



**Universidad
Europea**

Proyecto Informática II

App Control Médico

—
Daniel Santiago, Javier Liébana
Universidad Europea de Madrid
Ingeniería Informática Online
Curso 2020-2021



Título:

MedicApp

Autores:

Javier Liébana y Daniel Santiago

Tutor del proyecto:

Ana del Valle Corrales

Fecha de lectura:

15/02/2021

En Madrid

Ana del Valle Corrales
Profesora de la asignatura Proyecto de Informatica II

Resumen	3
Abstract	3
Agradecimientos	4
Derechos de autor	5
Introducción	6
Objetivos	6
Motivación	7
Antecedentes	7
Desarrollo del proyecto	7
Herramientas tecnológicas	8
Planificación	13
Descripción del trabajo realizado	17
Ventanas	17
Login	17
Ventanas médica	19
Ventana con acceso a los datos del usuario	20
Perfil	21
Citas	22
Sensor	23
Temperatura	24
Familiares	26
Paciente	27
Perfil	28
Citas	28
Sensor	30
Temperatura	31
Familiar	33
Resultado y validación	37
Conclusión	37
Innovación	37
Trabajo Futuro	37
Bibliografía y Webgrafía	37
Anexos	38
Manual de Usuario	38

Resumen

La aplicación MedicApp resuelve el problema de aquellas personas mayores que necesitan una atención médica diaria o de un control por parte de sus familiares de una forma rápida y sencilla ayudando hacia estos usuarios y a quienes los rodean dándoles esa sensación de tranquilidad y seguridad que tanto se disfruta. Esta aplicación contará con varios roles: médico, paciente y familiares, cada uno tendrá distintas funciones y ventanas, pero cabe destacar el usuario principal, el paciente, éste podrá mostrar a su médico asignado y a sus familiares un control de temperatura y humedad en el hogar diario, un chat con su médico de cabecera y un sensor de movimiento por si sale de casa. También se implementará un sistema de citas online en futuras actualizaciones.

Nuestra aplicación busca enlazar a médicos con pacientes y familiares de una forma simple y amigable para que cualquier usuario sea capaz de usarla. Es una aplicación gratuita y aunque nuestro público mayoritario serán usuarios de la tercera edad que residen en sus casas, perfectamente se podrían registrar pacientes de cualquier edad.

Abstract

The MedicApp application solves the problem of those elderly people who need daily medical attention or control by their relatives in a fast and simple way, helping these users and those around them, giving them that feeling of tranquility and security that is so enjoyed. . This application will have several roles: doctor, patient and family, each one will have different functions and windows, but it is worth highlighting the main user, the patient, who will be able to show his assigned doctor and his family a daily temperature control, chat with your GP and a motion sensor in case you leave home. An online dating system will also be implemented.

Our application seeks to link doctors with patients and families in a simple and friendly way so that any user is able to use it. It is a free application and although our majority audience will be elderly users who reside at home, patients of any age could perfectly be registered.

Agradecimientos

Queremos agradecer primero al equipo de desarrollo, a la Universidad Europea y sobre todo a nuestros profesores Alberto Fuentes Rodriguez y Ana del Valle Corrales por brindarnos la oportunidad de desarrollar una aplicación que de verdad nos ha inspirado para esforzarnos y seguir luchando pese a todos los contratiempos que hemos sufrido.

Iconos desarrollados por [FreePik](#).

Derechos de autor



Esta obra se distribuye bajo una licencia Creative Commons. Se permite la copia, distribución, uso y comunicación de la obra si se respetan las siguientes condiciones:

- Se debe reconocer explícitamente la autoría de la obra incluyendo esta nota y su enlace.
- La copia será literal y completa
- No se podrá hacer uso de los derechos permitidos con fines comerciales, salvo permiso expreso de los autores.

El texto precedente no es la licencia completa sino una nota orientativa de la licencia original completa(jurídicamente válida) que puede encontrarse en:
<http://creativecommons.org/licenses/by-nc-nd/3.0/deed.es>

Introducción

La aplicación MedicApp resuelve la necesidad de poner en contacto a pacientes y familiares con médicos de forma rápida y sencilla, integrando herramientas para facilitar las tareas a la hora de llevar el control de un paciente. Creemos que una aplicación de escritorio enfocada a la tercera edad es necesaria en estos tiempos donde salir de casa expone a las personas a un riesgo alto de contagio por lo que llevar un control vía online es la forma más segura de saber para los médicos si los pacientes están bien. Con nuestra aplicación reduciremos las revisiones médicas para estos usuarios en un 50%, es decir, evitamos de manera drástica su exposición al COVID-19 en centros de salud y hospitales.

Objetivos

- Implementar una base de datos en la nube y a tiempo real.
- Geolocalizar a los usuarios que requieran de esta necesidad.
- Desarrollar una aplicación sencilla y eficaz.
- Desarrollar ventanas amigables para todo tipo de usuarios.
- Conectar nuestra aplicación con sensores de movimiento, temperatura y humedad.
- Desarrollar un calendario de citas online.
- Chat funcional para médicos, pacientes y familiares.

La lista de objetivos iniciales recoge las principales puntos que definirán MedicApp, para la primera parte de esta aplicación se llevaría a cabo el desarrollo de los siguientes puntos.

Base de datos: pese a no ser una base de datos en tiempo real, la estructura está definida para serlo y en los siguientes sprint será implementada en la aplicación.

Interfaz: Estamos convencidos de que hemos logrado desarrollar una interfaz sencilla y amigable para usuarios de cualquier nivel.

Funcionalidad de la aplicación java: Hemos logrado una aplicación perfectamente funcional para los 3 roles de usuario. Presentará un login, registro para familiares y ventanas que mostrarán los datos del paciente indicado.

Chat: Será un registro de mensajes entre usuarios, no descartamos la creación de un chat funcional tras la implementación de la base de datos.

Calendario de citas: No hemos podido implementar sistema de citas online por falta de tiempo pero para los siguientes sprint será añadido sin falta.

El resto de funcionalidades se añadirán en futuros sprint.

Motivación

Nos sentimos muy atraídos por la idea debido a que es algo que no está hecho de una forma que de verdad sea útil y sencilla para los usuarios, y que de verdad es una obra que puede ayudar a la sociedad en estos tiempos por lo que no nos costó mucho ponernos manos a la obra.

Antecedentes

El principal antecedente fue la idea de nuestro profesor para realizar el proyecto, hemos cambiado algunas partes para hacerla a nuestra idea pero la inspiración viene de su idea compartida en clase.

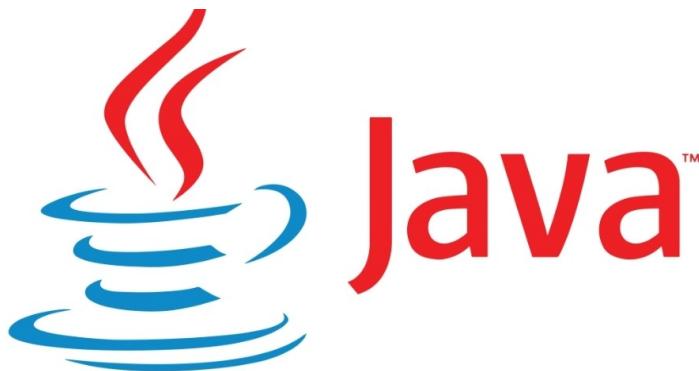
Desarrollo del proyecto

La realización de este proyecto ha sido compleja pues partimos de una idea inicial y por motivos relacionados con los cortos plazos para realizar esta aplicación hemos tenido que implementar muchas de las funcionalidades que queríamos añadir de una manera más sencilla cuando teníamos las ideas claras y sabíamos cómo hacer la mayoría de las cosas, también sumar impedimentos relacionados con el nivel de las tareas que teníamos propuestas o simplemente por errores surgidos a lo largo del proyecto que nos impedían avanzar en el desarrollo de este. A pesar de estas complicaciones hemos sabido llevar bien el desarrollo del proyecto gracias a herramientas de gestión de tareas, en nuestro caso usamos Trello, en la cual anotamos las tareas a realizar para ir completando poco a poco nuestra aplicación. También Github, para control de versiones, así posibilitar que varias personas puedan trabajar en un mismo proyecto desde distintos dispositivos a distintos tiempos, sin entorpecer el trabajo de los distintos compañeros del proyecto.

Para gestionar la realización de tareas hemos adoptado la metodología Agile Scrum a partir de sprints para realizar esas tareas en ciertas fechas y así ser más eficientes.

Herramientas tecnológicas

Java



Java es un lenguaje de programación orientado a objetos que se incorporó al ámbito de la informática en los años noventa. La idea de Java es que pueda realizarse programas con la posibilidad de ejecutarse en cualquier contexto, en cualquier ambiente, siendo así su portabilidad uno de sus principales logros. Fue desarrollado por Sun Microsystems, posteriormente adquirido por Oracle. En la actualidad puede utilizarse de modo gratuito, pudiéndose conseguir sin problemas un paquete para desarrolladores que oriente la actividad de programar en este lenguaje. Puede ser modificado por cualquiera, circunstancia que lo convierte en lo que comúnmente se denomina “código abierto”.

Cuando se programa con un lenguaje, se utiliza un tipo de código que la máquina es incapaz de reconocer. Es por ello que se requiere de un proceso de compilación, esto es, de “traducir” el lenguaje utilizado para que la máquina sea capaz de reconocerlo y procesarlo. Cada vez que un código requiere ser ejecutado en un ambiente distinto, deberá compilarse nuevamente. Con el desarrollo de Java se intentó evitar este tipo de circunstancia, haciendo que el código corriera en cualquier ambiente. El modo en que se logró este objetivo es desarrollando una máquina virtual que ejecutase el código compilado; esto significa que la “traducción” ya no se hace para que sea comprendida por un determinado sistema operativo; se hace para que la intérprete la “máquina virtual”, que puede correr en cualquier sistema operativo. De esta manera, solo es necesario realizar una compilación, que podrá ser utilizada en cualquier entorno que tenga instalada la máquina virtual java

Eclipse



Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

```
export default class ErrorBubble extends PureComponent {
  static propTypes = {
    error: PropTypes.string,
    className: PropTypes.string,
    children: PropTypes.node
  };

  render() {
    const {children, className, ...restProps} = this.props;

    const errorBubbleClasses = classNames(styles.errorBubble, className);

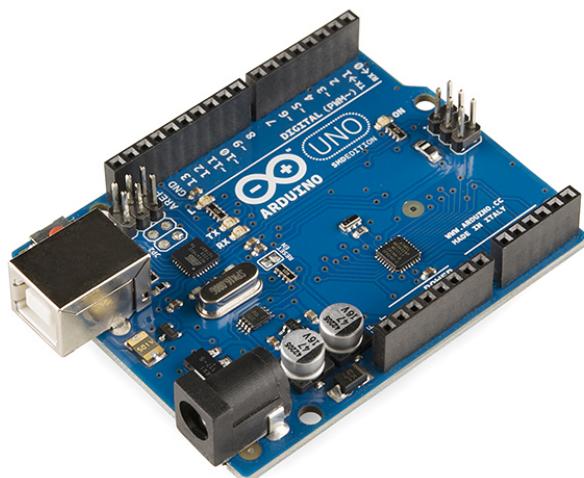
    return (
      <div className={styles.errorBubbleWrapper}>
        {children.map(child => cloneElement(child, restProps))}

        {restProps.error && (
          <P>
            {restProps.error}
          </div>
        )}
      </div>
    );
  }
}
```

Arduino:



Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General de GNU (GPL) y la Licencia Pública General Reducida de GNU (LGPL),¹ permitiendo la manufactura de las placas Arduino y distribución del software por cualquier individuo. Las placas Arduino están disponibles comercialmente en forma de placas ensambladas o también en forma de kits, hazlo tú mismo (del inglés DIY: "Do It Yourself").



