ha sido un éxito, se ha modificado la zona básica de salud
.La prueba ha sido un éxito, se ha modificado la zona básica de salud
.
-----
Ran 2 tests in 12.141s

OK
```

Figura 89. Resultado testeo ZBS Selenium Python

A través de la Figura 90 también se observa el éxito de las pruebas.



Vinaròs	
Centros: Centro de Salud del Grau	
<hr/>	
La Plana	
Centros: No hay centros en la ZBS	

Figura 90. Resultado testeo ZBS Selenium Web

2.6 Centros

De igual manera que en el caso de las ZBS, realizaremos dos testeos unitarios (uno para editar y otro para crear) un centro. Tal como se muestra en la Figura 91, los resultados en ambos casos son favorables, es decir, no se genera ningún error.

```
>>>
=== RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testCenters.py ===
La prueba ha sido un éxito, se ha modificado el centro de forma correcta
.La prueba ha sido un éxito, se ha creado el centro de forma correcta
.
-----
Ran 2 tests in 14.788s

OK
```

Figura 91. Resultado testeo Centros Selenium Python

Desde la página web, también se puede observar que los resultados han salido tal y como se esperaba.

Editar centro

Nombre

Hospital comarcal de Vinaròs

Teléfono

795663846

Zona básica de salud

Vinaròs

Salas

No hay salas en el centro

Este campo no se puede modificar, si quiere cambiar o añadir una sala, acceda a la misma

Editar

Figura 92. Resultado testeo Centros Selenium Web 1

Editar centro

Nombre

Centro de salud del Grau

Teléfono

622084958

Zona básica de salud

Gandía

Salas

No hay salas en el centro

Este campo no se puede modificar, si quiere cambiar o añadir una sala, acceda a la misma

Editar

Figura 93. Resultado testeo Centros Selenium Web 2

2.7 Salas

En el caso de las salas, los testeos se van a focalizar en:

- Intentar añadir o modificar un coordinador diferente al que ya hay pero que ya exista en la base de datos.

- Intentar añadir o modificar la IP de una aplicación diferente al que ya hay pero que ya exista en la base de datos.
- Intentar añadir o modificar la MAC de una aplicación diferente al que ya hay pero que ya exista en la base de datos.
- Añadir una sala con todos los datos teóricamente correctos.
- Editar una sala.

Los resultados se pueden visualizar en la y son en todos los casos los esperados (en los 3 primeros, no se modifica la sala, pero en los 2 siguientes sí).

```
>>>
==== RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testRooms.py ====
La prueba ha sido un éxito, la sala se ha modificado correctamente
.La prueba ha sido un éxito, no se ha modificado la sala ya que la MAC del coordinador está repetida
.La prueba ha sido un éxito, no se ha modificado la sala ya que la IP de la aplicación está repetida
.La prueba ha sido un éxito, no se ha modificado la sala ya que la MAC de la aplicación está repetida
.La prueba ha sido un éxito, la sala se ha creado correctamente
.
-----
Ran 5 tests in 33.348s

OK
```

Figura 94. Resultados testeo Salas Selenium Python

Al igual que en los casos anteriores, se pueden visualizar los resultados desde la Figura 95 que muestra los resultados desde la web.



Sala videollamadas planta baja	
Centro: Hospital comarcal de Vinaròs	
Coordinador: 2c:54:91:88:c9:e3	
Ip aplic.: 192.168.5.18 MAC aplic.: 00:1e:c2:9e:28:6c	
Sala videollamadas Selenium	
Centro: Centro de Salud Algemesi	
Coordinador: 69:ab:f4:3b:2c:e4	
Ip aplic.: 192.168.54.23 MAC aplic.: fe:54:2b:00:b5:3c	

Figura 95. Resultados testeo Salas Selenium Web

2.8 Dispositivos

En este caso se harán las consiguientes comprobaciones:

- Intentar añadir o modificar un dispositivo que tenga una MAC distinta a la suya pero que ya esté en la base de datos.
- Intentar crear o editar un dispositivo que ya tenga una habitación asignada (por ejemplo, que una determinada sala ya tenga asignada una báscula).
- Editar un dispositivo cambiando el tipo y asegurándonos de que solo se registra la información técnica pertinente.

- Crear un nuevo dispositivo con los valores correctos.

En la Figura 96 se observa como todos y cada uno de los casos están controlados, es decir, funcionan correctamente.

