



CENSALUD

Aplicación gestión de citas en centros de salud

Proyecto DAW

Ana Isabel Cuellar Maestro

INDICE:

● Anteproyecto:	
○ Título	2
○ Antecedentes	2
○ Definición del problema a solucionar	2
○ Justificación del proyecto	2
○ Objetivos y alcance	2
○ Procedimiento y tecnologías a usar	6
○ Presupuesto	6
○ Product Backlog	7
○ Sprint 2 Backlog	8
○ Descripción del trabajo en Sprint 2	8
○ Documentación incremento 1	9
○ Retrospectiva Sprint 2	15
○ Sprint 3 Backlog	15
○ Descripción del trabajo en Sprint 3	16
○ Documentación incremento 2	27
○ Retrospectiva Sprint 3	43
○ Sprint 4 Backlog	43
○ Descripción del trabajo en Sprint 4	44
○ Documentación incremento 4	52
○ Retrospectiva Sprint 4	60
○ Manual de uso y puesta en marcha	60
○ Conclusiones globales	76
○ Webgrafía	77

Anteproyecto

PUNTOS DEL ANTEPROYECTO:

1. Título

CenSalud. Gestión de citas en un centro de salud.

2. Antecedentes

La tecnología avanza cada vez más rápido, todo está cada vez más automatizado, en general la gente joven se suele manejar bien con las tecnologías, pero la generación de personas mayores les cuesta más ya que es una cosa nueva que tienen que aprender a su edad, esta página se enfocará en ser más intuitiva y que cualquier persona pueda manejarla sin dificultad alguna.

3. Definición del problema a solucionar

La aplicación será muy intuitiva y fácil de usar para todos con ella podrán pedir cita en el medico, consultar sus informes en el momento que lo necesite, además de si tiene familiares a cargo poder gestionar sus citas e informes.

Además, pueden pedir cita en el mismo centro, la recepcionista podrá acceder al portal del usuario para darle cita o mostrar sus informes.

A la misma vez si en la consulta con el especialista, necesita venir a otra consulta posterior, este podrá dar citas al usuario, añadir sus informes, etc.

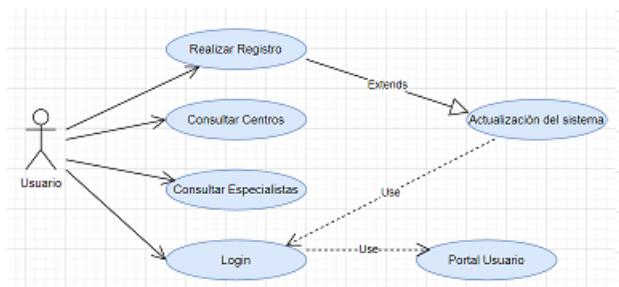
4. Justificación del proyecto

En los últimos tiempos está teniendo un aumento muy considerado los seguros de salud privados y como ya comentaba anteriormente, la tecnología avanza muy deprisa y todo se está automatizando, con esta aplicación los usuarios podrán acceder a la aplicación en cualquier momento que lo necesite y obtener toda su información.

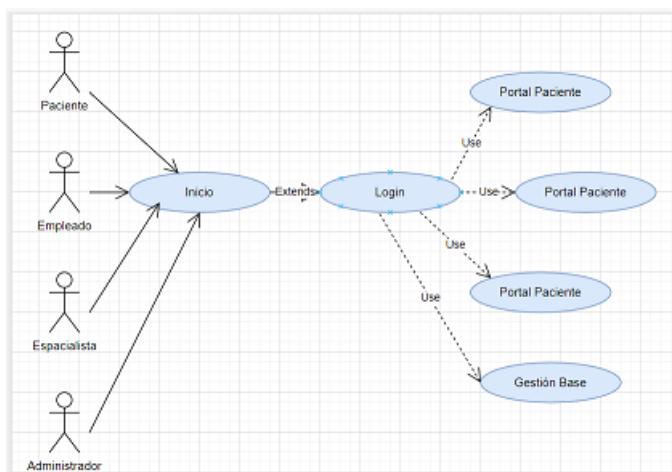
5. Objetivos y alcance (Tener en cuenta los plazos)

- Esta aplicación tendrá una página de inicio, en la cual el usuario podrá registrarse como nuevo usuario o si ya está registrado le llevará a un login para verificar que se encuentra registrado.

Aunque el usuario no este registrado, este podrá ver todos los centros que hay y los especialistas que trabajan en cada centro.

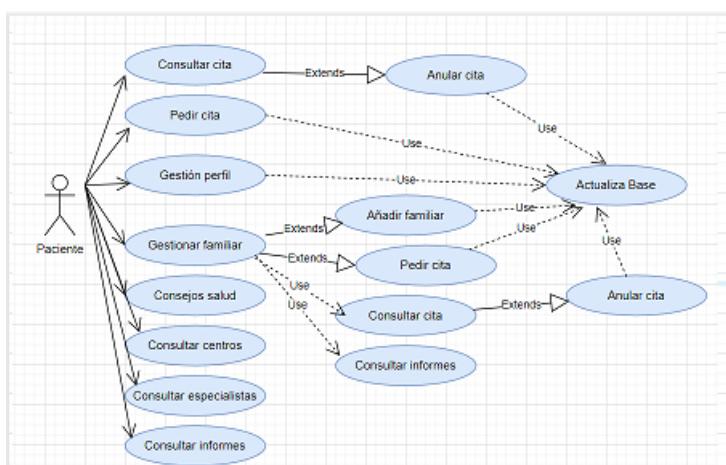


- Una vez se verifica al usuario, el programa verá que tipo de Rol tiene ya que esta aplicación contiene cuatro roles (Paciente, Especialista, Empleado y Administrador de BD).



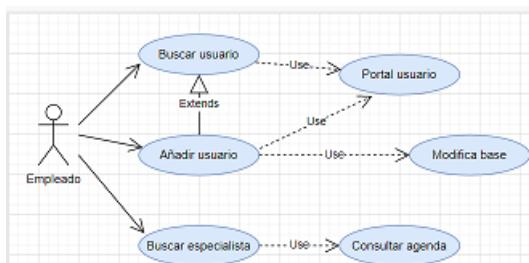
- Rol 1: Paciente entrará al portal del usuario.

- * Podrá pedir cita.
- * Ver si tiene citas pendientes y anularlas en caso de que sea necesario.
- * Ver sus historiales médicos.
- * Si tiene familiares a cargo, podrá gestionar sus citas, ver sus informes y sus historiales médicos, además de poder añadir más familiares.
- * Ver consejos de salud.
- * Gestionar su perfil.
- * Ver centros hospitalarios.
- * Ver especialistas.



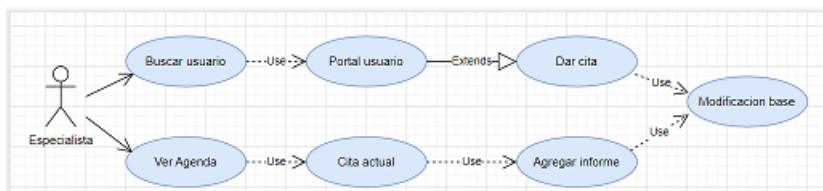
- Rol 2: Empleado entrará al portal del Empleado.

- * Buscar al usuario que le llevará al portal de este y desde hay podrá gestionar los datos del paciente.
- * Buscar especialista Consultar su agenda de los especialistas y ver que huecos libres tiene.
- * Añadir usuarios.



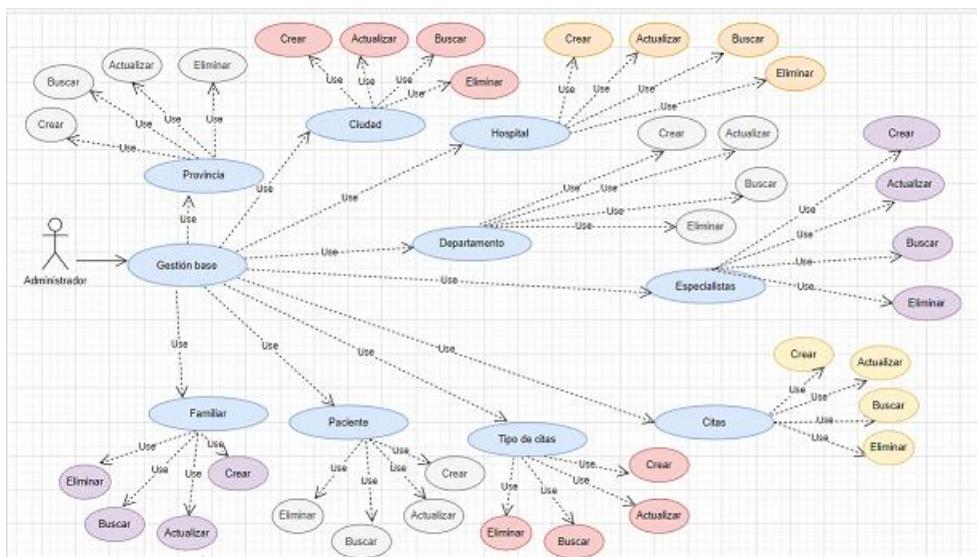
- Rol 3: Especialista entrará al portal del Especialista.

- * Buscar al usuario, que le llevará al portal del usuario donde podrá gestionar.
- * Ver su agenda de citas.
- * Desde la cita del usuario, añadirá el informe de la consulta.



- Rol 4: Administrador BD entrará al portal del Administrador.

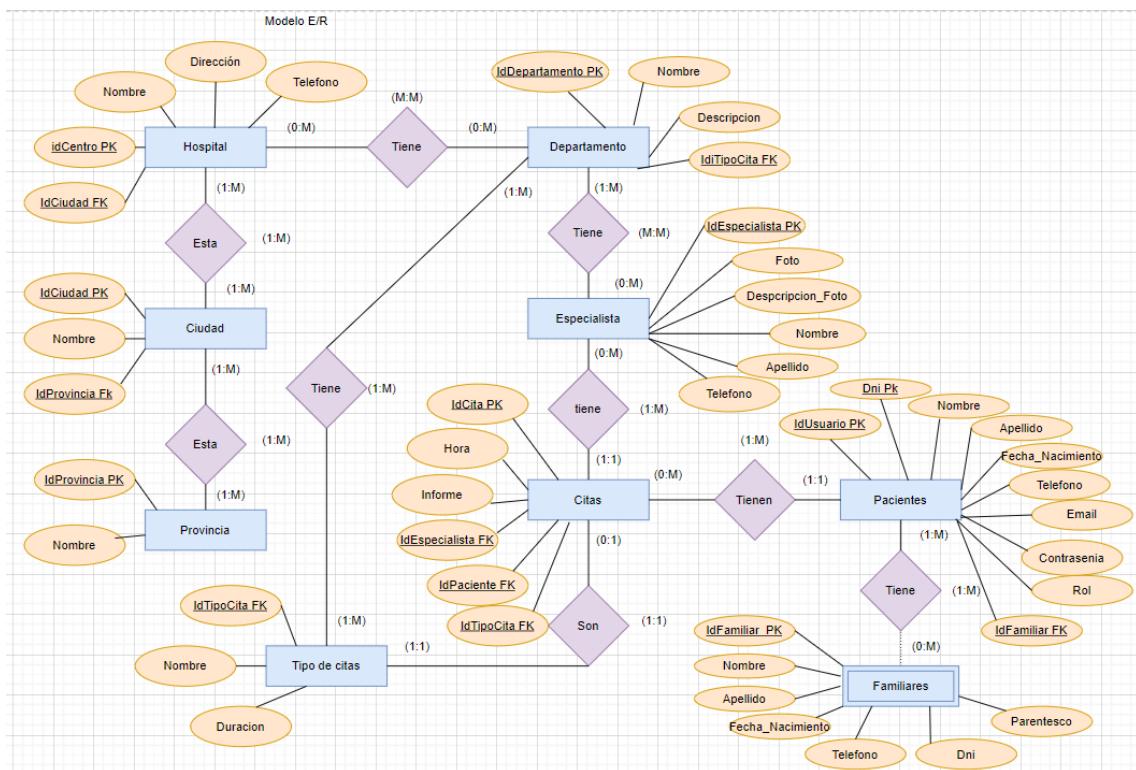
- * Podrá gestionar toda la base de datos.
- * En cada tabla podrá añadir, consultar, eliminar, modificar.