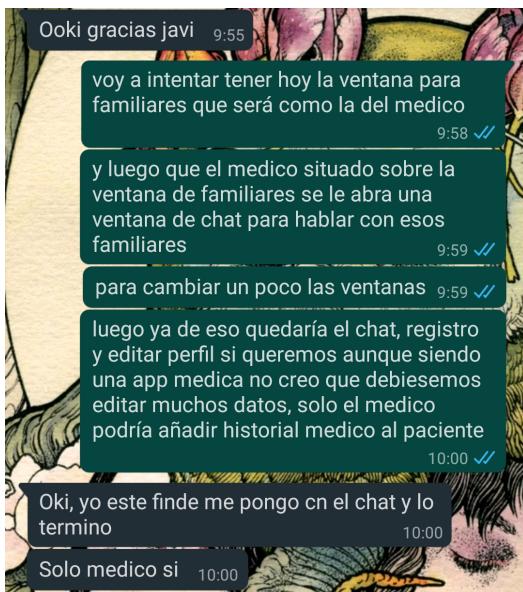
The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.5". The toolbar contains icons for file operations like Open, Save, and Upload. The code editor window displays the "Blink" sketch. The code is as follows:

```
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)  
  delay(1000);                      // wait for a second  
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW  
  delay(1000);                      // wait for a second  
}
```

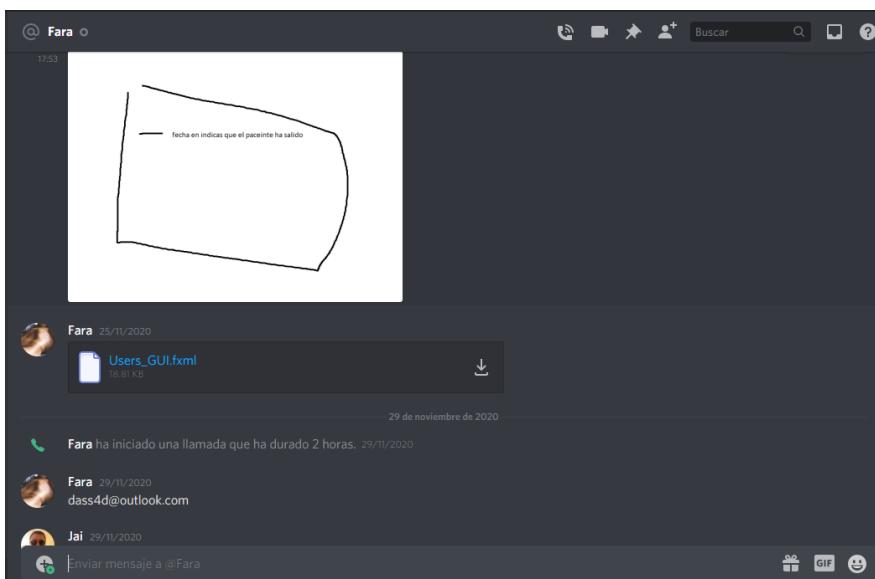
Planificación

Forma de organización del equipo: Cómo se organizará el equipo: canales de comunicación (Slack, GitHub, Trello, Mail, Whatsapp, etc.) tareas de cada uno, que se espera conseguir, temporalización. Como grupos de comunicación tenemos:

- Un grupo de Whatsapp para mensajes rápidos.



- Un grupo de Discord donde hacer reuniones semanales, dejar documentos o compartir clases.



- Foro de la asignatura: para consultar dudas con nuestro tutor para que este nos las pudiera resolver o al menos guiar hacia una posible resolución.

José Enrique Martín Gómez (User) - Ejemplos de aplicaciones - hace 3 meses

Buenas noches Alberto;

En el video de la primera clase comentaste que ibas a subir unos ejemplos de aplicaciones para que nos hagamos una idea de que incluir.

¿Los subiste? no los encuentro.

Un saludo!!

Responder

ALBERTO FUENTES RODRIGUEZ (User) - RE: Ejemplos de aplicaciones - hace 3 meses

Si,

Está en la sección presentación "Contexto del proyecto". También hay otra carpeta con los sensores de arduino, para que les echéis un vistazo.

Saludos!

Ocultar 1 respuesta (1 sin leer)

- Reparto de tareas con Trello: hemos podido gestionar las tareas a realizar y así evitar confusiones entre nosotros por si se realizaba la misma entre varios, que solo la realizará uno. Además de realizar plazos de entrega de tareas, para no retrasar el desarrollo del proyecto.

Agile Sprint Board

Backlog

+ Añada una tarjeta

Sprint Backlog

chat

+ Añada otra tarjeta

In Progress

css

+ Añada otra tarjeta

Complete

- Desarrollo fichero JSON
- ventana medico, cargar lista de pacientes
- reforzar seguridad login
- verificacion login registro
- listas de familiares
- Desarrollo base firebase
- resolver problemas registro
- cargar datos en perfil
- Metodos login-registro
- Conexion entre ventanas

+ Añada otra tarjeta

- Github: para trabajar sobre el proyecto.

This branch is 14 commits ahead of master.		
Pull request	Compare	No description, website, or topics provided.
<hr/>		
 javiliebana	subo el css de Dani.	542b196 3 hours ago
 .settings	subo el css de Dani.	3 hours ago
 img	iconos kawaii (✿◡‿◡)	16 hours ago
 jar	json utils	20 days ago
 src	subo el css de Dani.	3 hours ago
 .classpath	json utils	20 days ago
 .gitignore	pues eso, primer commit	3 months ago
 .project	pues eso, primer commit	3 months ago
 BBDD.json	subo el css de Dani.	3 hours ago
 build.fxbuild	pues eso, primer commit	3 months ago

- Google Drive para documentos.

The screenshot shows a Microsoft Word document with the following details:

- Title:** Proyecto Informática I
- Text:** App Control Médico
- Authors:** Daniel Alejandro Santiago, Javier Liébana
- University:** Universidad Europea de Madrid
- Degree:** Ingeniería Informática Online
- Year:** Curso 2020-2021

The document has a header bar with various icons and a ribbon menu. The main content area includes a navigation sidebar on the left with sections like 'Resumen', 'Abstract', 'Agradecimientos', 'Derechos de autor', 'Introducción', 'Objetivos', 'Motivación', 'Antecedentes', and 'Desarrollo del proyecto'. The central content area contains the title and authors information.

- Hemos implementado una metodología Agile Scrum.



Descripción del trabajo realizado

Lo que se ha buscado con este trabajo es disponer de los datos y las herramientas necesarias de una forma simple de gestionar para todos sus usuarios (Médicos, pacientes y familiares).

Basandonos en estos principios , la interfaz es minimalista a la par que vistosa.

Ventanas

Login

El login es “la carta de presentación” de la aplicación, existen los siguiente usuarios que pueden usar la aplicación:

El usuario Admin: Capaz de administrar la aplicación, es el encargado de dar de alta la cuenta de los diferentes médicos en la base de datos.

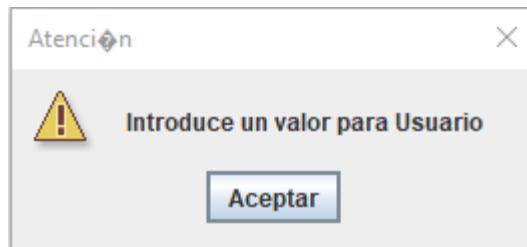
Médico: Debido a que su cuenta ya ha tenido que ser creada previamente por un administrador de sistema, lo único que deben hacer es loguearse para ver en primera instancia una lista con todos los pacientes asociados al mismo que veremos más adelante.

Pacientes: Deben crearse una cuenta para registrarse en la aplicación y luego poder loguearse en la misma para poder sus datos.

Familiares: Al igual que los pacientes deben darse de alta en la aplicación, pero a diferencia de estos, a la hora de realizar el registro deben indicar el familiar al cual vincularse.

Su funcionamiento es muy sencillo, carga la base de datos deserializando el archivo JSON y convirtiéndolo en un ArrayList para poder trabajar con ella dentro del código, cuando introducimos los datos en la ventana de Login los contrasta con la base de datos recorriendo esta y comparando los valores introducidos con los que en este momento se encuentran el ArrayList, si el usuario se encuentra lo siguiente que hace es comprobar el rol del que dispone para así poder cargar sus datos y funciones que son diferentes dependiendo de dicho rol.

Si por el contrario no se encontrase un usuario saldría el siguiente prompt que veremos a continuación:



Ventanas médico

Principal

Esta ventana contiene los pacientes que el médico tiene vinculados. Haciendo clic en cualquiera de ellos nos llevará a su ventana principal donde, como médicos tendrán funciones específicas que veremos a continuación



En la siguiente imagen podemos ver como queda vinculado un médico a un determinado paciente, para ello hay un apartado específico en la construcción del archivo JSON(base de datos) que nos indica para qué paciente qué médico debe atenderle:

```
"id_medico": "medic",
"v_usuario": "",
"lista_chat": [
```

Lo que podemos ver en la imagen sería un apartado del paciente en nuestro caso del usuario(paciente) jorge_canle.

Ventana con acceso a los datos del usuario

Las ventanas están divididas en dos partes que estan presentes ambas en todo momento:

- ❑ La primera, donde se muestra toda la información del paciente dividida en pestañas y que a continuación desglosamos.
- ❑ La segunda, la ventana de chat, donde el médico tendrá contacto directo con el paciente directamente.