```
>>>
=== RESTART: C:/Users/Laura/Desktop/Universitat/TFG/tfg/Tests/testDevices.py ===
La prueba ha sido un éxito, el dispositivo se ha modificado correctamente
.La prueba ha sido un éxito, no se ha modificado el dispositivo ya que ya hay un
o de este tipo asignado a ese coordinador
.La prueba ha sido un éxito, no se ha modificado el dispositivo ya que la MAC es
tá repetida
.La prueba ha sido un éxito, el dispositivo se ha creado correctamente
.
-----
Ran 4 tests in 28.957s

OK
```

Figura 96. Resultado testeo Dispositivos Selenium Python

Adicionalmente, en la Figura 97 y en la Figura 98 se puede observar desde la página web como el testeo ha funcionado correctamente.

The screenshot shows a web form titled 'Editar dispositivo'. It has a 'Coordinador' dropdown menu with the value '00:ab:01:a3:20:y1'. Below it, there are two input fields: 'MAC' with the value 'ca:ca:cccc:cccc' and 'Tipo' with a dropdown menu showing 'Temperatura'. There is a section titled 'Información técnica' with a warning message: 'Por favor, no cambie los valores por defecto a no ser que tenga conocimientos técnicos'. Below this, there are two input fields: 'Servicio' with the value '0000ffff-0000-1000-8000-00805f9b:' and 'Característica' with the value '0000ffff4-0000-1000-8000-00805f9b:'. At the bottom right, there is a green 'Editar' button.

Figura 97. Resultado testeo Dispositivos Selenium Web 1

The screenshot shows a web form titled 'Editar dispositivo'. It has a 'Coordinador' dropdown menu with the value '69:ab:f4:3b:2ce4'. Below it, there are two input fields: 'MAC' with the value 'selenium:test' and 'Tipo' with a dropdown menu showing 'Temperatura'. There is a section titled 'Información técnica' with a warning message: 'Por favor, no cambie los valores por defecto a no ser que tenga conocimientos técnicos'. Below this, there are two input fields: 'Servicio' with the value '0000ffff-0000-1000-8000-00805f9b:' and 'Característica' with the value '0000ffff4-0000-1000-8000-00805f9b:'. At the bottom right, there is a green 'Editar' button.

Figura 98. Resultado testeo Dispositivos Selenium Web 2

2.9 Conclusiones

Tal como se ha podido observar, tras la realización de 26 pruebas mediante Selenium, se han detectado un total de 3 errores (todos y cada uno de ellos, ya corregidos). En la se puede observar un resumen de dichos tests:

Tabla 1. Resumen tests Selenium

	Nº de tests	Nº de errores detectados	Nº de errores solucionados
Login	2	0	0
Eventos	4	0	0
Pacientes	2	2	2
Médicos	5	1	1
ZBS	2	0	0
Centros	2	0	0
Salas	5	0	0

La parte de comunicación entre la Raspberry, la aplicación del paciente y la del médico es mucho más compleja de testear con Selenium, por ende, se deja dicha sección para el testeo manual.

3 Instalaciones

3.1 Generales

3.1.1 Control de versiones Sourcetree

Sourcetree [1] es un software que proporciona una GUI para GIT. Este software está únicamente disponible para Windows y MAC, por ende, no lo podremos utilizar directamente en la Raspberry (en ese caso, se puede utilizar git por consola por ejemplo). Sourcetree nos ayudará a realizar el control de versiones sobre nuestro proyecto de manera más sencilla. La instalación es tan simple como descargar el ejecutable para el sistema operativo correspondiente (Windows) y ejecutarlo.

3.1.2 Visual Studio Code

El editor de código utilizado durante el desarrollo del proyecto es Visual Studio Code [2].

Algunas extensiones instaladas son:

- Angular 10 Snippets - TypeScript, Html, Angular Material, ngRx, RxJS & Flex Layout (solo necesario para Windows).
- Angular language service (solo necesario para Windows).
- Angular2-inline (solo necesario para Windows).
- GitLens – Git supercharged.
- Material Icon Theme.
- PHP Debug (solo necesario para Windows).
- PHP Inteliphense (solo necesario para Windows).
- Pylance.
- Python.
- TypeScript Importer (solo necesario para Windows).
- TODO Highlight.

3.1.2.1 *Instalación para Windows*

La instalación para Windows es tan simple como descargar el archivo .exe de la página web [3] y ejecutarlo. En este caso se ha descargado la versión de user installer de 64 bits.

3.1.2.2 *Instalación para Raspberry Pi*

En primer lugar es necesario descargar la versión para Ubuntu en formato .dev ARM desde la web oficial [3] . Posteriormente, se realiza su instalación desde la consola de comandos en el directorio donde se ha descargado el archivo (Downloads en este caso):