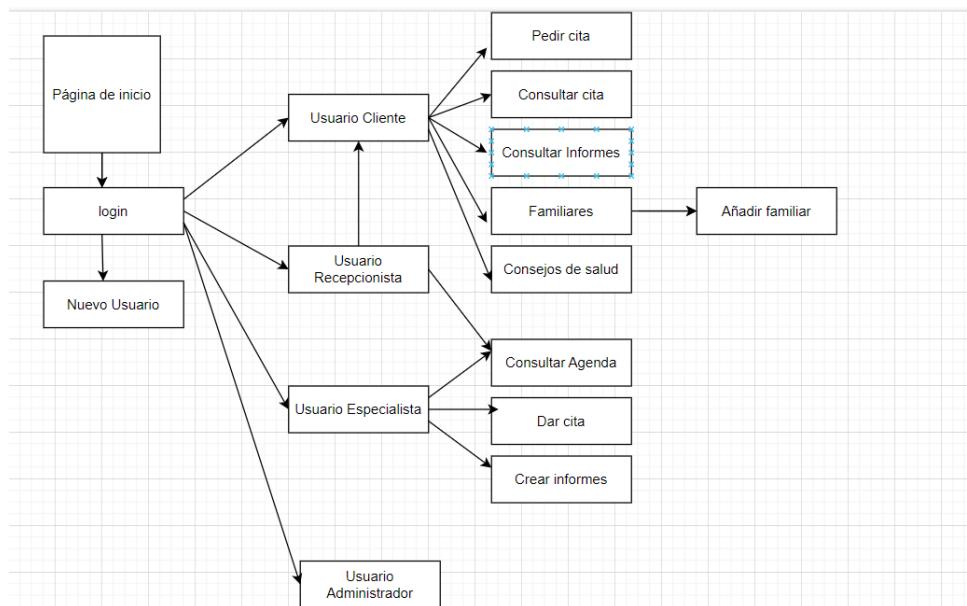
Con la metodología Scrum:



Modelo



Vistas



6. Procedimiento y tecnologías a usar (las tecnologías deberán estar de entre las estudiadas en el ciclo)

- Visual Studio Code, para la realización del código.
- MySQL Workbench, para la realización del modelo.
- Bootstrap Studio, para hacer la aplicación responsive.
- Paint, realización de imágenes.
- Adobe Express, realizar logos.
- Image Resizer, para ajustar el tamaño de las imágenes.
- Slider de Google, plantillas para imágenes.
- DALL-E para generar imágenes.

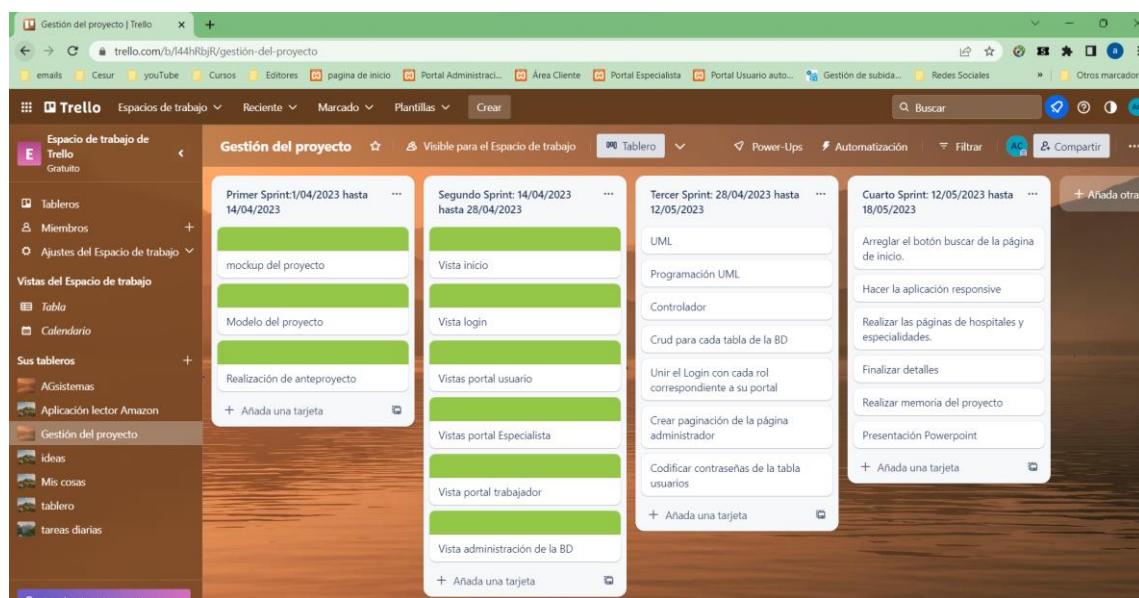
7. Presupuesto (en horas)

TAREAS	HORAS
Mockup	1
Modelo del proyecto	1
Vistas	4
UML	2
Programación UML	14
Realización del controlador	13
Finalización de detalles	1
Memoria del proyecto	3
Presentación PowerPoint	1
TOTAL	40

8. Product Backlog

Requisitos	Descripción
Fundamentales:	
Responsive	La aplicación debe de ser responsive, que se pueda usar desde cualquier dispositivo.
Fácil Manejo	Cualquier usuario debe ser capaz de manejar y entender cada funcionalidad que puede realizar en la aplicación.
Login	Realización de login y que distinga cada Rol de usuario.
Crear nuevo usuario	Debe ser capaz de crear nuevos usuarios, para visitar las páginas
Modificar datos	El usuario puede modificar datos en cualquier momento determinado.
Portal del usuario	Se trata de una aplicación para centros hospitalarios por lo que es requisito fundamental que el usuario pueda gestionar sus citas, consultas e informes.
Portal del personal autorizado	Es importante que pueda gestionar los datos de un paciente, en el mismo instante requerido por este y mediante el DNI facilitado por el paciente acceder a ellos.
Portas Especialista	La principal función que debe tener este portal es poder crear los informes clínicos, dar citas sería secundario ya que desde la aplicación o en el mismo centro el paciente puede obtenerla.
Portal Administrador	Debe funcionar correctamente, ya que es el cerebro de toda la aplicación, desde aquí se asignará los roles de personal autorizado y especialista.
Rol	Es fundamental ya que determinará a que portal irá cada usuario.
Mejoras futuras	
Vistas	Refactorización de las vistas con templates, para una mejor limpieza del código.
Página inicio	Crear las vistas para la página “Centros” y “Especialistas”
Portal del especialista	Al seleccionar un nombre de la tabla, este le aparezca directamente en el contenedor crear informe.
Recuperación de contraseña	En este momento aparece reflejado en la aplicación, debido al corto tiempo de desarrollo, si no da tiempo quedaría para una posible mejora.

9. Sprint 2 Backlog



En el siguiente Sprint se priorizará la programación con PHP para la conexión con la BD y la realización del CRUD en cada una de las tablas.

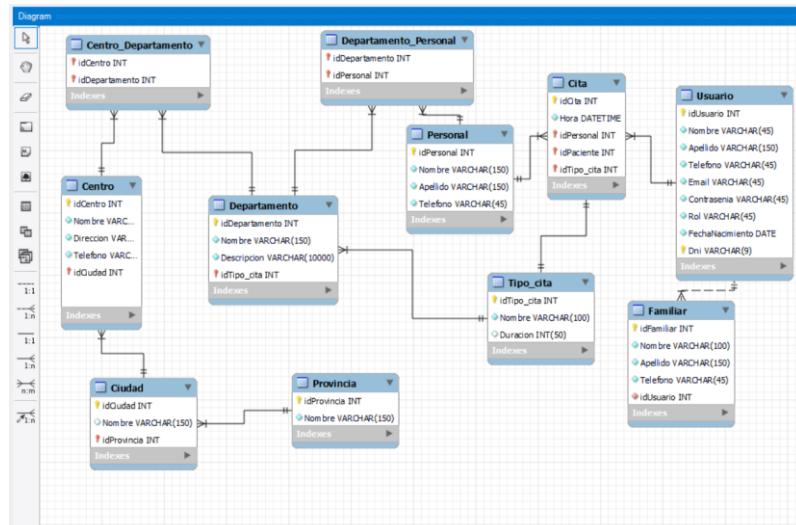
Por lo que priorizará la realización del “Portal Administrador” una vez realizado este portal, para buscar la información que se requiere desde los otros portales de la aplicación, solo sería realizar llamadas a la BD.

10. Descripción del trabajo en Sprint 2

En este segundo Sprint se han realizado todas las vistas que tendrá la aplicación, a falta de las páginas hospitalares y especialistas.

Las vistas se han realizado usando Bootstrap, JavaScript, HTML, CSS.

También la creación de la BD, para poder empezar a trabajar con ella, el Modelo relacional de la Base de datos.



Para realizar las pruebas de login solo admite el usuario “prueba” contraseña “prueba”.

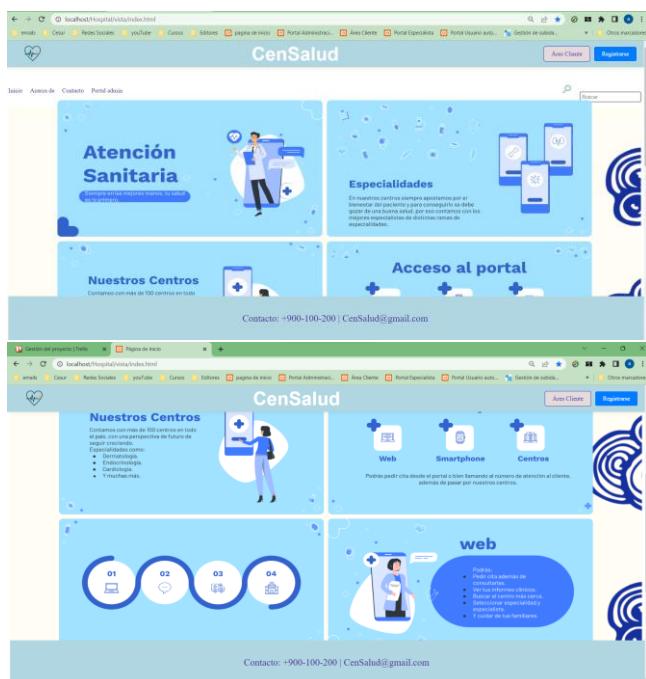
En el siguiente Sprint será una tarea priorizable la búsqueda de usuarios para comprobar el login.

En el siguiente punto de la documentación se detalla cada funcionalidad ya realizada en este Sprint para cada una de la página.

11. Documentación incremento 1

- Página de inicio:

La página principal de la aplicación contiene información de los centros hospitalarios en ‘Acerca de’, la cabecera y la barra de navegación será igual para todas las páginas de la aplicación, el pie de página contiene la información de como contactar con atención al cliente, este se queda fijo mientras baja y sube la página.



Funcionalidades:

- Icôno: Lleva a la página de inicio.
- Área Cliente: El usuario accede al portal correspondiente, según el rol que tenga asignado. Por ahora solo accede al de administrador y ya se puede navegar por las diferentes áreas privadas.



El login contiene el botón volver que lleva al usuario a la página de inicio, como se puede ver el logo se encuentra en todas las páginas de la aplicación y también lleva a la página de inicio.

- Registrarse: Registra un nuevo usuario, el usuario registrado mediante este formulario tendrá asignado el rol de usuario.



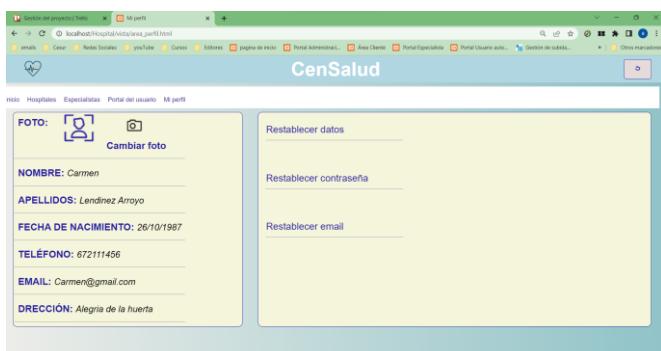
- Inicio: Lleva a la página de inicio.
- Acerca de: Lleva a una página que da información de los centros.
- Contacto: Te pone en contacto con el centro.
- Buscar: Realiza una búsqueda rápida de Centros en una provincia determinada.