The screenshot shows a medical application interface. At the top, there is a header with a profile picture of a doctor and the name "Javier Liebana". Below the header, there are three tabs: "Perfil" (selected), "Sensor", and "Temperatura". The "Perfil" tab contains the following information:

Nombre	Javier
Apellidos	Liebana
DNI	12345678A
Descripción	El paciente es un joven de 22 años invidente, también presenta alergía multiples plantas primaverales, cada años debe someterse a un control en dicha época para evitar riesgos.

Below the profile section, there is a sidebar with a blue header labeled "2020" containing three items: "Historial médico 2020-05-19", "Historial médico 2020-05-19", and "Historial médico 2020-05-22". To the right of the main content area is a vertical sidebar with a blue header labeled "Refrescar". It displays a list of messages from a patient named "pepe": "Hola, buenos dias. ¿Como se encuentra hoy?", "javi: Buenos días doctor, bastante mejor que ayer, como verá la humedad del hogar es la perfecta según me comentó ayer en consulta.", and a timestamp "2021-06-19 21:09:03.0". At the bottom right of the sidebar is a blue button labeled "Enviar".

Perfil

En esta ventana podremos ver los diferentes datos del paciente así como sus historiales médicos dividido por año y mes en la que se realizó.

Como médico tiene la funcionalidad de añadir nuevas historias médicas al paciente.

The screenshot shows a patient profile for "Javier Liebana". The top navigation bar has tabs for "Perfil", "Sensor", and "Temperatura", with "Perfil" selected. Below the tabs, there is a table with the following data:

Nombre	Javier
Apellidos	Liebana
DNI	12345678A
Descripción	El paciente es un joven de 22 años invidente, también presenta alergia a múltiples plantas primaverales, cada año debe someterse a un control en dicha época para evitar riesgos.

To the right of the table is a sidebar with a "Refrescar" button and a scrollable message history:

- 2021-06-19 21:09:03.0
 - pepe: Hola, buenos días. ¿Cómo se encuentra hoy?
 - javi: Buenos días doctor, bastante mejor que ayer, como verá la humedad del hogar es la perfecta según me comentó ayer en consulta.

The screenshot shows the medical history for the year 2020. It includes a section for "Historial médico 2020-05-19" which contains the message: "Se ha instalado la aplicación MedicApp en el hogar del paciente". Below this, there are two more entries: "Historial médico 2020-05-19" and "Historial médico 2020-05-22". A blue "Enviar" button is located at the bottom right of the history panel.

The screenshot shows the medical history for the year 2020. It includes a section for "Historial médico 2020-05-19" which contains the message: "El paciente presenta una fuerte tos y fiebre afirma haber tenido fiebre superior a los 40º, se le ha recetado Turministol 15g". Below this, there is one entry: "Historial médico 2020-05-22". A blue "Enviar" button is located at the bottom right of the history panel.

```

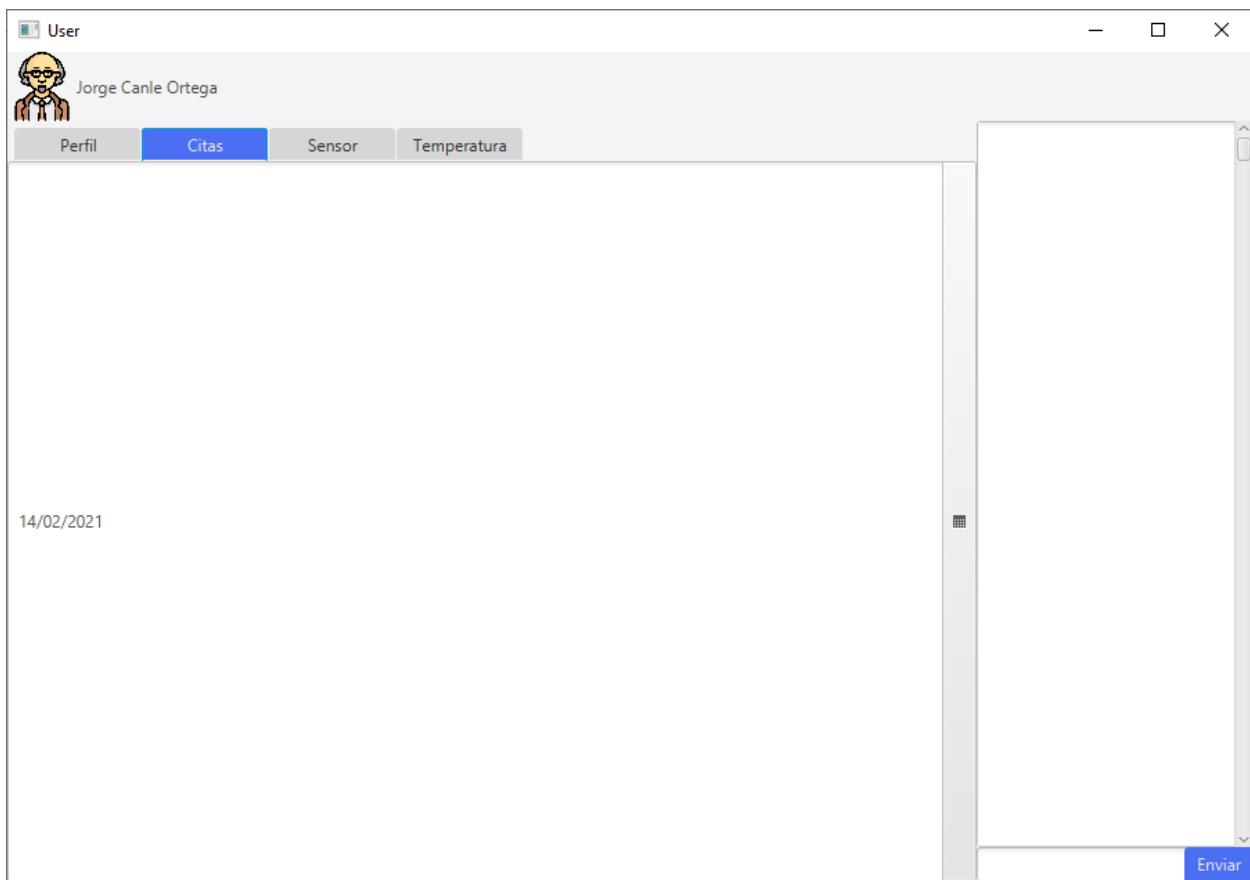
// cargamos las historias medicas en una arraylist
ArrayList<HistoriaMedico> lista_historias = Database.cargarHistoriasMedicas(paciente.getId_paciente());
// sacamos los años para ver cuantos tabs creamos
ArrayList<String> list_year = new ArrayList<String>();
for (HistoriaMedico h : lista_historias) {
    String date = h.getFecha().substring(0, 4);
    if (!list_year.contains(date)) {
        list_year.add(date);
    }
}
// ordenamos la lista de años
Collections.sort(list_year);
for (String s : list_year) {
    accordion = new Accordion();
    for (HistoriaMedico h : lista_historias) {
        if (s.contains(h.getFecha().substring(0, 4))) {
            TitledPane tp = new TitledPane();
            tp.setText("Historial médico " + h.getFecha());
            JFXTextArea descripcion = new JFXTextArea();
            descripcion.setText(h.getDescripcion());
            tp.setContent(descripcion);
            accordion.getPanes().add(tp);
        }
    }
    Tab year = new Tab(s, accordion);
    tabPane.getTabs().add(year);
}

```

Nos vamos a detener un momento en este apartado. Las tablas de los historiales como se puede ver en el código son dinámicas, esto quiere decir que en este “pane” se van creando según los historiales que se vayan guardando y no están creadas de forma fija por lo que según el año o mes en el que se cree un historial médico se irán añadiendo.

Citas

Simplemente es una venta donde disponer de un registro de las citas que el médico tenga con según qué paciente. Esta ventana no se ha implementado en esta versión de la aplicación.



Sensor

Esta pestaña permite al médico la posibilidad de llevar un control sobre una serie de registros recogida por un determinado sensor, en este caso como se muestra en la imagen

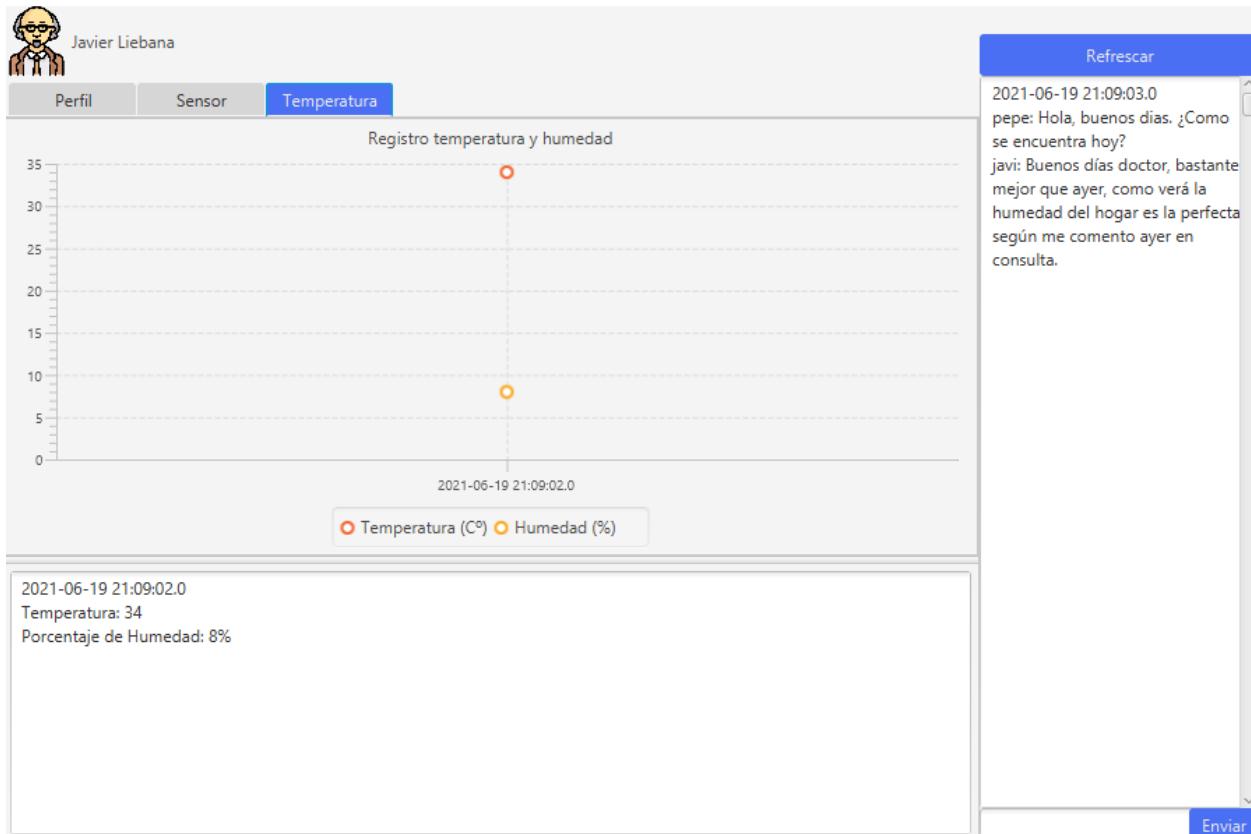
inferior y debido a los tiempo de pandemia que corren, el sensor recoge los datos de entrada y salida del domicilio por parte del paciente pudiendo así evaluar, junto con el paciente si la salida ha sido justificada y si este a podido estar en contacto con gente pudiendo ser peligroso para su salud.