```
# Instalamos el programa
$ sudo apt install ./code_1.64.2-1644444201_armhf.deb
```

3.1.3 Python

Python [4] será de utilidad para programar la parte de la Raspberry encargada de leer los valores de los distintos dispositivos y además para hacer los testeos en Windows para la página web.

3.1.3.1 Instalación para Windows

La instalación para Windows se basa en la descarga del ejecutable que se puede obtener de la página web oficial [4]. En este caso, la versión instalada es la 3.8. Hay que tener muy en consideración tanto esta versión como la del navegador que se vaya a utilizar para poder hacer la instalación de Selenium.

3.1.3.2 Instalación para Raspberry Pi

No es necesaria la instalación de Python para Raspberry Pi ya que en principio tanto la versión 3 de Python como su IDE viene previamente instalados en el sistema operativo de Raspberry Pi OS.

3.2 Raspberry Pi

3.2.1 Bluepy

Bluepy [5] es un módulo de Python que permite realizar comunicaciones con dispositivos BLE (Bluetooth Low Energy). En este caso, esta librería es indispensable para poder recibir los datos transmitidos por los distintos dispositivos. La instalación se puede realizar plenamente desde la consola:

```
# Actualizamos los repositorios e instalamos la versión específica para Python 3
$ sudo apt-get install python3-pip libglib2.0-dev
$ sudo pip3 install bluepy
```

3.2.2 PM2

PM2 [6], es un administrador de procesos demonio que permite autoejecutar aplicaciones periódicamente o al reiniciar el dispositivo de forma fácil. La instalación del software se realiza desde la consola de comandos:

```
$ sudo npm install -g pm2
```

En este caso, si lo que se quiere es que el software encargado de esperar peticiones de toma de medidas y de recoger los datos de los dispositivos se autoejecute de manera automática al reiniciarse la Raspberry (coordinador) basta con realizar el siguiente proceso:

```
$ cd TFG/Raspberry
# Es importante poner como interprete python 3, sino se indica nada, se lanza por defecto con python 2. También es importante poner sudo, ya que sin él, no leerá los valores de los dispositivos BLE que se basen en características
```

```
$ sudo pm2 start main.py --name PoCAR --interpreter python3
```

Indicamos la configuración inicial para que al reiniciarse la Raspberry se autoejecute de nuevo

```
$ sudo pm2 startup systemd
```

```
$ sudo pm2 save
```

Adicionalmente otros comandos interesantes son:

Ver el listado de aplicaciones y su estado

```
$ sudo pm2 list
```

Reiniciar la aplicación

```
$ sudo pm2 restart PoCAR
```

Para la aplicación. En este caso la parará pero no la borrará, es decir, se podrá reiniciar

```
$ sudo pm2 stop PoCAR
```

Eliminar la aplicación

```
$ sudo pm2 delete PoCAR
```

3.3 Windows

3.3.1 Xampp

Xampp [7] es una distribución de apache que permitirá entre otros usar PHP o MySQL. Ambos son necesarios en el sistema, el PHP para la API que realiza las peticiones a la base de datos; y MySQL para poder desarrollar bases de datos. La descarga e instalación se puede hacer de forma sencilla mediante la página web [7].

3.3.2 Node JS

Node JS [8] es un entorno de ejecución para JavaScript necesario en este caso para la instalación y funcionamiento de Angular. La descarga del ejecutable para Windows se puede descargar desde la web oficial, se recomienda usar la versión LTS para prevenir posibles fallos de las nuevas versiones menos testeadas. Para instalarlo, basta con lanzar el ejecutable. Es conveniente realizar los siguientes pasos para asegurar que el proceso de instalación de Angular sea correcto:

Actualizar NPM, el gesto de paquetes y descargar las dependencias actuales

```
npm install -g npm@latest
```

Borrar el cache de NPM

```
npm cache clean --force
```

Desactivar las auditorias NPM

```
npm set audit false
```

3.3.3 Angular

Angular [9], es un framework utilizado para el desarrollo multiplataforma (aplicaciones PWA). El proceso de instalación requiere del uso de NPM. La versión de Angular Client es muy importante, en este caso, se usa la versión 12.1.4. Por tanto la instalación será:

```
# Si hay otra versión de angular instalada, desinstalarla
npm uninstall -g angular-cli
npm cache clean

# Instalar una versión concreta de Angular
npm install -g angular-cli@12.1.4
```

3.3.4 Selenium

Selenium [10] permite realizar testeos sobre las aplicaciones webs. Su instalación para Python requerirá ir a la carpeta donde está instalado Python: C:\Users\<User>\AppData\Local\Programs\Python\Python<version> (en este caso: C:\Users\Laura\AppData\Local\Programs\Python\Python38-32). El proceso de instalación es:

```
# Acceder al directorio
cd C:\Users\Laura\AppData\Local\Programs\Python\Python38-32\Scripts

# Instalar selenium
pip install -U selenium
```

En este caso, como vamos a utilizar Google Chrome, habrá que acceder a la página [11] para descargar los drivers. Es muy importante saber que versión de navegador tenemos. En Google Chrome se puede obtener desde los tres puntos de la barra superior a la derecha → Ayuda → Información de Google Chrome. En este caso, tal como se puede ver en la Figura 99 se puede observar que la versión es la 103.0.5060.66.

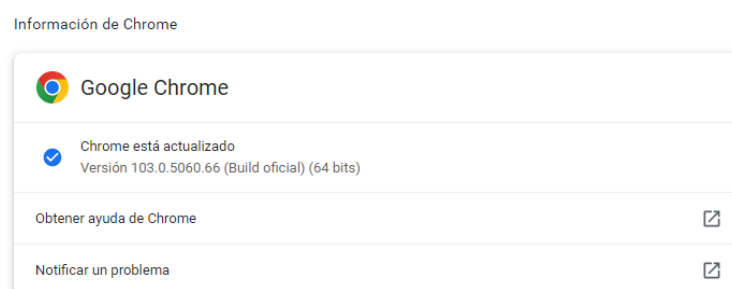


Figura 99. Información Chrome

Por tanto la versión a descargar será la [12], más concretamente, la de Windows de 32 bits (no hay de 64) en versión zip. Una vez descargado, se habrá de descomprimir y copiar su ejecutable en el directorio: C:\Users\Laura\AppData\Local\Programs\Python\Python38-32.

3.3.5 Axure

Axure [13] es un software para desarrollo de UX a nivel profesional para la realización de prototipos. Se ha utilizado este programa para poder crear los wireframes y los mockups. Este software es de pago, en este caso, se hace uso de

la licencia de estudiante universitario. La instalación es sencilla, basta con descargar el ejecutable desde la página web oficial y ejecutarlo.

3.4 Librerías Angular

3.4.1 PrimeNG y PrimeFlex

PrimerNG [14] y PrimeFlex [15] son una colección de componentes de interfaz de usuario para Angular utilizada para facilitar el desarrollo estético de las webs de manera unificada y más sencilla. La instalación se realiza desde el terminal:

```
# Instalación PrimeNG
npm install primeng --save
npm install primeicons --save

# Instalación PrimeFlex
npm install primeflex
```

Es importante añadir dichos estilos a la sección de estilos de Angular. Para ello, hay que modificar el archivo angular.json, más concretamente, en build → styles hay que añadir:

```
"node_modules/primeng/resources/primeng.css",
"node_modules/primeflex/primeflex.css",
"node_modules/primeicons/primeicons.css",
"node_modules/primeng/resources/themes/lara-light-teal/theme.css"
```

3.4.2 Angular 12.0+ calendar

La librería de Angular calendar [16] permitirá desarrollar el sistema de eventos más fácilmente, sobre todo a nivel de interfaz. La instalación se realiza a través de la consola:

```
npm install --save angular-calendar date-fns
```

Al igual que en PrimeNG y PrimeFlex es necesario añadir el estilo al archivo angular.json, en la sección de build → styles. La línea que hay que añadir es:

```
"node_modules/angular-calendar/css/angular-calendar.css"
```

3.4.3 AWS IoT Device SDK V2

La librería de AWS IoT Device [17], permite establecer la comunicación con AWS IoT mediante MQTT. Para que funcione correctamente es necesario en primer lugar instalar los paquetes necesarios:


```
npm install aws-iot-device-sdk-v2
npm install @types/node --save-dev
npm install util
npm install buffer
npm install browser
```

Es necesario añadir node al archivo tsconfig.json, más concretamente dentro de angularCompilerOptions → types. Debe quedar de la siguiente manera:

```
{
  "angularCompilerOptions": {
    "types" : ["node"]
  }
}
```

En el archivo de polyfills.ts se ha de añadir el siguiente código:

```
import 'zone.js';

import * as process from 'process';
import { Buffer } from 'buffer';

import 'zone.js';

window.process = process;
(window as any).global = window;
global.Buffer = global.Buffer || Buffer;
```

3.4.4 VideoSDK

La API de VideoSDK [18] permite realizar aplicaciones complejas de video y audio en directo. En este caso, es de gran utilidad para realizar la videollamada entre el paciente y el médico.

En este caso se ha utilizado la Prebuilt SDK. Para ello desde la consola:

```
# Es muy importante poner entre comillas el nombre o no lo encontrará
npm install '@videosdk.live/rtc-js-prebuilt'
```

Para que la API sea correctamente reconocida (es en JavaScript) se ha de crear un archivo llamado typings.d.ts en la raíz del módulo en que lo vayamos a utilizar. Dentro del archivo solo se ha de escribir la siguiente línea:

```
declare module '@videosdk.live/rtc-js-prebuilt';
```


4 Manuales de instrucciones

Se han elaborado diversos documentos de ayuda para facilitar el uso de todo el sistema a los distintos usuarios a los cuales va dirigida.

El manual para médicos cuenta con una detallada descripción de como hacer uso de todos los elementos de la página web tanto si se es administrador como si no. Adicionalmente se proporcionan instrucciones y consejos, como por ejemplo sobre elementos que no deberían utilizar sin disponer de conocimientos técnicos o sobre medidas de seguridad.

El manual para técnicos explica todo el proceso a realizar para añadir o editar tanto todos los elementos del sistema. Más concretamente, se explica como crear un nuevo coordinador y obtener todos los datos relevantes necesarios para registrar el sistema en la web. También se explica como instalar la PWA de los pacientes y los parámetros necesarios que se necesita conocer sobre los dispositivos en los cuales se ha instalado.

Por último, hay un cartel explicativo de como utilizar la aplicación del paciente. Este ha sido creado, porque a pesar de ser muy fácil, los pacientes podrían no haber realizado nunca una videollamada y eso les ayudaría a gestionarlo de mejor manera.

5 Referencias bibliográficas

- [1] "Sourcetree." <https://www.sourcetreeapp.com/> (accessed Jul. 03, 2022).
- [2] "Visual Studio Code." <https://code.visualstudio.com/> (accessed Jul. 03, 2022).
- [3] "Visual Studio Code Descargas." <https://code.visualstudio.com/#alt-downloads> (accessed Jul. 03, 2022).
- [4] "Python." <https://www.python.org/downloads/> (accessed Jul. 03, 2022).
- [5] I. Harvey, "bluepy," Oct. 19, 2013. <https://github.com/lanHarvey/bluepy> (accessed Jun. 26, 2022).
- [6] "PM2." <https://pm2.keymetrics.io/> (accessed Jun. 28, 2022).
- [7] "Xampp." <https://www.apachefriends.org/es/index.html> (accessed Jul. 03, 2022).
- [8] "Node.js." <https://nodejs.org/es/> (accessed Jul. 03, 2022).
- [9] "Angular." <https://angular.io/> (accessed Jun. 29, 2022).
- [10] "Selenium." <https://www.selenium.dev/> (accessed Jun. 26, 2022).
- [11] "ChromeDriver." <https://chromedriver.chromium.org/downloads> (accessed Jul. 03, 2022).
- [12] "ChromeDriver 103.0.5060.53."
<https://chromedriver.storage.googleapis.com/index.html?path=103.0.5060.53/> (accessed Jul. 03, 2022).
- [13] "Axure." <https://www.axure.com/> (accessed Jul. 03, 2022).
- [14] "PrimeNg." <https://www.primefaces.org/primeng/> (accessed Jun. 29, 2022).
- [15] "PrimeFlex." <https://www.primefaces.org/primeflex/> (accessed Jun. 29, 2022).
- [16] M. Lewis, "Angular 12.0+ calendar," Apr. 26, 2016. <https://openbase.com/js/angular-calendar/documentation> (accessed Jun. 29, 2022).
- [17] AWS, "AWS IoT Device SDK V2." <https://github.com/aws/aws-iot-device-sdk-js-v2> (accessed Jul. 03, 2022).
- [18] "VideoSdk." <https://videosdk.live/> (accessed Jun. 29, 2022).