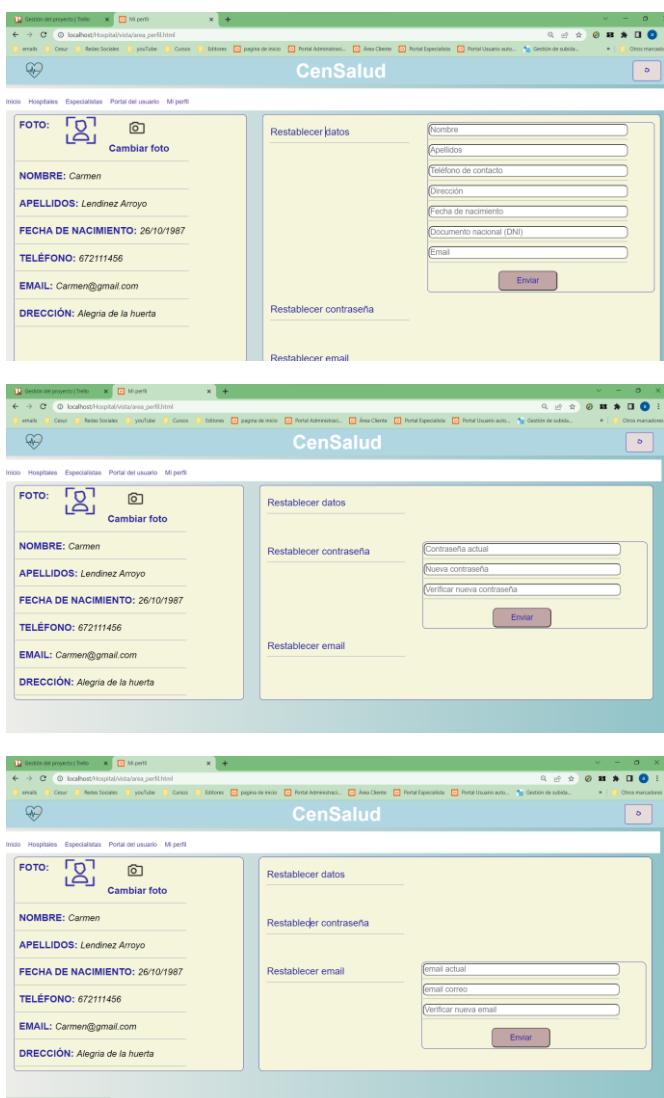
Según el Rol asignado a cada usuario:

- Área Cliente:



Empezando por la cabecera, en el botón de mi perfil, el cliente puede acceder a consultar o modificar sus datos, en el panel izquierdo de la página aparecen los datos con los que está registrado en el derecho, cada enlace despliega un formulario donde el paciente puede modificar.





El botón enviar no se accionará si falta algún campo por rellenar de dicho formulario que se quiere modificar.

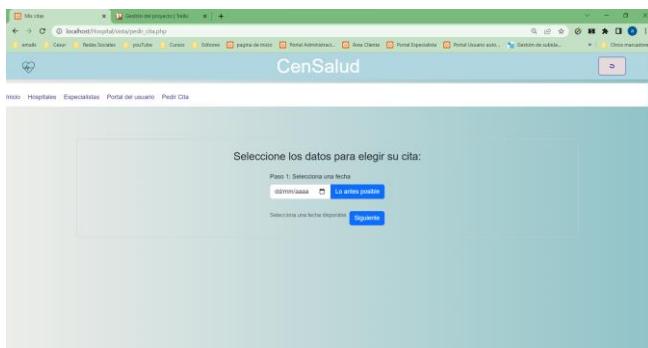
Como se puede apreciar, también se encuentra el botón que lleva a la página principal del área de cliente.

En la barra de navegación de

Cerrar sesión, cierra automáticamente y lo devuelve a la página de inicio.

- En la barra de navegación:
 - Inicio: Página de inicio.
 - Hospitales: Aparece una lista de todos los centros.
 - Especialidades: Aparece todos los especialistas que trabajan en los centros.
 - Portal de usuario: Indica que está en el portal del usuario y recarga la página.
- En la cabecera del cuerpo un mensaje de bienvenida con el nombre del usuario.
- Etiquetas para con las que puede interactuar el usuario:

- Pedir cita, desde aquí el usuario podrá pedir cita en cualquier centro mediante cuatro pasos:
 1. Seleccionar una fecha.
 2. Seleccionar una Especialidad.
 3. Seleccionar localidad y centro.
 4. Ofrece la cita y en caso de no venir bien al usuario puede buscar otra cita.



- Mis citas, en esta página se mostrará todas las citas que el usuario tenga programadas.
En la cabecera le dirá el número de citas que tiene y en el cuerpo mostrará el detalle de cada cita.



- Historial, El usuario podrá consultar sus informes clínicos que el especialista le ha redactado en cada consulta.



- Familiares, en esta página el usuario podrá gestionar a los familiares que tenga a su cargo.
En la parte izquierda de la página saldrán los el nombre del familiar adjunto a dicho usuario, donde podrá actualizar sus datos, que le redirige a un formulario.

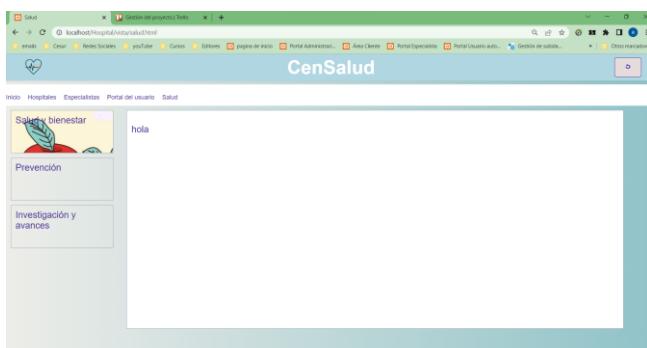
The screenshot shows a web-based form titled "CenSalud". It contains several input fields for entering family information: Apellidos, Teléfono de contacto, Dirección, Vinculo familiar, Fecha de nacimiento, and nacional (DNI). Below the fields is a red "Actualizar" button.

Ver su historial (redirige a la página informes clínicos), bien eliminar familiar o bien añadir familiar, que redirige al formulario anteriormente mencionado para modificar los datos.

En la parte derecha, las etiquetas pedir cita (redirige a la página pedir cita), citas próximas (redirige a la página “Mis citas”) e informes (redirige a la página “Historial”).



- Salud, esta página muestra consejos de salud.



- **Portal personal autorizado:** En este portal el personal trabajador del centro podrá:
 - Dar cita al paciente mediante el DNI, le redirige a la página dar cita.
 - Consultas del paciente, se abre un desplegable y mediante el DNI le redirige a la página que desee consultar.

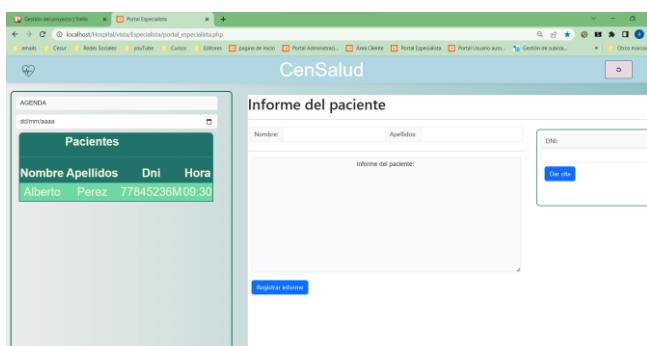
The screenshot shows a dropdown menu titled "Paciente". It includes a search bar for "Dar cita para el paciente" and "NIF paciente". Below is a dropdown menu with the following options: "Consultar sobre el paciente", "Consultar citas del paciente", "Consultar Informes del paciente", and "Consultar familiares del paciente". The "Consultar sobre el paciente" option is highlighted with a blue background.

- Consultar los especialistas, centros y consultar agenda de los especialistas.
- En el contenedor de información mostrará la información seleccionada por el personal autorizado.



- **Portal especialista:** En este portal el especialista puede consultar su agenda en la parte izquierda de la página, seleccionará una fecha y le mostrará las citas que tiene en ese día en concreto.

En la parte derecha de la página redactará el informe del paciente y si necesita volver a revisión mediante el DNI del paciente le podrá dar cita (redirige a la página pedir cita).



- **Portal Administrador:** Desde este portal, el administrador tendrá el control absoluto de la base de datos, podrá añadir, eliminar, leer y modificar en todas las tablas.

Desde aquí dará el rol a cada usuario.

En la parte izquierda aparecen las tablas de la base de datos, según la tabla que seleccione, en el contenedor de la derecha podrá realizar cualquier cosa en dicha tabla.



12. Retrospectiva Sprint 2

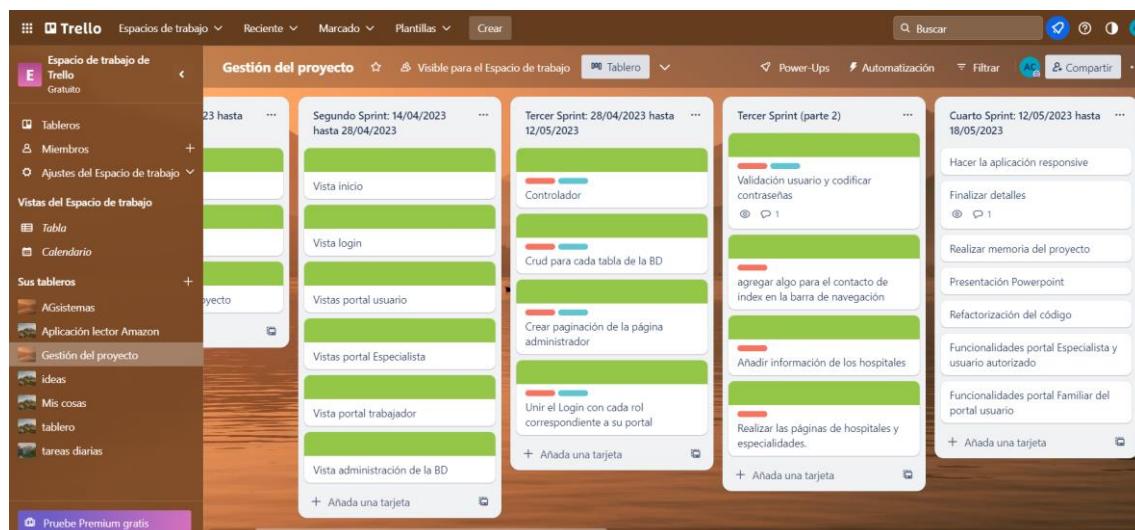
En este sprint he realizado las vistas, usando el modelo vista controlador, mi idea es tratar de enlazar cada función con el CRUD por lo que mi pensamiento no es realizar un CRUD para cada tabla, si no realizar un patrón donde por parámetros entre las tablas y campo para la búsqueda en la Base de datos.

He iniciado con la tabla usuarios, ya que no tiene ninguna FK asociada y es la más sencilla para realizar el CRUD, una vez realizada todas las demás tablas se realizarán de la misma forma, a excepción de las FK que tengan enlazadas. También he empezado los CRUD para ciudad, provincia, etc.

Esto me está llevando más tiempo del que pensaba, pero creo que una vez realizado, las funcionalidades restantes serán más rápidas de desarrollar.

Las vistas podrían haberlas realizado con templates y llamarlas con código PHP usando la función file_get_contents () e imprimiendo el html, de esta forma el código quedaría más limpio, pero es una cosa nueva que estoy aprendiendo a usar en las prácticas y sería una posible mejora para el futuro.

13. Sprint 3 Backlog



En este Sprint se ha priorizado el Back-End de la aplicación, empezando por el mantenimiento de la Base de datos que se realiza a través del portal del administrador.

Se ha realizado la implementación del login con la base de datos, donde según el rol de cada usuario le llevará a su portal.

Se ha realizado las funcionalidades del portal del Administrador, Paciente (usuario) y página de inicio.

Además, para la página de inicio se ha añadido la página "acerca de" da información de cómo y porque surge estos centros hospitalarios, la página de contacto que según la ciudad introducida proporciona el número de teléfono, la página hospitales que muestra fotos del centro con la dirección y la página especialistas que muestra a los especialistas y las especialidades que tienen cada uno.

14. Descripción del trabajo en Sprint 3

En primer lugar, he realizado un patrón para el CRUD.