Temperatura

En esta ventana podemos ver en una tabla los datos de la temperatura capturados por el sensor.

Se muestra la temperatura y la humedad del hogar, este dato se va refrescando cada 30 minutos.



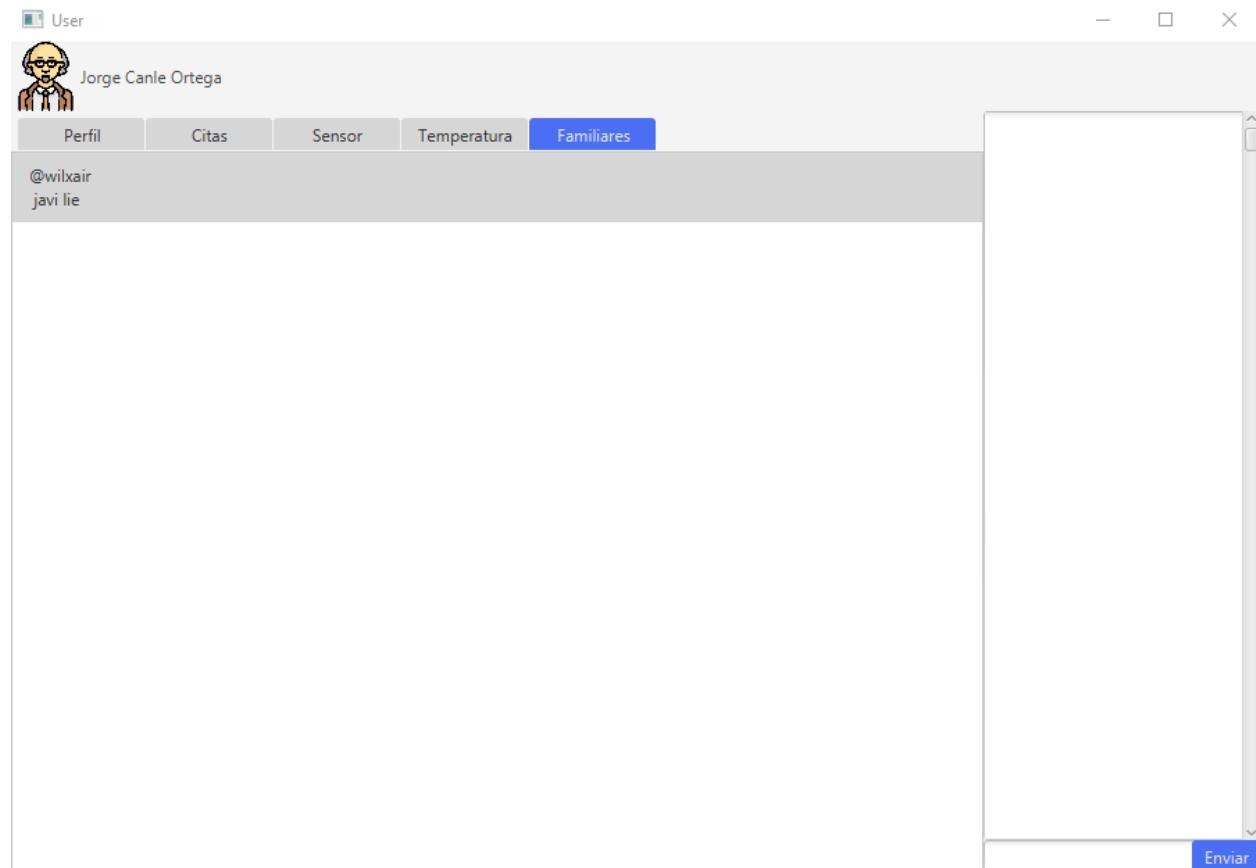
```
for (SensorTemp t : lista_temp) {  
    dato_temp += t.getFecha() + "\nTemperatura: " + t.getTemp() + "\nPorcentaje de Humedad: " + t.getHumedad()  
    + "%\n";  
}  
textAreaTemp.setText(dato_temp);
```

"Pintamos" esos datos en una tabla definida por la temperatura y la humedad recogida y la fecha exacta tal y como podemos visualizar en la ventana.

```
// cargamos los datos del line chart  
final CategoryAxis xAxis = new CategoryAxis();  
final NumberAxis yAxis = new NumberAxis();  
xAxis.setLabel("Month");  
graph_temp.getData().clear();  
graph_temp.setTitle("Registro temperatura y humedad");  
  
XYChart.Series series1 = new XYChart.Series();  
series1.setName("Temperatura (Cº)");  
XYChart.Series series2 = new XYChart.Series();  
series2.setName("Humedad (%)");  
  
for (SensorTemp t : lista_temp) {  
  
    series1.getData().add(new XYChart.Data(t.getFecha(), Double.parseDouble(t.getTemp())));  
    series2.getData().add(new XYChart.Data(t.getFecha(), Double.parseDouble(t.getHumedad())));  
}  
graph_temp.getData().addAll(series1, series2);  
  
// cargamos el chat y sus mensajes  
Chat chat_user = Database.getChat(user.getId());  
chat = chat_user;
```

Familiares

En esta ventana se listan los familiares que han sido asociados a un determinado paciente, lo que permite al médico abrir una ventana de chat directa con el mismo.



Como podemos ver, es una herramienta muy útil en caso de que el médico tenga que informar o recomendar algo directamente a los familiares del paciente, ya sea porque el propio paciente no puede valerse por sí mismo o simplemente por tratar algún tipo de información en la que sea necesario una comunicación directa.

Paciente

Esta ventana al igual que la del médico está dividida en dos partes siendo la primera donde disponer de las diferentes pestañas a las que el paciente tiene acceso y que a continuación listaremos tal como la ventana de chat, permitiendo que esté presente en todo momento en caso de que el paciente necesite realizar cualquier consulta con su médico asignado.

Perfil

Muestra la información del paciente almacenada en la base de datos junto con los diferentes historiales médicos asociados a dicho paciente.

El paciente solo puede leer la información expuesta en dicha página ya que no es necesario que interactúe con ella.

The screenshot shows a web-based application for a patient named Javier Liebana. At the top, there's a header with a user icon and the name "Javier Liebana". Below the header, there are three tabs: "Perfil" (selected), "Sensor", and "Temperatura". The "Perfil" tab displays the following data:

Nombre	Javier
Apellidos	Liebana
DNI	12345678A
Descripción	El paciente es un joven de 22 años invidente, también presenta alergia a múltiples plantas primaveriles, cada año debe someterse a un control en dicha época para evitar riesgos.

Below this section, there's a date selector showing "2020" and a collapsed section titled "Historial médico 2020-05-19" containing the message: "Se ha instalado la aplicación MedicApp en el hogar del paciente". At the bottom, there are two more collapsed sections: "Historial médico 2020-05-19" and "Historial médico 2020-05-22".

To the right of the main content area is a vertical sidebar containing a "Refrescar" button at the top, followed by a scrollable list of messages:

- 2021-06-19 21:09:03.0
- pepe: Hola, buenos días. ¿Cómo se encuentra hoy?
- javi: Buenos días doctor, bastante mejor que ayer, como verá la humedad del hogar es la perfecta según me comentó ayer en consulta.

At the bottom right of the sidebar is a blue "Enviar" button.

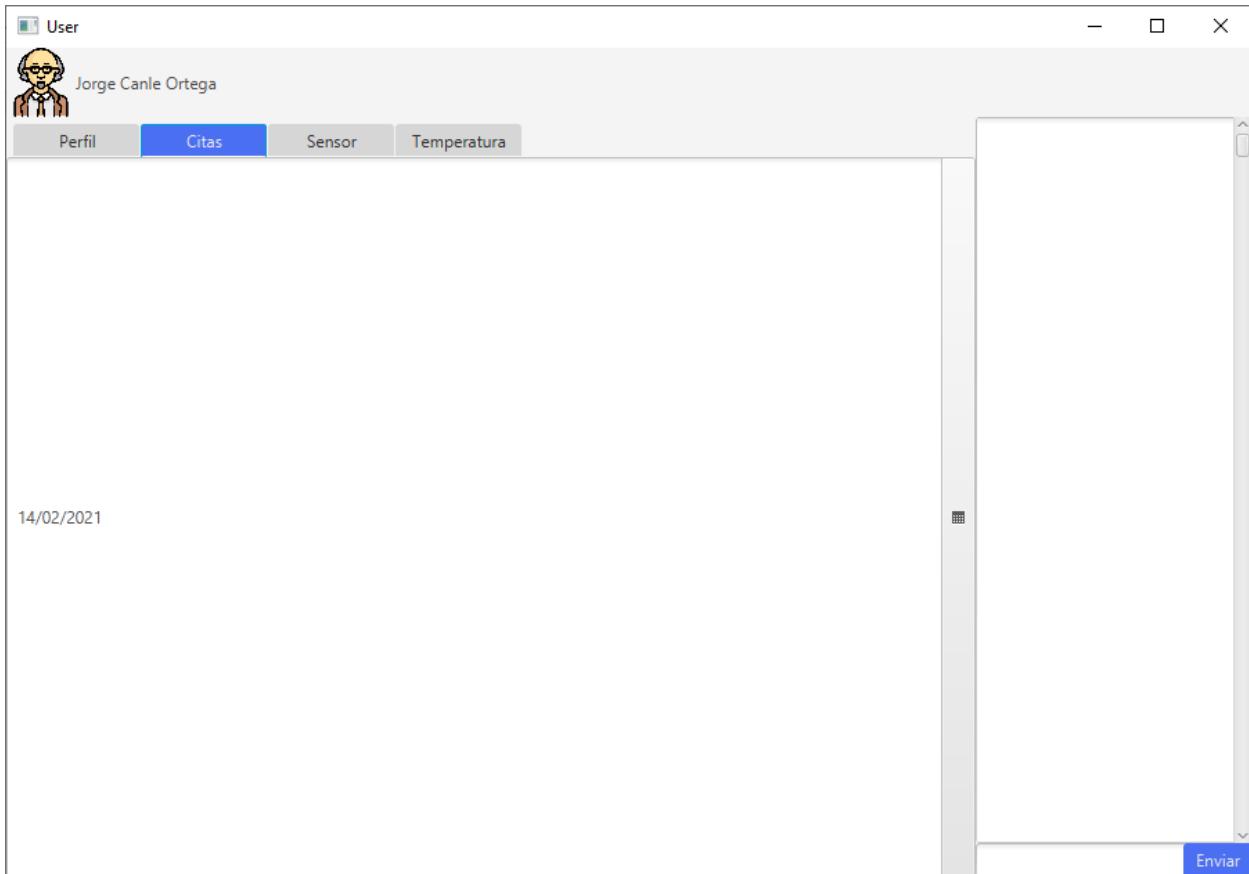
Al logearnos con el nombre de usuario paciente y su correspondiente contraseña el programa carga sus datos correspondientes de la base de datos.



Citas

Permite establecer una cita con el médico. Para ello comprueba los días libres que el médico tiene en su agenda y te permite elegir entre estos.

** Estamos trabajando en esta implementación para ver un uso más extendido de esta funcionalidad.



Sensor

Esta pestaña al igual que en el médico solo se puede ver por el paciente los datos registrados por el sensor asociado en este caso al igual que pasaba en la pestaña del médico se puede ver como registra la entrada y salida del domicilio por el paciente.

Es solo de lectura ya que no es necesario que el paciente interactúe con ella, es un proceso automático que sirve únicamente para llevar un control específico de algo necesario para el paciente.

User
Viendo el perfil de: Jorge Canle Ortega

Perfil Citas Sensor Temperatura

02/11/2017 Ha salido del domicilio
015/11/2017 Ha salido del domicilio
02/12/2019 Ha salido del domicilio

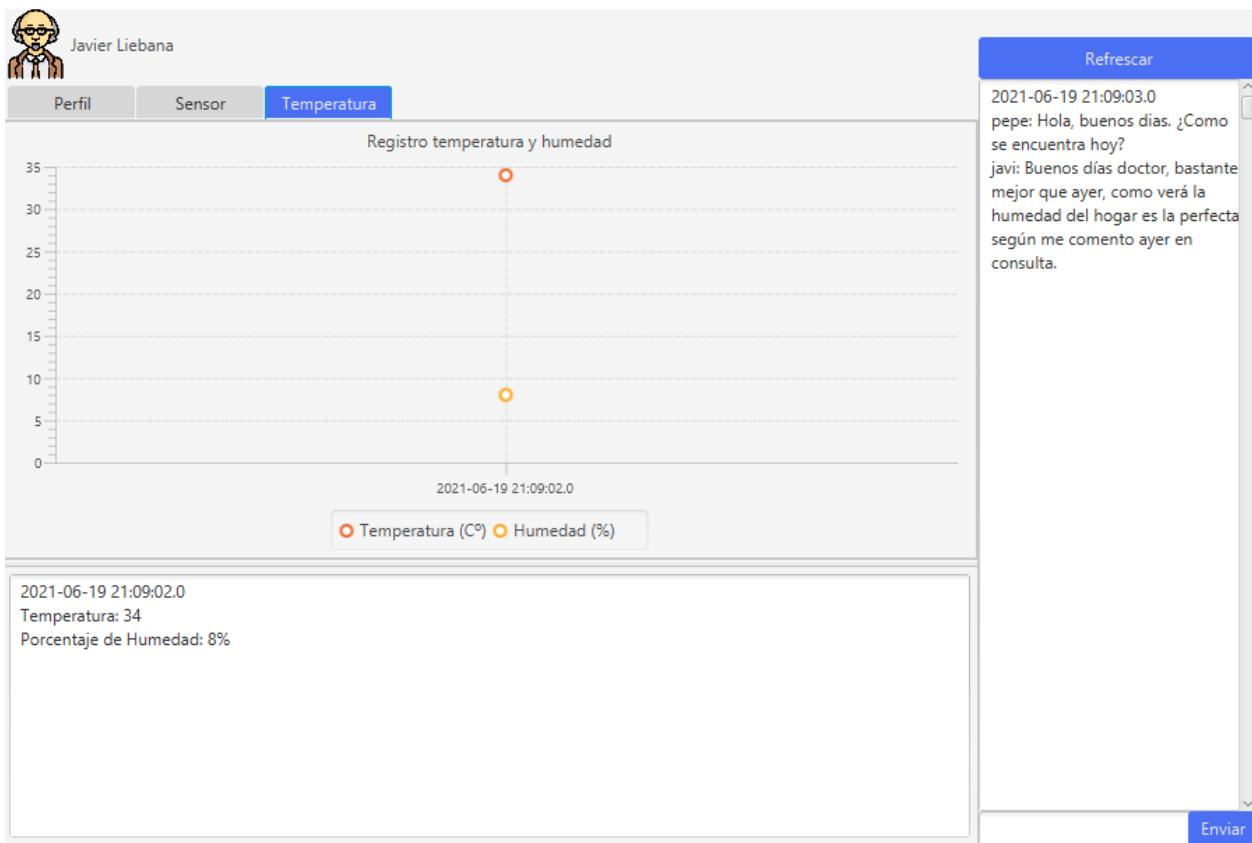
Enviar

Temperatura

En esta ventana podemos ver en una tabla los datos de la temperatura capturados por el sensor.

Se muestra la temperatura y la humedad del hogar, este dato se va refrescando cada 30 minutos.

“Pintamos” esos datos en una tabla definida por la temperatura y la humedad recogida y la fecha exacta tal y como podemos visualizar en la ventana.



```
// fecha actual
DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy");
LocalDateTime now = LocalDateTime.now();
String fecha = dtf.format(now).toString();

for (SensorTemp st : user.getLista_sensor_temp()) {
    if (st.getTemp_n().equals("0")) {
        //si la temperatura de la noche es 0 es porque se ha creado ya un objeto con la temperatura de dia
        isday = false;
    } else if (st.getFecha().equals(fecha)) {
        //si la fecha corresponde a la fecha del dia de hoy, el usuario ya ha registrado su temperatura diaria
        newday = false;
    }
}
```



Familiar

El familiar es capaz de ver toda la información asociada al paciente vinculado al a vez que establecer una comunicación directa con el médico del paciente.

Solo puede ver los datos de las diferentes pestañas y en ningún momento puede modificarlos.

Esto es así para aquellas personas que no puedan valerse por sí mismas o necesiten de una supervisión por parte de una persona cercana a los mismos.

No entraremos a explicar de nuevo las ventanas porque son iguales a las del médico pero sin posibilidad de modificar nada como acabamos de comentar.



Javier Liebana

Perfil Sensor Temperatura

Nombre	Javier
Apellidos	Liebana
DNI	12345678A
Descripción	El paciente es un joven de 22 años invidente, también presenta alergia a múltiples plantas primaverales, cada año debe someterse a un control en dicha época para evitar riesgos.

Refrescar

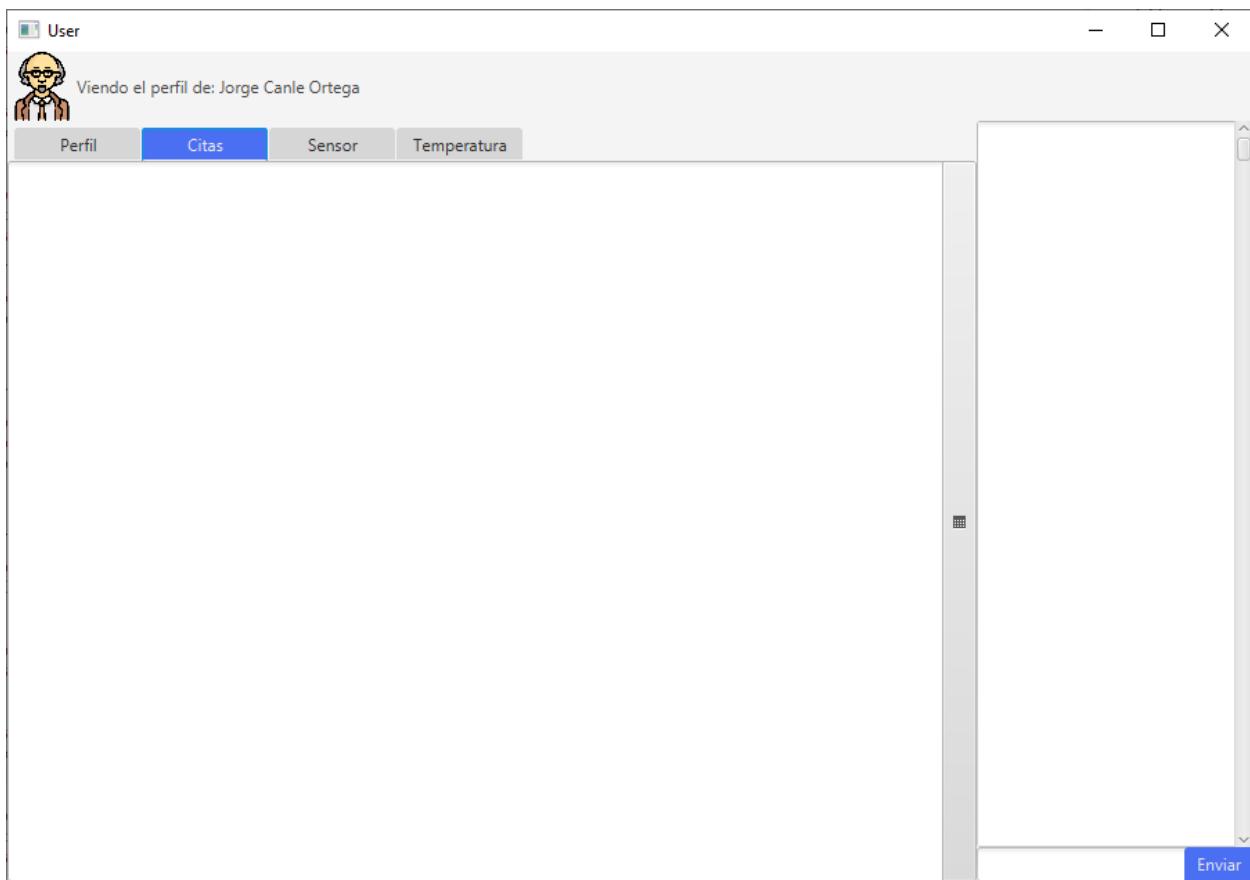
2021-06-19 21:09:03.0
pepe: Hola, buenos días. ¿Cómo se encuentra hoy?
javi: Buenos días doctor, bastante mejor que ayer, como verá la humedad del hogar es la perfecta según me comentó ayer en consulta.