El patrón del CRUD se encuentra en el modelo y contiene seis funciones:

```
function crud_get_all($tabla) {
```

Esta función realiza la consulta Select de toda la tabla que entra por parámetro.

```
function crud_get_all($tabla) {//consulta toda la tabla
    try {
        // Conexión base de datos
        //require_once("Conexion_BD.php");
        $dsn = 'mysql:host=localhost;dbname=hospital';
        $username = 'root';
        $password = '';
        $pdo = new PDO($dsn, $username, $password);
        // Crear una consulta dinámica utilizando los parámetros proporcionados
        $sql = "SELECT * FROM $tabla"; //realiza la consulta
        $stmt = $pdo->prepare($sql); //prepara la consulta
        $stmt->execute(); //ejecuta

        // Retornar los resultados
        return $stmt->fetchAll(PDO::FETCH_ASSOC); //devuelve todos los registros
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
}
```

```
function crud_select($tabla, $campo, $condicion) {
```

la función crud_select() realiza lo mismo que la función anterior pero en este caso realizando la condición, por lo que la búsqueda se reduce más, por ejemplo se pueden realizar consultas y buscar por id.

La tabla, el campo y la condición entran por parámetro.

```
function crud_select($tabla, $campo, $condicion) {//busca por condición
    try {
        // Conectar a la base de datos utilizando PDO
        $dsn = 'mysql:host=localhost;dbname=hospital';
        $username = 'root';
        $password = '';
        $pdo = new PDO($dsn, $username, $password);
        // Crear una consulta dinámica utilizando los parámetros proporcionados
        $sql = "SELECT * FROM $tabla WHERE $campo = $condicion"; //realiza la consulta
        $stmt = $pdo->prepare($sql); //prepara la consulta
        $stmt->execute(); //ejecuta

        // Retornar los resultados
        return $stmt->fetchAll(PDO::FETCH_ASSOC); //devuelve todos los registros
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
}
```

```
function crud_insertar($tabla, $campos) {
```

Esta función realiza la acción insertar en la BD, por parámetros entran la tabla en la que se realizará la inserción del registro y sus determinados campos.

```

function crud_insertar($tabla, $campos) {//inserta
try {
    // Conectar a la base de datos utilizando PDO
    $dsn = 'mysql:host=localhost;dbname=hospital';
    $username = 'root';
    $password = '';
    $pdo = new PDO($dsn, $username, $password);

    // Crear una consulta dinámica utilizando los parámetros proporcionados
    $clave = implode(' ', array_keys($campos)); //saca las columnas de las Keys
    $placeholders = ':' . implode(' ', array_keys($campos)); //prepara los placeholders para el funcionamiento de la función bindValue()
    $sql = "INSERT INTO $tabla ($clave) VALUES ($placeholders)"; //consulta
    $stmt = $pdo->prepare($sql); //preparación
    $stmt->execute(); //ejecución
    return $stmt->rowCount();
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
}
}
}

```

function crud_actualizar(\$tabla, \$campos, \$condicion)

Realiza las Modificación de las tablas de la BD, por parámetro entra el nombre de la tabla, campos y la condición.

Para hacer una modificación se realiza mediante las claves primarias, de esta forma no hay fallo de que tenga registros repetidos ya que cada registro contiene una clave diferente, por lo que la condición será la clave.

```

function crud_actualizar($tabla, $campos, $condicion) {//actualiza
try {
    // Conectar a la base de datos utilizando PDO
    $dsn = 'mysql:host=localhost;dbname=hospital';
    $username = 'root';
    $password = '';
    $pdo = new PDO($dsn, $username, $password);

    // Crear una consulta dinámica utilizando los parámetros proporcionados
    $set = '';
    foreach ($campos as $key => $value) {
        $set .= "$key = :$key, "; //clave es igual a clave
    }
    $set = rtrim($set, ', '); // quita la última coma

    $sql = "UPDATE $tabla SET $set WHERE $condicion"; //crea la consulta
    $stmt = $pdo->prepare($sql); //prepara la consulta

    // Vincular los valores a los marcadores de posición
    foreach ($campos as $key => $value) {
        $stmt->bindValue(': ' . $key, $value); // se reemplaza con la función bindValue()
    }

    // Ejecutar la consulta y retornar el número de filas afectadas
    $stmt->execute();
    return $stmt->rowCount();
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
}
}
}

```

function crud_borrar(\$tabla, \$id) {

Esta función realiza la acción eliminar registros. Como comentaba anteriormente cada registro tiene una clave única por lo que esta función entra por parámetro la tabla que donde se va a eliminar el registro y el id del registro.

```

function crud_borrar($tabla, $id) {//elimina
try {
    // Conectar a la base de datos utilizando PDO
    $dsn = 'mysql:host=localhost;dbname=hospital';
    $username = 'root';
    $password = '';
    $pdo = new PDO($dsn, $username, $password);

    // Crear una consulta dinámica utilizando los parámetros proporcionados
    $sql = "DELETE FROM $tabla WHERE id" . $tabla . " = " . $id;// el id de la tabla es igual al id que entra por parámetro
    $stmt = $pdo->prepare($sql);

    // Ejecutar la consulta y retornar el número de filas afectadas
    $stmt->execute();
    return $stmt->rowCount();
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
}
}
}

```

function crud_borrar_relacion(\$tabla,\$id1, \$valor1, \$id2, \$valor2)

Esta función ha sido creada para poder eliminar registros unidos por una tabla, por ejemplo en la tabla Centro y la tabla departamentos su relación es de muchos a muchos por lo que se crea una tabla intermedia con el id de cada una. Por ese motivo por parámetro entra el nombre de la tabla intermedia, el id de la primera tabla, el valor de la primera tabla, el id de la segunda tabla y el valor de la segunda tabla y se realiza la consulta para eliminar el registro con los ids y valores de los parámetros.

```

function crud_borrar_relacion($tabla,$id1, $valor1, $id2, $valor2) {//elimina relación de muchos a muchos
try {
    // Conectar a la base de datos utilizando PDO
    $dsn = 'mysql:host=localhost;dbname=hospital';
    $username = 'root';
    $password = '';
    $pdo = new PDO($dsn, $username, $password);

    /* Crear una consulta dinámica
     * Tabla (M-M) si el id1 es igual al valor1 que entra y el id2 es igual al valor2 que entra
     * DELETE FROM hospital.centro_departamento where idCentro=1 AND idDepartamento=1
     */
    $sql = "DELETE FROM " . $tabla . " WHERE id" . $id1 . " = " . $valor1 . " AND id" . $id2 . " = " . $valor2;
    $stmt = $pdo->prepare($sql);

    // Ejecutar la consulta y retornar el número de filas afectadas
    $stmt->execute();
    return $stmt->rowCount();
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
}
}
}

```

PÁGINAS

Provincia

En primer lugar, se requiere el CRUD

```
<body><?php
require_once('../../../modelo/crud.php');
```

Se preparan las acciones con isset () para poder trabajar con la base de datos:

- Insertar;

```
if (isset($_POST['registrar'])) {//acción añadir
    $nombre = $_POST['nombre'];
    try {
        crud_insertar('Provincia', array('nombre' => $nombre));//pasa la tabla y los campos como array
    } catch (PDOException $e) {
        echo 'Error al insertar la provincia: ' . $e->getMessage();
    }
}
```

En este caso la tabla contiene dos campos el “id” y campo “nombre”, el id no hace falta ya que se incrementa solo en la BD, por lo que solo pongo el campo “nombre”.

Como ya había dejado preparado en el CRUD la función crud_insertar(), que le entra por parámetro el nombre de la tabla (Provincias) y los campos, en este caso solo uno (Nombre).

En caso de dar error devuelve un mensaje.

- Modificar;

```
if (isset($_POST['modificar'])) {//acción modificar
    $id = $_POST['idProvincia'];//id que le pasa cuando hace onclick y llama a la función seleccionar
    $nombre = $_POST['nombre'];//campo que muestra

    try {//actualiza los datos por el id de provincia que se ha guardado en el hidden
        crud_actualizar('Provincia', array('nombre' => $nombre), "idProvincia = '$id');//tabla, campos,condicion(id)
    } catch (PDOException $e) {
        echo 'Error al insertar la provincia: ' . $e->getMessage();
    }
}
```

Para modificar el registro, el id me viene cuando el usuario hace click en un registro, es decir cuando se pinta la tabla más abajo, con el evento onclick realiza la llamada a la función seleccionar de JavaScript.

```
'<tr onclick="seleccionar(' . $provincia['idProvincia'] . ')";>
```

Según la fila en la que hace click el usuario llama a la función JavaScript seleccionar, esta función va al controlador, realiza un Ajax con la acción seleccionar y el id que viene por parámetro.

```

crud.php M   controlador_provincia.php X
controlador > controlador_provincia.php
1  <?php
2  require_once('../modelo/crud.php');
3
4  if (isset($_POST['action'])){
5
6      if ($_POST['action'] == 'seleccionar'){
7          try {
8              $provincia = crud_select('Provincia', 'idProvincia', $_POST['id']);
9              header("Content-Type: application/json");
10             echo json_encode($provincia);
11         } catch (PDOException $e) {
12             echo 'Error al insertar la provincia: ' . $e->getMessage();
13         }
14     }
15 }
16
17 ?>

```

El controlador devuelve un JSON, de esa manera el controlador no devuelve un HTML, el JSON se parsea en la respuesta cuando acaba el Ajax, si todo está correcto pone el nombre de los campos de ese id en el label, en este caso solo es un campo “Ciudad”.

```

function seleccionar(id) {
    var http = new XMLHttpRequest();
    var url = '/Hospital2/controlador/controlador_provincia.php';
    var params = 'action=seleccionar&id=' + id;
    http.open('POST', url, true);

    //
    http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
    // maneja la respuesta del servidor
    http.onreadystatechange = function() {
        // respuesta ajax correcta
        if (http.readyState == 4 && http.status == 200) {
            var provincia = JSON.parse(http.response);
            document.getElementById('nombre').value = provincia[0].Nombre; // dato mapeados
            document.getElementById('registrar').style.display = 'none'; // cambia la visibilidad de los botones
            document.getElementById('modificar').style.display = 'block';
            // deja el id oculto para luego usarlo en modificar
            document.getElementById('idProvincia').value = id;
            document.getElementById('Anadir').style.display = "block";
        }
    }
    http.send(params);
}

```

El botón registrar es el que se muestra por defecto, el botón modificar y cancelar están ocultos.

Provincia	
Cadiz	<input type="button" value="Borrar"/>
C髍doba	<input type="button" value="Borrar"/>

Anterior 1 2 3 Siguiente

Nombre:

Al seleccionar haciendo click en una fila aparece el botón modificar, donde en el mismo formulario se pueden modificar los datos.

Provincia

Cadiz	<input type="button" value="Borrar"/>
Córdoba	<input type="button" value="Borrar"/>

Nombre: Córdoba

Si selecciona el botón cancelar, mediante el evento onclick llama a la función limpiarFormulario(), esta función lo que hace es limpiar los campos del formulario con value="" y oculta el botón modificar y registrar, justo al contrario que hace la función seleccionar().

```
function limpiarFormulario() {
    document.getElementById('nombre').value = "";
    document.getElementById('registrar').style.display = "block";
    document.getElementById('modificar').style.display = "none";
    document.getElementById('Anadir').style.display = "none";
}
```

- Eliminar;

```
if (isset($_POST['borrar'])) {//acción eliminar
    $id = $_POST['idProvincia'];

    try {//borra la provincia por id
        crud_borrar('Provincia', $id);
    } catch (PDOException $e) {
        echo "No se ha podido eliminar ya que la provincia está asignada en otro elemento.";
    }
}
```

En la parte de cliente, si el usuario pulsa el botón borrar

Provincia

Almeria	<input type="button" value="Borrar"/>
Malaga	<input type="button" value="Borrar"/>

Nombre: Almeria

Mediante el formulario, anteriormente mencionado que guarda el id, si hace click en el botón borrar, acciona “borrar” pasando el id que tiene oculto, además mediante la función confirm() confirma si quiere eliminar el registro.