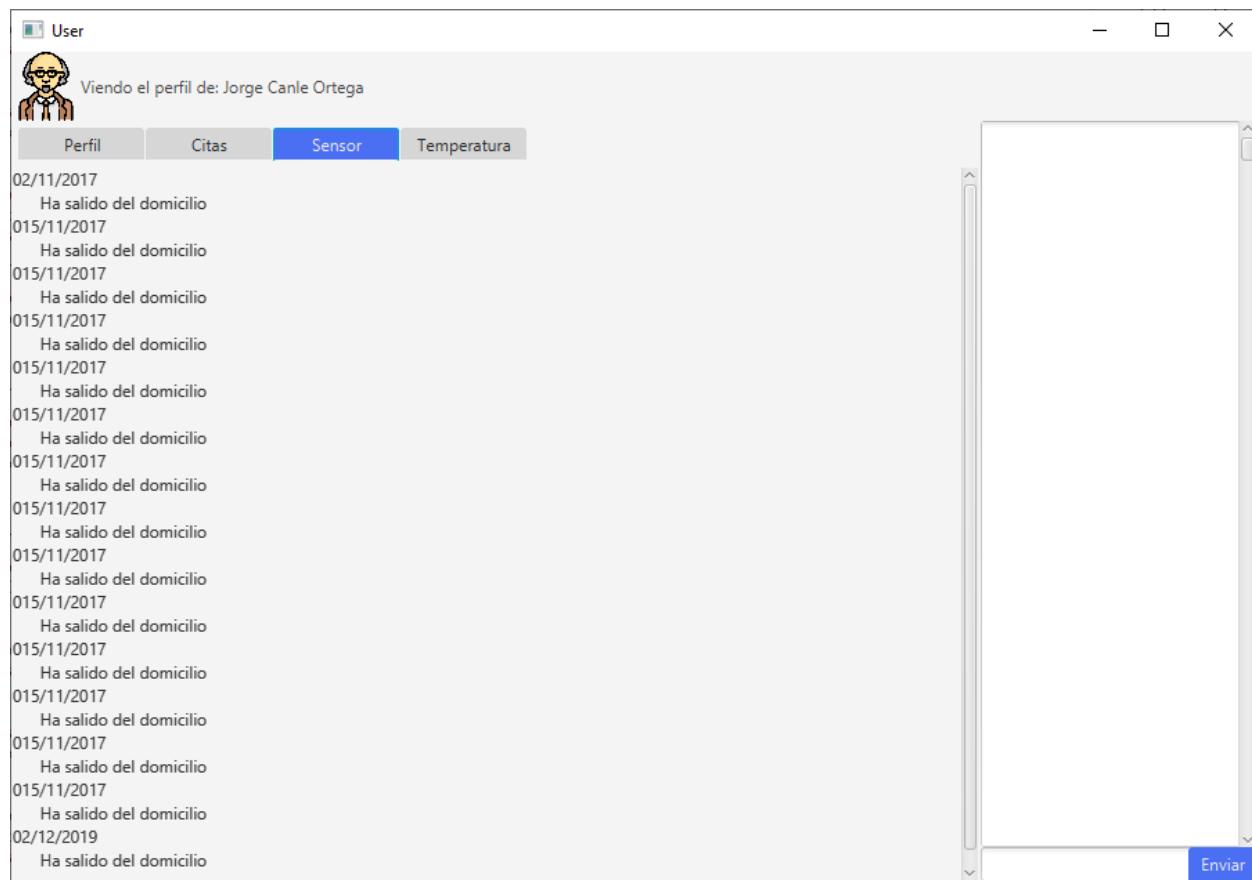
2020

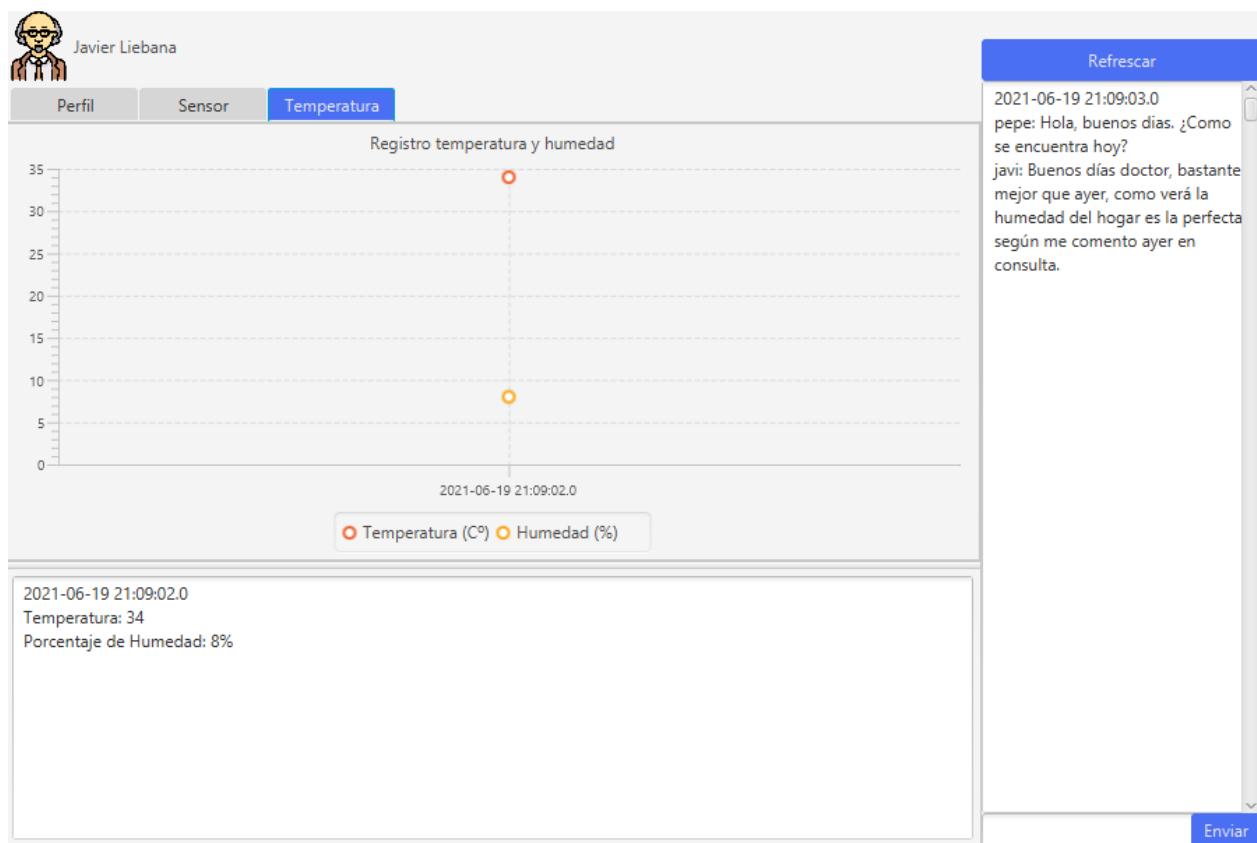
▼ Historial médico 2020-05-19
Se ha instalado la aplicación MedicApp en el hogar del paciente

► Historial médico 2020-05-19
► Historial médico 2020-05-22

Enviar







Resultado y validación

Este trabajo tiene como finalidad la escalabilidad en cuanto a sensores se refiere, ya que permite añadir los necesarios para recoger toda la información necesaria del paciente para así poder llevar un control exhaustivo del mismo.

El resultado ha sido positivo, hemos realizado diversas pruebas tanto de caja negra como blanca y ha pasado todos los controles de manera correcta. También hemos realizado pruebas de funcionalidad con familiares para analizar su nivel de satisfacción con la aplicación, obteniendo buenos resultados.

Implementación

Javier Liébana	Ventana principal médico, user y familia Base de datos
Daniel Santiago	Chat, login, registro y mockup Deserialización/Serialización JSON

Conclusión

Innovación

Este sistema permite la comunicación directa entre médico/paciente al igual el acceso directo a los datos tanto recogidos por un sensor como introducidos por el paciente por parte del médico, siendo esto una manera más rápida y eficaz de realizar una valoración previa del estado del paciente sin necesidad de que este tenga que trasladarse.

Trabajo Futuro

De cara al futuro tenemos dos puntos importantes que analizar:

- Base de datos a tiempo real: Para el correcto funcionamiento de la mayoría de nuestras funcionalidades debemos implementar una base de datos funcional y también nos gustaría que fuese a tiempo real para una posible expansión hacia la industria del móvil.
- Arduino: para el siguiente avance se requiere la instalación de varios sensores para hacer plenamente funcional el registro de datos del usuario (paciente). Debemos primero adquirir los conocimientos de programación con arduino para luego poder enlazar nuestra aplicación con dichos sensores.

Bibliografía y Webgrafía

[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

[https://es.wikipedia.org/wiki/Eclipse_\(software\)](https://es.wikipedia.org/wiki/Eclipse_(software))

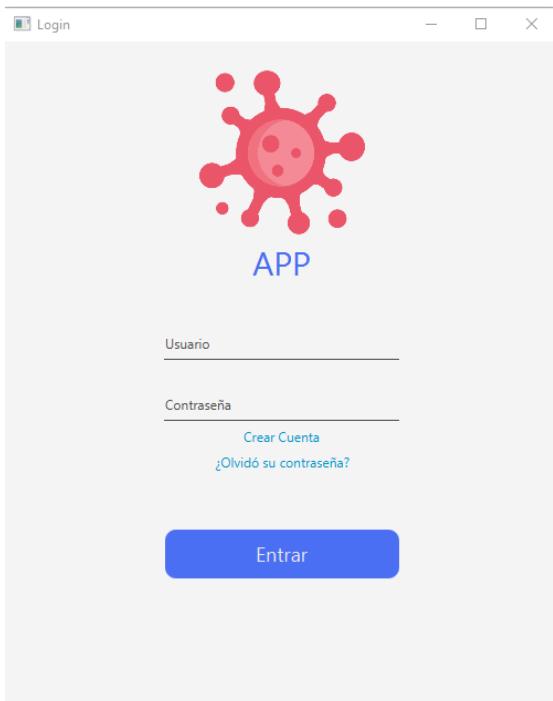
https://es.wikipedia.org/wiki/Arduino_IDE