Después vendría la preparación para realizar la tabla de provincias y la paginación.

```
$provincias=crud_get_all('provincia');//trae la tabla provincia
$total = count($provincias);/// empieza la paginación contando todos los usuarios
$página = isset($_GET['page']) ? $_GET['page'] : 1;//si me entra algo por página muestra esa página si no se va a la página 1
$porPagina = 2;//cantidad a mostrar por página
$paginasTotales = ceil($total / $porPagina);//ceil() redondea fracciones hacia arriba, realiza el cálculo de las páginas totales
$inicio = ($pagina - 1) * $porPagina;//
```

y pintar la tabla con PHP, añadiendo las columnas que va a tener la tabla, con una columna más para el botón borrar.

```
echo "<table class='table table-hover'>";
| echo "<tr>
| | <th scope='col'>Provincias</th>
| | <th scope='col'></th>
| </tr>";
```

A continuación, se sigue preparando la paginación para recorrer la tabla que entra por parámetro en este caso “provincias”, el inicio y cuantos tiene que mostrar por página.

```
/* $provincias es el array de provincias que viene desde el controlador
$usar_pagina = array_slice($provincias, $inicio, $porPagina); //recorre desde el inicio hasta cuantos tiene que mostrar $porPagina*/
```

Con el bucle recorre cada fila de la tabla, e imprime cada campo, como anteriormente he comentado el id no se tiene que mostrar por lo que en el formulario lo oculta con un input de tipo hidden y hace la función seleccionar cuando se hace click desde la parte del cliente y de este modo ya tengo la tabla pintada por pantalla con todas las funciones.

```
foreach ($usar_pagina as $provincia) {
    echo '<tr onclick="seleccionar(' . $provincia['idProvincia'] . ')">
        <td scope="row">' . $provincia['Nombre'] . '</td>
        <td>' . '<form action="" method="POST"><input type="hidden" name="idProvincia" value=' . $provincia['idProvincia'] . '">
            <button onclick="if(!confirm(\'' . Estás seguro de borrar el registro?')) event.preventDefault();" class="btn btn-primary" type="submit" name="borrar" value="Borrar">Borrar</button>
        </form>' . '</td>
    </tr>';
}
echo "</table>";
```

Ya quedaría el html, que contiene el botón cancelar que como ya he comentado realiza la función limpiarFormulario() y por defecto está oculto.

La paginación, que se realiza llamando a la función “paginación ()” creada en el archivo paginación.php y por último el formulario con los campos y los botones registrar que esta visible por defecto y el modificar que se encuentra oculto.

```
<button class="btn btn-primary" onclick="limpiarFormulario()" type="submit" id="Anadir" name="Anadir" value="Anadir"
| style="display: none;">Cancelar</button>
<!--formulario de usuarios-->
<nav class="paginacion" aria-label="Paginacion">
| <?php echo paginacion($paginasTotales, $pagina, '?'); //llama a la función de la paginación?>
</nav>
<div id="panel-modificar" class="d-flex">
    <form method="post" class="">
        <input type="hidden" name="idProvincia" id="idProvincia" value="">
        <div class="input-group mb-1 d-inline-flex p-1 bd-highlight">
            <!--clase de bootstrap-->
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">Nombre: </span>
            </div>
            <input required type="text" class="form-control" id="nombre" name="nombre" aria-describedby="basic-addon3">
            <button class="btn btn-primary" type="submit" id="registrar" name="registrar" value="Enviar">Registrar</button>
            <button class="btn btn-primary" type="submit" id="modificar" name="modificar" value="Modificar" style="display: none;">Modificar</button>
        </div>
    </form>
</div>
body>
```

Para la paginación he creado el archivo paginación.php, en este archivo realizo la función paginación.

A esta función le entran tres parámetros, \$paginasTotales (el número total de páginas disponibles), \$pagina (la página actual que está mostrando) y \$url (la URL base de la página).

```

vista > paginacion.php U
crud.php M paginacion.php X
  ↗
1  <?php
2  function paginacion($paginasTotales, $pagina, $url){
3      $inicio = max(1, $pagina - 2);
4      $fin = min($paginasTotales, $pagina + 2);
5      $html = "<ul class='pagination'>";
6      //verifica si la página actual no es la primera página y, si es así, agrega un botón "Anterior" que enlaza con la página anterior.
7      if ($pagina > 1) {
8          $html .= "<li class='page-item'><a class='page-link' href='".$url."&page=".($pagina-1)."'">Anterior</a></li>";
9      }
10     //si es mayor, agrega un botón para la página de inicio.
11     if ($inicio > 1) {
12         $html .= "<li class='page-item'><a class='page-link' href='".$url."&page=1'">1</a></li>";
13         //si es mayor que dos agrega un botón para las páginas anteriores que están disponibles.
14         if ($inicio > 2) {
15             $html .= "<li class='page-item disabled'><a class='page-link'>...</a></li>";
16         }
17     }
18     //recorre todas las páginas para agregar un botón y agrega la clase "active" para indicar en la página que está
19     for ($i=$inicio; $i<$fin; $i++) {
20         $activo = ($pagina == $i) ? " active" : "";
21         $html .= "<li class='page-item $activo'><a class='page-link' href='".$url."&page=$i'">$i</a></li>";
22     }
23     //es a la contra del inicio, si es menor que páginas totales agrega un botón para la última página.
24     if ($fin < $paginasTotales) {
25         //si es menor que páginas totales-1 un botón para las páginas siguientes que están disponibles.
26         if ($fin < $paginasTotales - 1) {
27             $html .= "<li class='page-item disabled'><a class='page-link'>...</a></li>";
28         }
29         $html .= "<li class='page-item'><a class='page-link' href='".$url."&page=$paginasTotales'">$paginasTotales</a></li>";
30     }
31     //verifica si la página actual no es la última página y, si es así, agrega un botón "siguiente" que enlaza con la página siguiente.
32     if ($pagina < $paginasTotales) {
33         $html .= "<li class='page-item'><a class='page-link' href='".$url."&page=".($pagina+1)."'">siguiente</a></li>";
34     }
35     $html .= "</ul>";
36     //devuelve el HTML
37     return $html;

```

Validación

Para la validación utilizo la palabra clave “required” comprobando que los campos no entran vacíos y sea obligatorio llenar el campo, esto se realiza en la parte de cliente, debido al poco tiempo que hay para realizar la aplicación no se realizará la validación en la parte del servidor, quedará pendiente para futuras mejoras.

```

<div class="input-group-prepend">
    <span class="input-group-text" id="basic-addon3">Nombre: </span>
</div>
<input required type="text" class="form-control" id="nombre" name="nombre"
    aria-describedby="basic-addon3">

```

Como se puede comprobar si se intenta entrar sin llenar los campos saltará un mensaje para informar que debe llenar el campo.



Ciudad

Para la página de la tabla ciudad he utilizado el mismo modo que la tabla provincias, en lo que se diferencia es que la tabla ciudad tiene una FK que la une con la tabla Provincia, por lo que se necesita la clave de provincia.

Esto lo realizo cuando voy a pintar la tabla por pantalla realizando una consulta a la tabla “Provincia” para que me dé el nombre de cada una de ellas por su id y de esta forma mostrar el nombre y no la clave, ya que no se sabría a qué provincia pertenece.

```
$provincia = crud_select('Provincia', 'idProvincia', $ciudad['idProvincia']) [0];
echo '<tr onclick="seleccionar" . $ciudad['idCiudad'] . ')';
    <td scope="row">' . $ciudad['Nombre'] . '</td>
    <td>' . $provincia['Nombre'] . '</td>
    <td>' . '<form action="" method="POST"><input type="hidden" name="idCiudad" value=' . $ciudad['idCiudad'] . '">
        <button onclick="confirm(\'' . $provincia['Nombre'] . '\')"; class="btn btn-primary" type="submit" name="borrar" value="Borrar">Borrar</button>
    </form>' . '</td>
</tr>';
}
echo "</table>";
```

Para mostrar las provincias realizo una consulta y mediante un <Select> con id=“idProvincia”, siendo este selector importante ya que obtienen todos los id y un bucle foreach me va a mostrar todas las provincias existentes en la tabla.

```
$provincias=crud_get_all('provincia');//trae la tabla provincia
echo '<select id="idProvincia" name="idProvincia" class="form-select">'; //imprime el select con el idProvincia

foreach ($provincias as $provincia) { //Recorre las provincias
    echo '<option selected value=' . $provincia['idProvincia'] . '>' . $provincia['Nombre'] . '</option>; //Imprime una opcion por cada provincia
}

echo '</select>'; //Fin del select
?>
```

Ciudades	Provincia	Borrar
Vera	Almeria	<button>Borrar</button>
Adra	Almeria	<button>Borrar</button>

1	2	3	Siguiente
Ciudad:	Provincia:	Jaén	Registrar
		Almeria	
		Malaga	
		Granada	
		Sevilla	
		Cadiz	
		Córdoba	
		Jaén	

Y de este modo, la tabla Ciudad estaría terminada.

Centro y Departamento

El código para la creación de la página en la que se muestra la tabla Centro se realiza igual que la tabla ciudad, cada centro depende de una ciudad por lo que requiere los id de la tabla Ciudad y para cada centro me traigo su ciudad con el idCiudad.

La tabla Departamento se realiza lo mismo que la tabla provincias, no tiene ninguna FK.

Pero la relación de la tabla Centro y la tabla Departamento es de M:M por lo que genera una tabla intermedia con las claves primarias de cada tabla como claves secundarias en la tabla que genera que es la tabla Centro_Departamento.

La página que muestra la tabla Centro_Departamento la he puesto junto con la vista de la tabla Centro, se puede hacer por separado creando otra página, pero de este modo muestro que departamentos tiene cada centro.

En primer lugar, hago la llamada a la función crud_get_all() para realizar una consulta de toda la tabla y pinto los campos que contiene mediante php, añadiendo un campos de más para el botón borrar.

Como lo que quiero son los centros y departamentos se recorre toda la tabla y en la variable \$departamento con la función crud_select() me guarda los idDepartamento de cada registro de la tabla “Departamentos”, en la variable \$centro realizo el mismo proceso para guardar los idCentro de cada registro.

También obtengo los nombres de cada centro y de cada departamento y los id los oculto con un imput de tipo hidden, para cuando se accione el botón borrar mediante el evento onclick, realice la confirmación de que quiere realizar la eliminación del registro mediante el id.

```
$centroDepartamentos=crud_get_all('Centro_Departamento');//trae la tabla centro_departamento
echo '<table class="table table-hover">';
echo '<tr>
    <th scope="col">Centro</th>
    <th scope="col">Departamento</th>
    <th scope="col"></th>
</tr>';
foreach ($centroDepartamentos as $centroDepartamento) [
    $departamento = crud_select('departamento', 'idDepartamento', $centroDepartamento['idDepartamento'])[0];
    $centro = crud_select('centro', 'idCentro', $centroDepartamento['idCentro'])[0];
    echo '<tr>
        <td scope="row">' . $centro['Nombre'] . '</td>
        <td scope="row">' . $departamento['Nombre'] . '</td>
        <td> ' . '<form action="" method="POST"><input type="hidden" name="idDepartamento" value=' . $centroDepartamento['idDepartamento'] . '">
            <input type="hidden" name="idCentroDepartamento" value=' . $centroDepartamento['idCentro'] . '">
            <button onclick="confirm(\\"¿Estás seguro de borrar el registro?\");" class="btn btn-primary" type="submit" name="borrarDepartamento" value="Borrar">Borrar
        </button>
        </form>' . '</td>
    </tr>';
]
echo "</table>";
```

Cuando se acciona el botón borrar mediante la función isset() con el id de cada centro y cada departamento llamo a la función crud_borrar_relacion() que he creado en el CRUD.

```
//borra un elementos de la tabla M:M Centro_Departamento
if (isset($_POST['borrarDepartamento'])) {
    $idDepartamento = $_POST['idDepartamento'];
    $idCentro = $_POST['idCentroDepartamento'];
    try {
        crud_borrar_relacion('Centro_Departamento','Centro', $idCentro, 'Departamento', $idDepartamento);
    } catch (PDOException $e) {
        echo "No se ha podido eliminar el departamento ya que está asignado en otro elemento.";
    }
}
```

Al igual que he realizado en la representación de la tabla “Ciudad” las provincias realiza la representación de los centros y los departamentos.