<https://es.wikipedia.org/wiki/NoSQL>

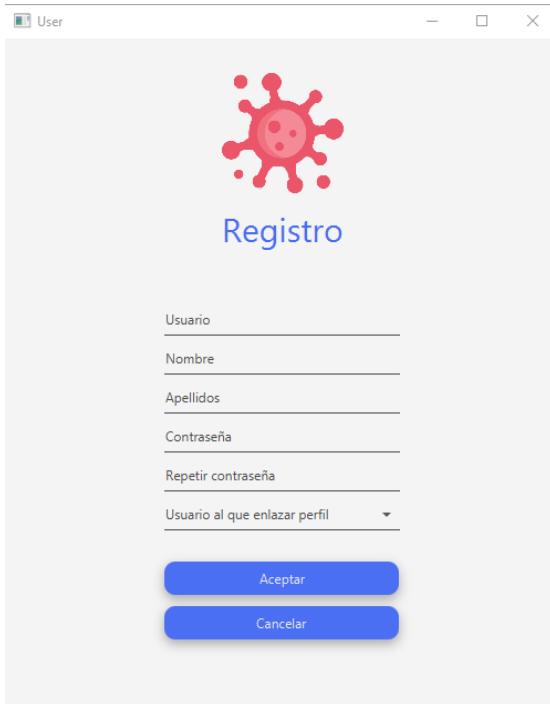
Anexos

Manual de Usuario

1. Ventana principal - Logarse en la aplicación



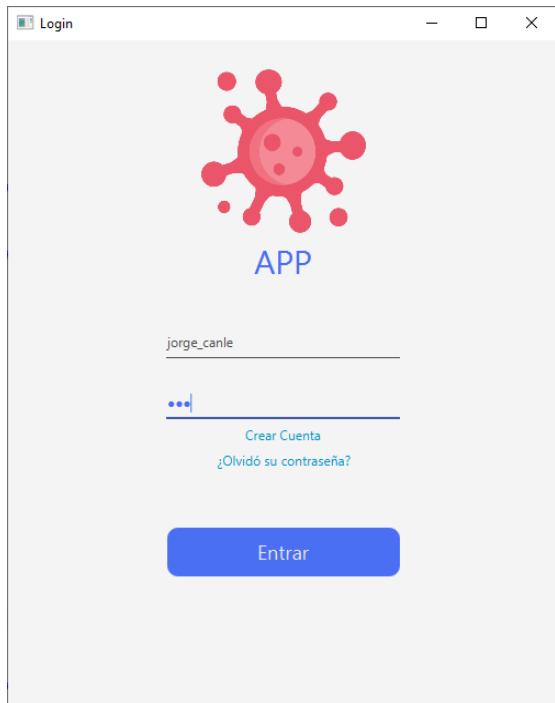
2. Para ello es necesario disponer de una cuenta previamente creada, por lo que pulsamos en Crear Cuenta y nos aparecerá la siguiente ventana:



Introducimos nuestros Datos como son el Nombre y los Apellidos a la vez que una contraseña para poder acceder a la aplicación.

*Si somos un familiar deberemos, en el apartado “Usuario al que enlazar el perfil” seleccionar el usuario familiar que obviamente ha debido crearse una cuenta previamente.

3. Una vez creado volvemos a la ventana de login e introducimos nuestros datos:



4. Ahora ya podremos acceder a nuestra ventana principal con sus pestañas y el chat.
- a. Perfil: Se mostrarán los datos relacionados con el paciente y sus historiales médicos.

The screenshot shows a software window titled "User" with a profile picture of a man and the name "Jorge Canle Ortega". The window has four tabs at the top: "Perfil" (selected), "Citas", "Sensor", and "Temperatura". The "Perfil" tab displays the following information:

Nombre	Jorge
Apellidos	Canle Ortega
DNI	12345678A
Descripción	Paciente masculino de 53 años de edad que concurre al servicio de guardia del hospital Centenario por un cuadro caracterizado por un episodio de hematemesis de escasa intensidad, dos horas previas a la consulta, refiere almuerzo 3 horas antes y que la misma fue precedida de náuseas. Tres meses anteriores al vómito sanguinolento el paciente refiere dolor abdominal de intensidad creciente con el correr de los meses, de carácter quemante y opresivo en la región epigástrica que se irradia hacia hipocondrio izquierdo y espalda y que se alivia con la ingesta de alcalinos o antiácidos comerciales. Refiere además distensión abdominal del mismo tiempo de evolución. Niega fiebre, diarrea, constipación, disfagia, odinofagia, fiebre. Como antecedente de jerarquía el paciente refiere ser bebedor de 1 ½ litros de vino tinto por día desde hace más de 30 años

Below the description, there is a timeline with buttons for 2017, 2018, 2021, and 2022. Under 2017, there are three entries:

- Historial médico 12/11/2017
- Historial médico 13/11/2017
- Historial médico 14/11/2017

In the bottom right corner of the main panel, there is a blue "Enviar" button.

En el caso del médico dispondremos de un botón que nos permitirá añadir los diferentes historiales médicos del paciente.

The screenshot shows a medical software window titled "User". At the top, there is a header bar with a user icon and the name "Jorge Canle Ortega". Below the header, there is a navigation bar with tabs: "Perfil" (selected), "Citas", "Sensor", "Temperatura", and "Familiares".

The "Perfil" tab contains the following information:

- Nombre: Jorge
- Apellidos: Canle Ortega
- DNI: 12345678A
- Descripción: que la misma fue precedida de náuseas.
Tres meses anteriores al vómito sanguinolento el paciente refiere dolor abdominal de intensidad creciente con el correr de los meses, de carácter quemante y opresivo en la región epigástrica que se irradia hacia hipocondrio izquierdo y espalda y que se alivia con la ingesta de alcalinos o antiácidos comerciales.
Refiere además distensión abdominal del mismo tiempo de evolución.
Niega fiebre, diarrea, constipación, disfagia, odinofagia, fiebre.
Como antecedente de jerarquía el paciente refiere ser bebedor de 1 ½ litros de vino tinto por día desde hace más de 30 años

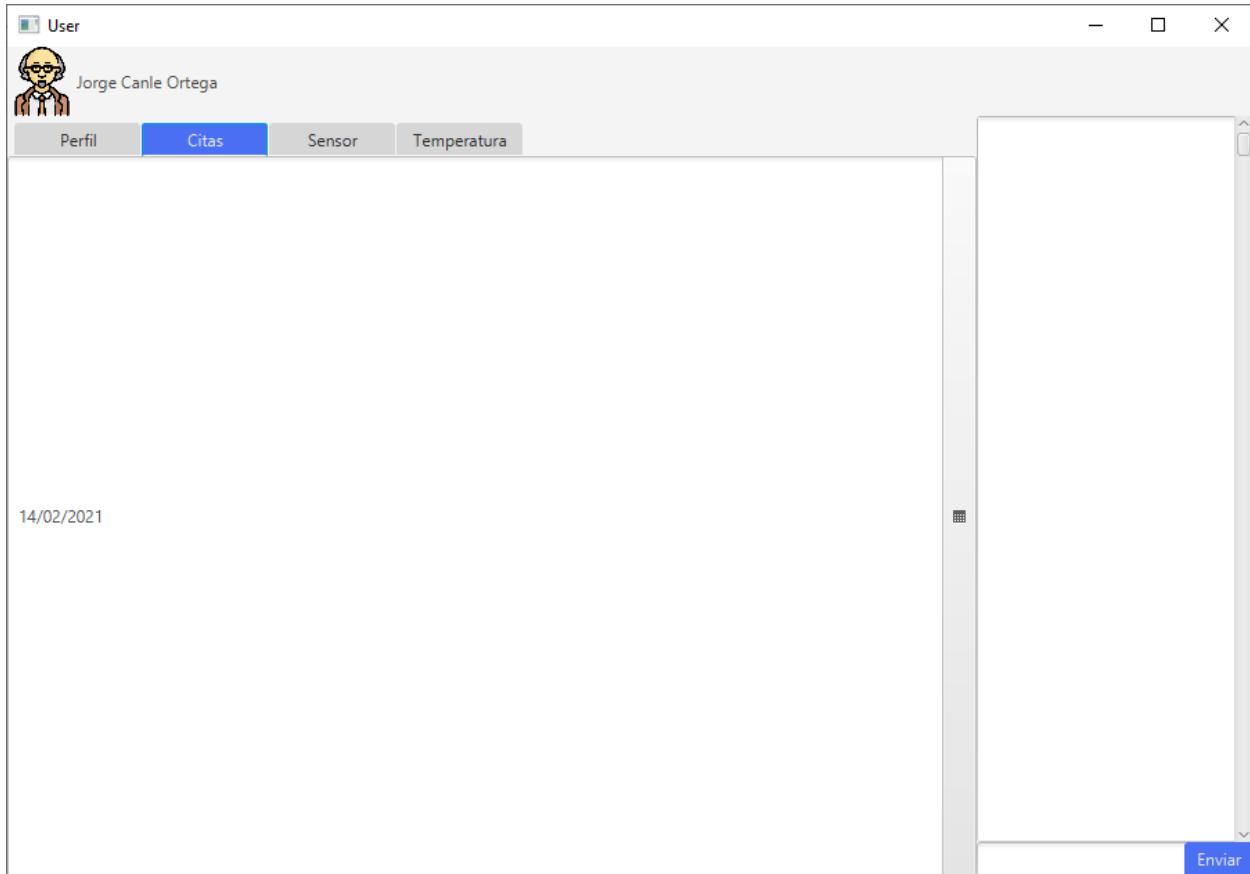
Below the description is a blue button labeled "Añadir Hist. Médica".

At the bottom of the main panel, there is a timeline with years: 2017, 2018, 2021, 2022. To the right of the timeline, there is a list of historical entries:

- Historial médico 12/11/2017
- Historial médico 13/11/2017
- Historial médico 14/11/2017

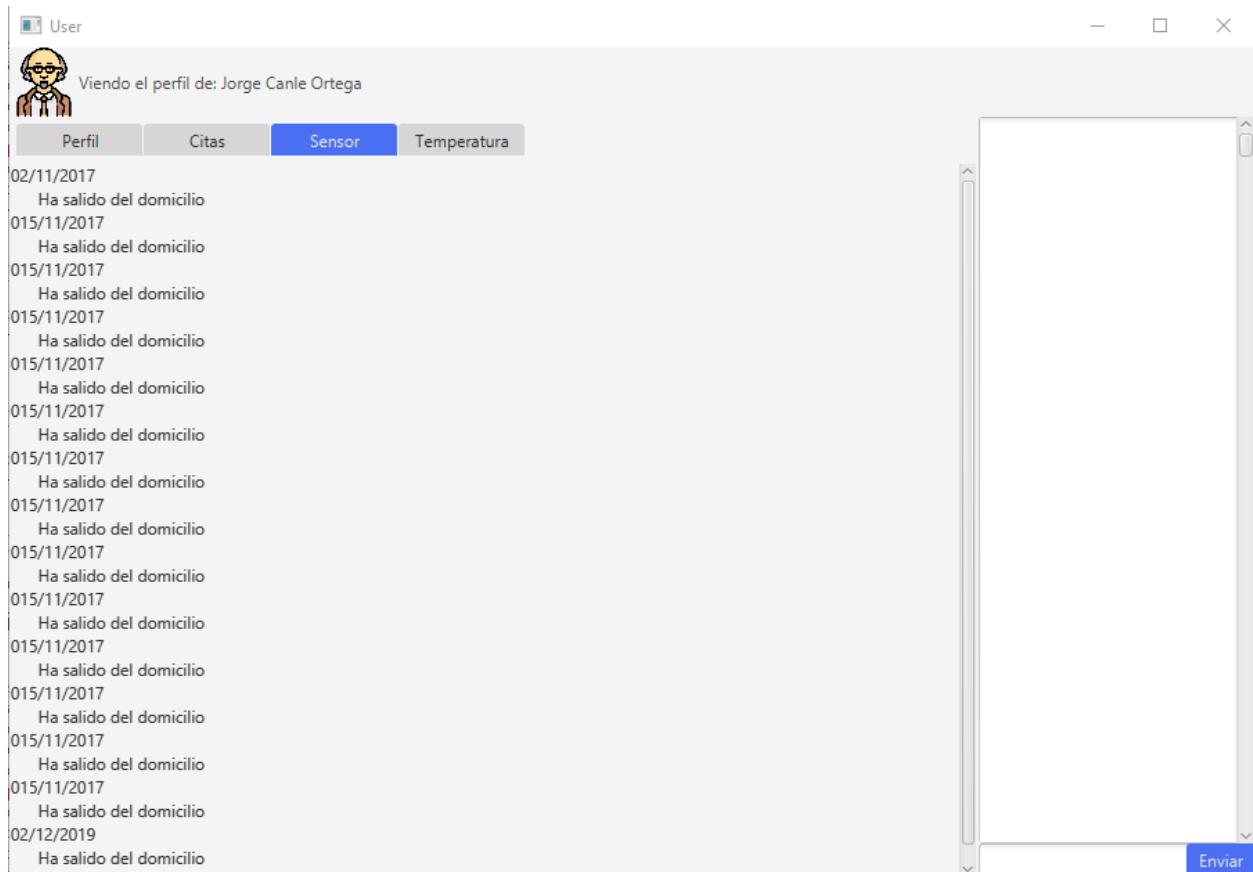
On the far right, there is a blue "Enviar" button.

b. Citas: Permite al usuario agendar una cita con su médico.

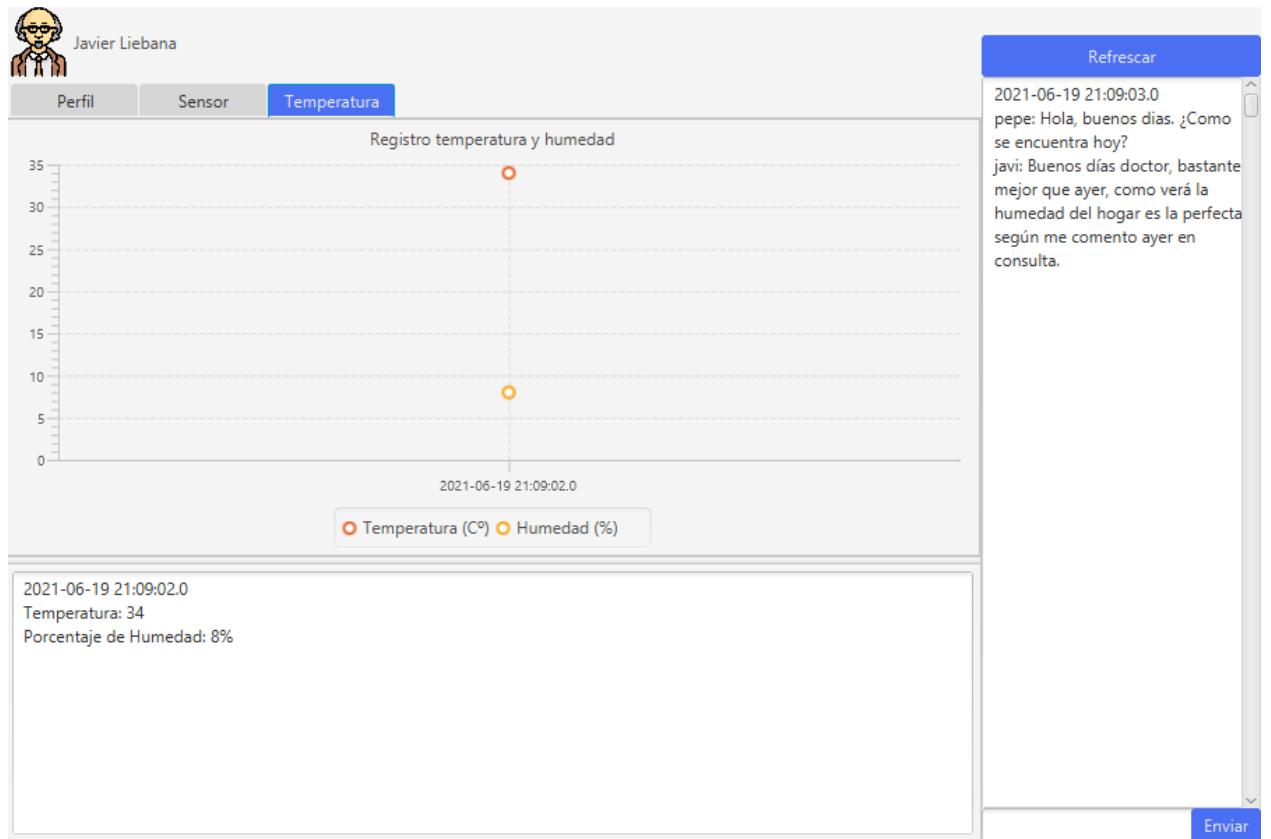


En el caso del médico le permite ver su agenda del mes planificada.

- c. Sensor: Permite ver los datos recogidos por un sensor en específico. Es el mismo para los tres tipos de usuarios (médico, paciente, familiar).

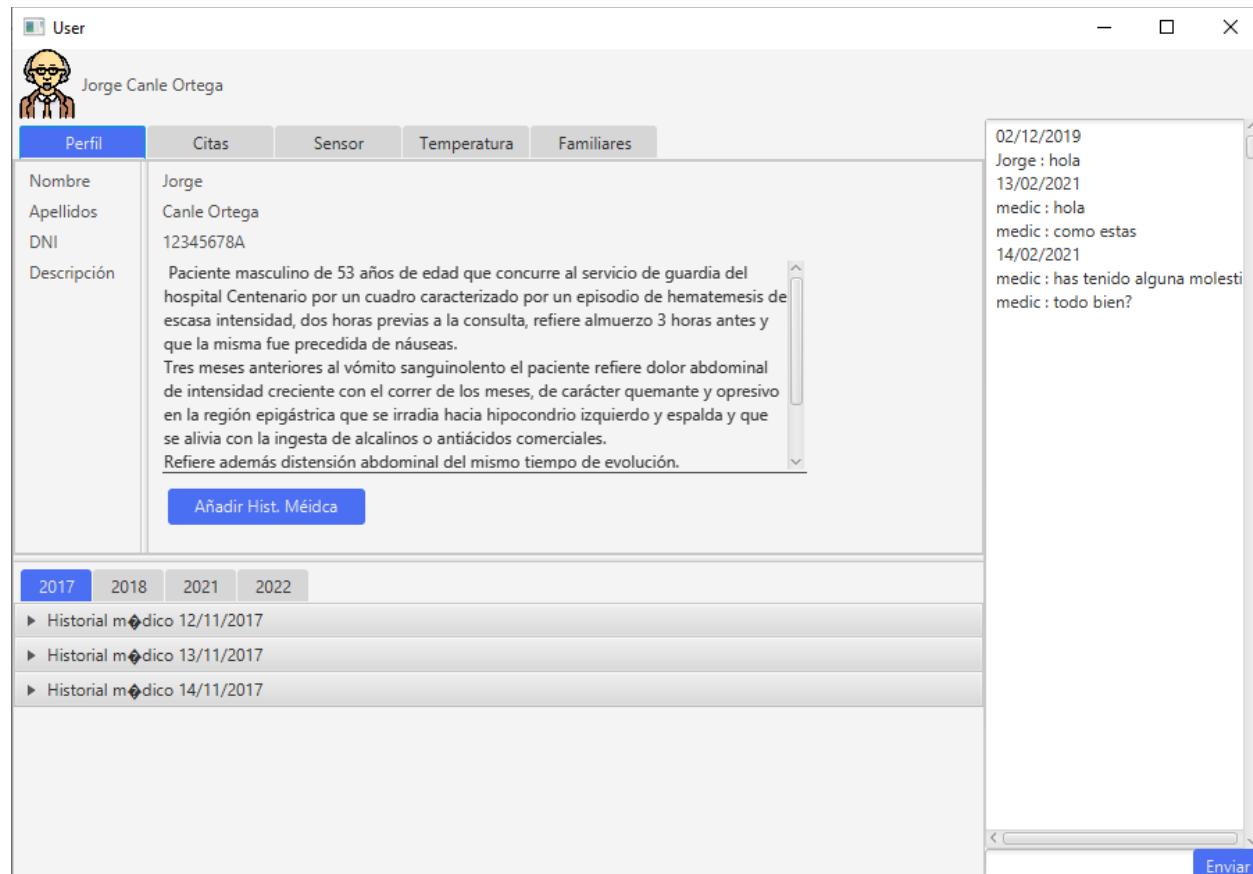


- d. Temperatura: Esta pestaña permite tener un control de la temperatura y humedad diario dentro del hogar. Estos datos se renuevan cada 30 minutos.



5. Finalmente dispone de un chat directo con el médico asignado donde quedan registradas las conversaciones.

Para hacer uso de este chat solo hay que escribir en la caja blanca situada en la esquina inferior derecha y una vez se termine de escribir pulsar enviar para que el texto quede guardado y pendiente de que el médico lo lea.



The screenshot shows a medical software interface. At the top, there's a header bar with a logo and the word "User". Below it, a patient profile is displayed for "Jorge Canle Ortega", featuring a portrait icon. The profile includes fields for Nombre (Jorge), Apellidos (Canle Ortega), and DNI (12345678A). A detailed medical history is listed under Descripción, mentioning symptoms like hematemesis, abdominal pain, and distension. Below the profile, a timeline shows historical medical records for 2017, 2018, 2021, and 2022. On the right side, a messaging window is open, showing a conversation between the patient ("Jorge") and the doctor ("medic"). The messages are:

- 02/12/2019 Jorge : hola
- 13/02/2021 medic : hola
- medic : como estas
- 14/02/2021 medic : has tenido alguna molestia?
- medic : todo bien?

A blue "Enviar" button is located at the bottom right of the messaging window.