El formulario solo tiene <Select> para obtener todos los departamentos y todos los centros, en este caso no realizo la acción modificar por el corto tiempo para la realización de la aplicación, si se quiere hacer alguna modificación se tiene que borrar y volver a crear el registro, por lo que solo contiene el botón registrar departamento en centro.

```
xampp >htdocs > Hospital > vista > Administrador > centro.php
  ... para el Departamento
  ...
  <form method="post" class="form">
    <div class="input-group mb-1 d-inline-flex p-1 bd-highlight">
      <div class="input-group-prepend">
        <span class="input-group-text" id="basic-addon3">Centro:</span>
      </div>
      <?php
        //Igual que en la tabla ciudad
        $centros=crud_get_all('centro');//trae la tabla ciudad
        echo '<select id="idCentroDepartamento_Centro" name="idCentroDepartamento_Centro" class="form-select">'; //imprime el select con el idCiudad
        foreach ($centros as $centro) { //Recorre las ciudades
          echo '<option selected value=' . $centro['idCentro'] . '>' . $centro['Nombre'] . '</option>; //Imprime una opcion por cada ciudad
        }
      ?>
      </select>

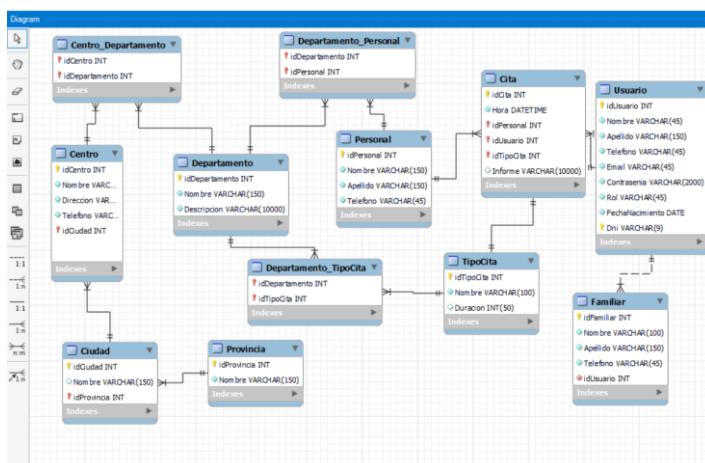
      <div class="input-group-prepend">
        <span class="input-group-text" id="basic-addon3">Departamento:</span>
      </div>
      <?php
        $departamentos=crud_get_all('departamento');//trae la tabla ciudad
        <select id="idCentroDepartamento_Departamento" name="idCentroDepartamento_Departamento" class="form-select">
          <!--imprime el select con el idCiudad-->
        <?php
        foreach ($departamentos as $departamento) { //Recorre las ciudades
          echo '<option selected value=' . $departamento['idDepartamento'] . '>' . $departamento['Nombre'] . '</option>; //Imprime una opcion por cada ciudad
        }
      ?>
      </select>
      <!--fin del select-->
      <button class="btn btn-primary" type="submit" id="RegistrarCentroDepartamento" name="registrarCentroDepartamento" value="EnviarCentroDepartamento">Registrar departamento en centro</button>
    </div>
  </form>
```

El botón de registro de tipo “submit” realiza mediante el id= “registrarCentroDepartamento” la acción de registrar mediante isset(), cogiendo cada campo mediante el name= “idCentroDepartamento_Departamento” para los Departamentos y el name= “idCentroDepartamento_Centros”, realizo la llamada a la función crud_insertar() creada en el CRUD.

```
// acciones en la tabla M:M Centro_Departamento
if (isset($_POST['registrarCentroDepartamento'])) {
  $idCentro = $_POST['idCentroDepartamento_Centro'];
  $idDepartamento = $_POST['idCentroDepartamento_Departamento'];
  try {
    crud_insertar('Centro_Departamento', array('idCentro' => $idCentro, 'idDepartamento' => $idDepartamento));
  } catch (PDOException $e) {
    echo "No se ha insertado.";
  }
}
```

Este ha sido el modo que he encontrado para realizar la unión de las tablas con relación M:M, por este motivo he creado en patrón del CRUD la función crud_borrar_relación ()).

Para la creación de las demás páginas que representan las demás vistas que tiene la aplicación se ha seguido el mismo proceso.



15. Documentación incremento 2

Funcionalidades

Página de inicio:



Acerca de

Está página la función que tiene es dar información de como se crearon y la historia de los Centros CenSalud.

Hospitales

En esta página se muestran los centros con la dirección de cada uno.

Especialistas

Se muestran los Especialistas que hay en todos los centros.

Contacto

Aquí el usuario puede buscar por ciudad el centro y le aparecerá las direcciones junto con su teléfono.





Para el Back-End de pagina en el modelo he creado un archivo `modelo_contacto.php`, que contiene la función para la realización de la llamada a la BD, esta consulta solo sirve para mostrar las direcciones de los centros y sus teléfonos.

```

contacto.php M | modelo_contacto.php U X
modelo > modelo_contacto.php
1 <?php
2 //Busca la direccion y telefono de los centros por ciudad
3 function obtener_centros_por_ciudad($condicion){
4     try {
5         // Conectar a la base de datos utilizando PDO
6         $dsn = 'mysql:host=localhost;dbname=hospital';
7         $username = 'root';
8         $password = '';
9         $pdo = new PDO($dsn, $username, $password);
10        //realiza la consulta
11        $sql = "SELECT c.Direccion, c.Telefono
12        FROM Centro c
13        JOIN Ciudad ci ON c.idCiudad = ci.idCiudad
14        WHERE ci.Nombre = '$condicion'";
15        $stmt = $pdo->prepare($sql); //prepara la consulta
16        $stmt->execute(); //ejecuta
17
18        // Retornar los resultados
19        return $stmt->fetchAll(PDO::FETCH_ASSOC); //devuelve todos los registros
20    } catch (PDOException $e) {
21        echo "Error: " . $e->getMessage();
22    }
23 }
24 ?>

```

Y en `contacto.php`, mediante un formulario el usuario escribe la Ciudad y al pulsar buscar le devuelve la consulta.

```

<form action="" method="post">
    <input required type="text" class="form-control" id="nom_ciudad" name="nom_ciudad"
           aria-describedby="basic-addon3" placeholder="Ciudad" />
    <button class="btn btn-primary" style="margin-top:10px" type="submit" id="buscar"
           name="buscar" value="Enviar">>Buscar</button>
    <!--Tabla para mostrar los datos -->
<?php
    require_once('../modelo/modelo_contacto.php');
    if (isset($_POST['buscar'])) {
        $ciudad = $_POST['nom_ciudad'];
        try{
            $busqueda=obtener_centros_por_ciudad($ciudad);
        }catch(PDOException $e) {
            echo 'Error en la búsqueda de la ciudad: ' . $e->getMessage();
        }
        echo "<table class='table table-hover'>";
        foreach($busqueda as $resultado){
            echo '<tr>
                    <td scope="row">' . $resultado['Direccion'] . '</td>
                    <td scope="row">' . $resultado['Telefono'] . '</td>
                </tr>';
        }
        echo "</table>";
    }
?>
</form>

```

Para la utilización de las funcionalidades mencionadas anteriormente, el usuario no necesita estar registrado, para registrarse mediante el botón registrarse.

Registro



Cuando el usuario introduce sus datos en el formulario y pulsa el botón enviar, se realiza una llamada a la función crud_inserta que se ha realizado en el CRUD.

```

registro.php M X
vista > registro.php
1  <?php
2  require_once('../modelo/crud.php');
3  $msg = '';
4  if (isset($_POST['registrar'])) {
5      //Trae los datos del formulario
6      $nombre = $_POST['nombre'];
7      $apellido = $_POST['apellidos'];
8      $dni = $_POST['dni'];
9      $telefono = $_POST['telefono'];
10     $fnacimiento = date('Y-m-d', strtotime(str_replace('-', '/', $_POST['fnacimiento'])));
11     $email = $_POST['email'];
12     $contrasenia = $_POST['password'];
13
14     $valido = true;
15     if($valido){
16         try {//con la función insertar los inserto en la BD, por defecto el rol será usuario y lo redirige a la página del login
17             crud_insertar('Usuario', array(['dni' => $dni, 'nombre' => $nombre, 'apellido' => $apellido,
18                                         'telefono' => $telefono, 'FechaNacimiento' => $fnacimiento, 'contrasenia' => password_hash($contrasenia, PASSWORD_DEFAULT),
19                                         'email' => $email, 'rol' => 'Usuario']);
20             header("location: login.php");
21         } catch (PDOException $e) {
22             echo 'Error al insertar usuario: ' . $e->getMessage();
23         }
24     }
25 }

```

Por defecto el Rol será el de “Usuario” y le redirige a la página del Login para poder ir al Área Cliente.

La validación se realiza con “required” en el lado del cliente, impidiendo que los campos entren vacíos, en caso de que ocurra algún error mandara el mensaje “Error al insertar el usuario”.

```
<div class="container_registro">
<div class="login_box">
    <h2>Registro de usuario</h2>
    <!--Mensaje de error-->
    <?php if (!empty($msg)): ?>
    <p><?php echo $msg; ?></p>
    <?php endif; ?>
    <!--formulario----->
    <form method="post" class="form_login">

        <input type="text" id="nombre" name="nombre" required placeholder="Nombre"><br>

        <input type="text" id="apellidos" name="apellidos" required placeholder="Apellidos"><br>

        <input type="text" id="dni" name="dni" required placeholder="Documento (DNI)"><br>

        <input type="text" type="datetime-local" id="fnacimiento" name="fnacimiento" required
               placeholder="Fecha de Nacimiento"><br>

        <input type="tel" id="telefono" name="telefono" required placeholder="Teléfono"><br>

        <input type="email" id="email" name="email" required placeholder="Email"><br>

        <input type="password" id="password" name="password" required placeholder="Contraseña"><br>

        <input type="submit" id="registrar" name="registrar" value="Enviar">
    </form>
</div>
</div>
```

Login



Para el login he usado la validación por parte del servidor, en la que al pulsar el botón login y si los campos no entran vacíos, guardo el usuario en una variable y el password en otra este valor me viene por el formulario del login mediante POST.

```

<div class="container_login">
    <div class="login_box">
        <h2>Login</h2>
        <!--Mensaje de error-->
        <?php if (!empty($msg)): ?>
        <p><?php echo $msg; ?></p>
        <?php endif; ?>
        <form action="" method="post">
            <div class="form_login">
                <label for="username">Email</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="form_login">
                <label for="password">Password</label>
                <input type="password" id="password" name="password" required>
            </div>
            <div class="form_login button">
                <input type="submit" id="login" name="login" value="Login">
            </div>
            <div class="contra_olvidar">
                <!-- <p><a href="#">¿Has olvidado tú contraseña?</a></p>-->
                <p><a href="index.php">Volver</a></p>
            </div>
        </form>
    </div>
</div>

```

Declaro la variable “encontrado=false” para saber si lo ha encontrado al usuario en la base de datos.

Realizo una llamada a la función crud_select del CRUD para buscar al usuario y con la función de php password_verify () verifico que la contraseña introducida es correcta.

Si lo ha encontrado, inicia la sesión y se guardan los datos de usuario en las variables de sesión, según el rol de cada uno lo redirige al portal correspondiente.

```

1 <?php
2 require_once('../modelo/crud.php');
3 // si ha pulsado el botón login y los campos username y password no vienen vacios,
4 //validación por parte del servidor.
5 if (isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) {
6     $username = $_POST['username'];
7     $password = $_POST['password'];
8
9     $encontrado = false;//no encontrado
10    $usuario = crud_select('Usuario', 'Email', $username);//se busca por email
11    if ($usuario && password_verify($password, $usuario[0]['Contrasenia'])) {//con password_verify verifica la contraseña
12        $encontrado = true;//si lo ha encontrado
13        session_start();//si encuentra al usuario inicia sesión y se guardan los datos del usuario en las variables de sesión
14        $_SESSION['usuario'] = $usuario[0]['Nombre'];
15        $_SESSION['idusuario'] = $usuario[0]['idusuario'];
16        $_SESSION['contraseniaHash'] = $usuario[0]['Contrasenia'];
17        $_SESSION['apellido'] = $usuario[0]['Apellido'];
18        $_SESSION['telefono'] = $usuario[0]['Telefono'];
19        $_SESSION['fnacimiento'] = date("d-m-Y", strtotime(str_replace('-', '/', $usuario[0]['FechaNacimiento'])));
20        $_SESSION['email'] = $usuario[0]['Email'];
21        $_SESSION['dni'] = $usuario[0]['Dni'];
22        //según el Rol manda al usuario a su portal
23        //Si es administrador hay que mandarlo al portal del admin si no al portal de usuario
24        if($usuario[0]['Rol'] == 'Especialista'){
25            header('Location: especialista/portal_especialista.php');
26        }
27        if ($usuario[0]['Rol'] == 'Administrador') {
28            header('Location: administrador/portal_administrador.php');
29        }
30        if($usuario[0]['Rol'] == 'Usuario'){
31            header('Location: portal_usuario.php');
32        }
33        if($usuario[0]['Rol'] == 'Usuario_autorizado'){
34            header('Location: personal_autorizado/portal_usuario_autorizado.php');
35        }
36        exit; // Es importante salir del script después de la redirección
37    }
}

```

En el caso de no haberlo encontrado lanza un mensaje.

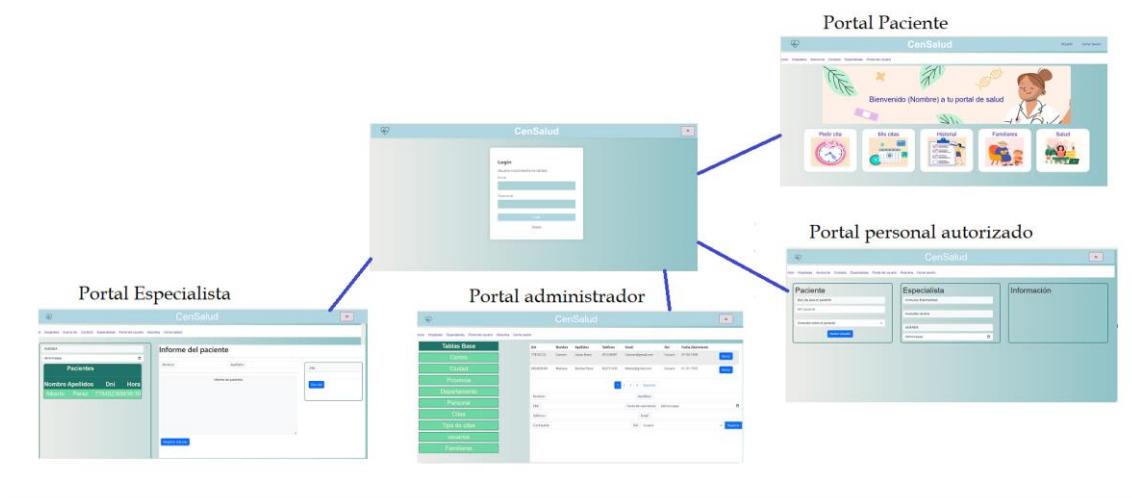
```

if (!$encontrado) {//si no lo ha encontrado
| $msg = 'Usuario o contraseña no válidos';
}
?>

```



Como ya he mencionado anteriormente según el rol del usuario redirige al portal correspondiente.



Portal administrador

El back-end del administrador se ha explicado en el punto 14. Descripción del trabajo, su funcionalidad es poder crear registros en todas las tablas, darle el Rol al usuario personal, en definitiva, el mantenimiento.

Este portal muestra en la parte izquierda una tabla que contiene el nombre de todas las vistas, cuando selecciona una vista, mediante el evento onclick que pone en espera cada página que contiene cada vista cuando hace click le redirige a la página, esta página se encuentra dentro de un iframe y se muestra en la parte derecha del portal, por defecto cuando entra muestra la tabla usuarios.

```

  portal_usuario_autorizado.php M  Portal_administrador.php X  autorizado_especialistas_citas.php U  autorizado_especialistas.php U
vista > Administrador > Portal_administrador.php
      <div class="col-md-4 float-left">
        <!--div para mostrar las tablas de BD-->
        <table class="tab_administrador texto">
          <tr>
            |   <th>Tablas Base</th>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'centro.php'">Centro</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'ciudad.php'">Ciudad</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'provincia.php'">Provincia</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'departamento.php'">Departamento</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'personal.php'">Personal</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'cita.php'">Citas</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'tipocita.php'">Tipo de citas</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'usuario.php'">Usuarios</a>
          </tr>
          <tr>
            |   <a href="#" onclick="document.getElementById('iframe').src = 'familiar.php'">Familiares</a>
          </tr>
        </table>
      <div class="col-md-8 float-right ">
        <!--es el que primero se abre por defecto-->
        <div class="tam_derecho">
          <div>
            <iframe title='administrador' id='iframe' src="usuario.php">
          </div>
        </div>
        <div class="clearfix"></div>
      </div>
    </div>
  
```

Dni	Nombre	Apellidos	Teléfono	Email	Rol	Fecha Nacimiento
77455212L	Carmen	López Bravo	451236987	Carmen@gmail.com	Usuario	07-06-1999
88546963M	Matias	Benitez Pérez	452111478	Matias@gmail.com	Usuario	01-01-1970

Tiene efecto hover cuando pasa por encima del registro.

El botón cancelar y modificar por defecto está oculto, cuando se selecciona un registro los datos del registro se muestran en los campos, muestra el botón de modificar y directamente se pueden modificar los datos sobre esos, la contraseña no se muestra y el DNI no se puede modificar, también aparece el botón cancelar.

En el punto 14. Descripción del trabajo se explica más detalladamente, ya que el patrón del CRUD está enfocado para el mantenimiento de las vistas.

The screenshot shows a web-based application interface titled "CenSalud". On the left, there is a sidebar with a green header titled "Tablas Base" containing links for Centro, Ciudad, Provincia, Departamento, Personal, Citas, Tipo de citas, usuarios, and Familiares. The main content area displays a table with columns: Dni, Nombre, Apellidos, Teléfono, Email, Rol, and Fecha Nacimiento. Two rows of data are shown, each with a "Borrar" button. Below the table, a modal window is open for the first row, showing fields for Nombre (Carmen), Apellidos (López Bravo), Dni (77455212L), Teléfono (451236987), Email (Carmen@gmail.com), Rol (Usuario), and Fecha Nacimiento (07-06-1999). The modal also contains a "Cancelar" button and a "Modificar" button.

Como se puede comprobar debajo de los registros se muestra la paginación.

This screenshot is similar to the one above, showing the same application interface. The table and modal window are identical. However, at the bottom of the table, there is a navigation bar with buttons for "Anterior", "1", "2", "3", and "Siguiente", indicating a paginated list.

Los centros se componen de departamentos, por lo que para crear un departamento debe de tener centros, también se ubican en ciudades lo que deben tener una ciudad asociada y la ciudad una provincia.

En la parte de arriba se muestran los registros de las vistas y en la parte de abajo la relación con la correspondiente vista.

This screenshot shows a different section of the application. On the left, the "Tablas Base" sidebar is present. The main content area has two tables. The top table lists "Nombre", "Dirección", "Teléfono", "Ciudad", and "Borrar" buttons. The second row shows "Mar adentro" with address "C/nueva,17", phone "952223344", city "Torremolinos", and a "Borrar" button. The bottom table lists "Centro" and "Departamento". It shows three entries: "Mar adentro" with "Podología", "Mar adentro" with "Dermatología", and "Salud" with "Oftalmología". Below these tables, a registration form is visible with fields for "Centro" (dropdown menu with "Mar adentro" selected), "Dirección", "Teléfono", "Ciudad" (dropdown menu with "Marbella" selected), and a "Registrar" button.

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia
- Departamento
- Personal
- Citas
- Tipo de citas
- usuarios
- Familiares

Ciudades	Provincia	Borrar
Torremolinos	Málaga	<input type="button" value="Borrar"/>
Martos	Jaén	<input type="button" value="Borrar"/>

1 2 Siguiente

Ciudad: Provincia: Córdoba Registrar

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia**
- Departamento
- Personal
- Citas
- Tipo de citas
- usuarios
- Familiares

Provincia	Borrar
Málaga	<input type="button" value="Borrar"/>
Jaén	<input type="button" value="Borrar"/>

1 2 3 4 Siguiente

Nombre: Registrar

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia
- Departamento**
- Personal
- Citas
- Tipo de citas
- usuarios
- Familiares

Nombre	Descripción	Borrar
Endocrinología	Nutrición	<input type="button" value="Borrar"/>
Podología	Pies	<input type="button" value="Borrar"/>

1 2 3 4 Siguiente

Nombre: Descripción: Registrar

Departamento	Personal	Borrar
Urología	Isabel Montes Alpes	<input type="button" value="Borrar"/>

Departamento: Urología Personal: Isabel Montes Alp: v Registrar personal en departamento

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia
- Departamento
- Personal**
- Citas
- Tipo de citas
- usuarios
- Familiares

Nombre	Apellido	Telefono	Borrar
Isabel	Montes Alpes	263222963	<input type="button" value="Borrar"/>

1

Nombre: Apellido: Telefono: Usuario: Registrar

CenSalud

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia
- Departamento
- Personal
- Citas**
- Tipo de citas
- usuarios
- Familiares

Hora	Usuario	Personal	Tipo Cita	Borrar
2023-05-11 20:14:00	Ana López Suarez	Isabel Montes Alpes	Rutinaria	<button>Borrar</button>
2023-05-11 22:19:00	Ana López Suarez	Isabel Montes Alpes	Rutinaria	<button>Borrar</button>

Siguiente

Horas: dd/mm/aaaa ->--> 88756952N López Suarez Montes Alpes, Isabel Rutinaria

CenSalud

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia
- Departamento
- Personal
- Citas
- Tipo de citas**
- usuarios
- Familiares

Nombre	Duración	Borrar
Primer consulta	15	<button>Borrar</button>
Revisión	10	<button>Borrar</button>

Siguiente

Nombre: Duración:

Departamento	Tipo de Cita	Borrar
Urología	Primer consulta	<button>Borrar</button>
Urología	Revisión	<button>Borrar</button>
Urología	Rutinaria	<button>Borrar</button>

Departamento: Urología Tipo de cita: Rutinaria

CenSalud

Inicio Hospitales Especialistas Portal del usuario Hola Ana Cerrar sesión

Tablas Base

- Centro
- Ciudad
- Provincia
- Departamento
- Personal
- Citas
- Tipo de citas
- usuarios
- Familiares**

Nombre	Apellido	Teléfono	Familiar	Borrar
Nombre:	Apellido:	Teléfono:	Usuario: 88756952N López:	<input type="button" value="Registrar"/>

Portal Usuario

The diagram illustrates the user interface flow from the CenSalud system to the user portal. It begins with a login screen, followed by a dashboard with various service icons. From the dashboard, users can access specific services such as 'Pedir cita' (Request Appointment) and 'Historial' (History).

La etiqueta salud, contiene información sobre consejos de como cuidar la salud, en esta página el usuario pulsará sobre una tarjeta de la izquierda y en el panel de la derecha le mostrará la información relacionada.

Para la realización de esta página mediante iframe en el lado derecho de la página con el evento onclick lo pongo en espera y cuando el usuario hace click le sale la información asignada.

```
<div class="row">
    <!--izquierdo, etiquetas a elegir -->
    <div class="col-md-3 float-left">
        <div class="card tam_tarjeta_salud" onclick="document.getElementById('iframe').src ='contenido_salud.html'>
            
            Salud y bienestar
        </div>
        <div class="card tam_tarjeta_salud" onclick="document.getElementById('iframe').src ='contenido_investigación.html'>
            
            Investigación
        </div>
        <div class="card tam_tarjeta_salud" onclick="document.getElementById('iframe').src ='contenido_prevención.html'>
            
            Prevención
        </div>
    </div>
    <!--derecho, contenidos de información -->
    <div class="col-md-9 float-right">
        <div class="tam_derecho">
            <div style="margin: 10px 20px 30px 10px;">
                <iframe title='consejos' id='iframe' src="contenido_salud.html">
            </div>
        </div>
    </div>
    <div class="clearfix"></div>
</div>
```

Salud y bienestar

Gozar de una buena salud es de suma importancia, ya que mejora nuestra calidad de vida y poder vivir una vida plena, sea cual sea la edad en la que nos encontramos.

Llevar una vida saludable no es tan fácil ya que por costumbres, trabajo, ir deprisa, etc. Nuestros hábitos son poco saludables y aun siendo consciente de ello nos lleva a tener problemas de salud en un futuro. Una motivación para cambiar nuestros hábitos saludables es ver los beneficios y ser conscientes de lo que nos pueden aportar en nuestro estado físico, mental y afectivo.

- Previene problemas de sobrepeso, nos ayuda a perder peso.
- Aumenta los niveles de energía.
- Mejora la acción respiratoria.
- Ayuda a tener un desarrollo y crecimiento más saludable.
- Aumenta el rendimiento en general: fuerza, velocidad, resistencia.
- Hay una gran mejoría de la coordinación.

Investigación

En los últimos años, los avances tecnológicos han tenido un gran impacto en la industria de la salud, lo que ha permitido avances significativos en el diagnóstico y tratamiento de enfermedades como el cáncer y las enfermedades raras.

En el campo del cáncer, la tecnología ha permitido el desarrollo de terapias más específicas y personalizadas. La secuenciación del genoma del cáncer ha permitido identificar mutaciones genéticas específicas que pueden ser tratadas con medicamentos dirigidos a esas mutaciones.

Además, la tecnología de imagen avanzada como la tomografía por emisión de positrones (PET) y la resonancia magnética (MRI) permiten la detección temprana del cáncer y su localización precisa.

En el caso de las enfermedades raras, la tecnología también ha tenido un impacto significativo.

La secuenciación del genoma humano ha permitido identificar mutaciones genéticas específicas que causan enfermedades raras, lo que ha llevado al

The screenshot shows a web page titled "Previsión" (Prevention) from the "CenSalud" website. On the left, there is a sidebar with three icons: "Salud y bienestar" (Health and well-being), "Investigación" (Investigation), and "Prevención" (Prevention). The main content area contains text about cancer prevention, mentioning sun protection, healthy diet, and avoiding tobacco and alcohol.

La etiqueta histórica

En esta página se muestran los informes clínicos que el especialista le ha creado en cada cita, mediante el inicio de sesión introduzco el id de usuario como condición que entra por parámetro en la función del CRUD crud_select () para consultar la vista “citas” mediante la condición, en este caso como el usuario que está logado por sesión es quien consulta sus citas, utilizo la variable de sesión.

```
<?php
require_once('../modelo/crud.php');
session_start();
//se obtienen las citas por id usuario guardado en el login
$citas = crud_select('cita', 'idUsuario', $_SESSION['idUsuario'] );
?>
```

Como la tabla citas se compone de los campos idUsuario, idPersonal y idTipoCita.

Result Grid						
	idCita	Hora	idPersonal	idUsuario	idTipoCita	Informe
▶	1	2023-05-11 20:14:00	1	7	3	La paciente ha recibido el alta tras el tratamiento.
	3	2023-05-11 20:17:00	1	1	3	Molestias estomacales y vomito se le recomienda...
	4	2023-05-24 22:18:00	1	7	2	El paciente se encuentra estable.
	5	2023-05-26 16:11:00	1	8	1	El paciente no requiere cirugía, se ha recuperad...
	6	2023-05-02 16:16:41	1	8	3	La paciente presenta fuerte dolor de cabeza
	7	2023-05-02 16:16:41	1	8	3	La paciente presenta fuerte dolor de cabeza
	8	2023-05-11 00:00:00	1	7	3	NUL
	9	2023-05-11 00:00:00	1	7	3	NUL
	10	2023-05-11 00:00:00	1	7	1	NUL
*	NUL	NUL	NUL	NUL	NUL	NUL

Realizo la llamada a la función crud_select para obtener los campos de la vista "TipoCitas" y "Personal" y de este modo obtengo los campos y los imprimo por pantalla.

```

<div id="accordion">
    <?php
        //para cada cita obtenida en la consulta anterior
    foreach ($citas as $cita) {
        //se obtiene el tipo de cita y el personal de la cita
        $tipo_cita = crud_select('Tipocita', 'idTipocita',$cita['idTipocita']);
        $personal = crud_select('Personal', 'idPersonal',$cita['idPersonal']);
        //para cada cita se pinta una tarjeta con la informacion de la cita, nombre, hora, informe. Se ha intentado usar un carrusel de bootstrap pero no funciona
        echo '<div class="card">
            <div class="card-header" id="headingOne">
                <h5 class="mb-0">
                    <h4 data-target="#'. $tipo_cita[0]['Nombre'] .'">
                        ' . $tipo_cita[0]['Nombre'] . ' ' . $personal[0]['Nombre'] . ' ' . $cita['Hora'] . '
                    </h4>
                </h5>
            </div>
            <div id="'. $tipo_cita[0]['Nombre'] .'" class="collapse show" aria-labelledby="headingOne" data-parent="#accordion">
                <div class="card-body"> - 
                    ' . $cita['informe'] .
                </div>
            </div>
        </div>';
    }
?
</div>
```

La etiqueta Mis Citas

En esta página el código esta realizado como la página la página anterior “Mis informes” la única diferencia es que contiene el botón borrar y le dice en la cabecera de la página si tiene alguna cita proximamente.

```

if($citas)
    echo '<p>Tienes una cita programada</p>';
else
    echo '<p>No tienes ninguna cita programada</p>';
?>
    </div>
    <?php
        //por cada cita
    foreach ($citas as $cita) {
        //obtener el tipo de cita y el personal por id
        $tipo_cita = crud_select('Tipocita', 'idTipocita',$cita['idTipocita']);
        $personal = crud_select('Personal', 'idPersonal',$cita['idPersonal']);
        //para cada citá se pinta un bloque de cita
        echo '<div class="tarjeta-citas">
            Próxima cita<br>
            Tu proxima cita es ' . $tipo_cita[0]['Nombre']. ' con el especialista ' . $personal[0]['Nombre'] . ' ' . $personal[0]['Apellido'] . ' el ' . $cita['Hora'] . '<br>
            <form action="" method="GET"><input type="hidden" name="idCita" value=' . $cita['idCita'] . '">
            <button onclick="confirm(\';Estás seguro de borrar el registro?\');" class="btn btn-primary" type="submit" name="borrar" value="Borrar">Borrar</button>
        </form>';
    </div>';
?>
```

Mediante el evento onclick realiza la confirmación de que va a ser eliminada y mediante un submit se acciona el botón con name= “borrar” y con el campo “idCita” oculto con hidden elimina el registro.



La etiqueta Pedir cita

Para la creación de esta página la he realizado mediante cinco pasos, en primer lugar la página principal que contiene el html, para seleccionar la fecha y la hora que el usuario decida o si

quiere lo antes posible, para lo antes posible por defecto da las citas al día siguiente a las 09:00h, esta función podría mejorarse en un futuro.

```
</nav>
<!-- Botón que abre el Modal -->


<h3 class="text-center mb-4">Seleccione los datos para elegir su cita:</h3>

    <div class="step active" id="step-1">
        <label for="fecha" class="form-label">Paso 1: Selecciona una fecha</label>
        <form action="pedir_cita_p2.php" method="post" name="mostrar-datos-usuario">
            <div class="input-group mb-3">
                <input type="datetime-local" class="form-control" id="fecha" name="fecha"
                    aria-describedby="fecha-ayuda">
                <button type="submit" class="btn btn-primary">Lo antes
                    posible</button>
            </div>
            <small id="fecha-ayuda" class="form-text text-muted">Selecciona una fecha disponible.</small>
            <button type="submit" id="siguientep1" name="siguientep1" value="siguientep1"
                class="btn btn-primary">Siguiiente</button>
        </form>
    </div>


```



Una vez seleccionada la fecha y la hora se pulsa el botón siguiente y pasa a la página siguiente, la fecha seleccionada anteriormente se guarda mediante hidden y obtengo los departamentos.

```
<div class="container">
    <h3 class="text-center mb-4">Seleccione los datos para elegir su cita:</h3>
    <div class="step" id="step-2">
        <label for="especialidad" class="form-label">Paso 2: Selecciona una especialidad hospitalaria</label>
        <form action="pedir_cita_p3.php" method="post" name="mostrar-datos-usuario">
            <?php
                // se ponen los datos del paso anterior en hidden para pasarlos al paso siguiente al pulsar siguiente
                echo '<input type="hidden" name="fecha" value="' . $fecha . '>';
            ?>
            <select class="form-select mb-3" name="departamento" id="departamento" required>
                <?php
                    // se pintan todos los departamentos
                    $departamentos=crud_get_all('departamento');
                    foreach ($departamentos as $departamento) {
                        echo '<option selected value=' . $departamento['idDepartamento'] . '>' . $departamento['Nombre'] . '</option>';
                    }
                ?>
            </select>
            <button type="submit" id="anteriorp1" name="anteriorp1" value="anterior1"
                class="btn btn-secondary">Anterior</button>
            <button type="submit" id="siguientep2" name="siguientep2" value="siguientep2"
                class="btn btn-primary ms-3">Siguiiente</button>
        </form>
    </div>

```



Pasa a la página siguiente guardando los datos obtenidos anteriormente mediante hidden y obtengo los campos de la tabla ciudad.

```

<!-- Botón que abre el Modal -->


<h3 class="text-center mb-4">Seleccione los datos para elegir su cita:</h3>
    <div class="step" id="step-3">
        <label for="localidad" class="form-label">Paso 3: Selecciona una localidad</label>
        <form action="pedir_cita_p4.php" method="post" name="mostrar-datos-usuario">
            <?php
                //se ponen los datos del paso anterior en hiddens para pasarlo al paso siguiente

            echo '<input type="hidden" name="fecha" value="" . $fecha . ">';
            echo '<input type="hidden" name="departamento" value="" . $departamento . ">';
            ?>
            <div class="row">
                <select class="form-select mb-3" name="ciudad" id="ciudad" required>
                    <option value="">Selecciona una localidad...</option>
                    <?php
                        // se pintan todos los departamentos
                        $ciudades=crud_get_all('ciudad');
                        foreach ($ciudades as $ciudad) { //Recorre las ciudades
                            echo '<option selected value=' . $ciudad['idCiudad'] . '>' . $ciudad['Nombre'] . '</option>; //Imprime
                        }
                    ?>
                </select>
            </div>
            <button type="submit" class="btn btn-secondary">Anterior</button>
            <button type="submit" id="siguientep3" name="siguientep3" value="siguientep3"
                    class="btn btn-primary ms-3">Siguiente</button>
        </form>
    </div>


```



Se selecciona el nombre del centro con el mismo proceso anteriormente mencionado.

```
<div class="container">
    <h3 class="text-center mb-4">Seleccione los datos para elegir su cita:</h3>
    <div class="step" id="step-3">
        <label for="localidad" class="form-label">Paso 4: Selecciona un centro</label>
        <form action="pedir_cita_p5.php" method="post" name="mostrar-datos-usuario">
            <?php
                // se ponen los datos del paso anterior en hiddens para pasarlo al paso siguiente

            echo '<input type="hidden" name="fecha" value="" . $fecha . ">';
            echo '<input type="hidden" name="ciudad" value="" . $ciudad . ">';
            echo '<input type="hidden" name="departamento" value="" . $departamento . ">';
        ?>
            <div class="row">
                <select class="form-select mb-3" id="centro" name="centro" required>
                    <option value="">Selecciona un centro...</option>
                <?php
                    // se pintan todos los centros (faltaria buscarlos solo por la ciudad seleccionada en el paso anterior)
                    $centros=crud_get_all('centro');
                    foreach ($centros as $centro) { //Recorre las centros
                        echo '<option selected value=' . $centro['idCiudad'] . '>' . $centro['Nombre'] . '</option>; //Impr
                    }
                ?>
                </select>
            </div>
            <button type="submit" class="btn btn-secondary">Anterior</button>
            <button type="submit" id="siguientep4" name="siguientep4" value="siguientep4"
                class="btn btn-primary ms-3">Siguiente</button>
        </form>
    </div>
</div>
```



Se selecciona el tipo de cita.

```
<div class="container">
    <h3 class="text-center mb-4">Seleccione los datos para elegir su cita:</h3>
    <div class="step" id="step-5">
        <label for="localidad" class="form-label">Paso 5: Selecciona el tipo de cita</label>
        <form action="confirmar_cita.php" method="post" name="mostrar-datos-usuario">
            <?php
                // se ponen los datos del paso anterior en hiddens para pasarlo al paso siguiente

            echo '<input type="hidden" name="fecha" value="" . $fecha . ">';
            echo '<input type="hidden" name="centro" value="" . $centro . ">';
            echo '<input type="hidden" name="ciudad" value="" . $ciudad . ">';
            echo '<input type="hidden" name="departamento" value="" . $departamento . ">';
        ?>
            <div class="row">
                <select class="form-select mb-3" id="tipo_cita" name="tipo_cita" required>
                    <option value="">Selecciona un tipo de cita...</option>
                <?php
                    // se pintan todos los tipos de cita (faltaria buscarlos por departamento)
                    $tipos_citas=crud_get_all('tipoCita');
                    foreach ($tipos_citas as $tipo_cita) { //Recorre los tipos de cita
                        echo '<option selected value=' . $tipo_cita['idTipocita'] . '>' . $tipo_cita['Nombre'] . '</option>';
                    }
                ?>
                </select>
            </div>
            <button type="submit" class="btn btn-secondary">Anterior</button>
            <button type="submit" id="confirmar" name="confirmar" value="confirmar"
                class="btn btn-primary ms-3">Confirmar</button>
        </form>
    </div>
</div>
```



Y por último se confirma la cita.



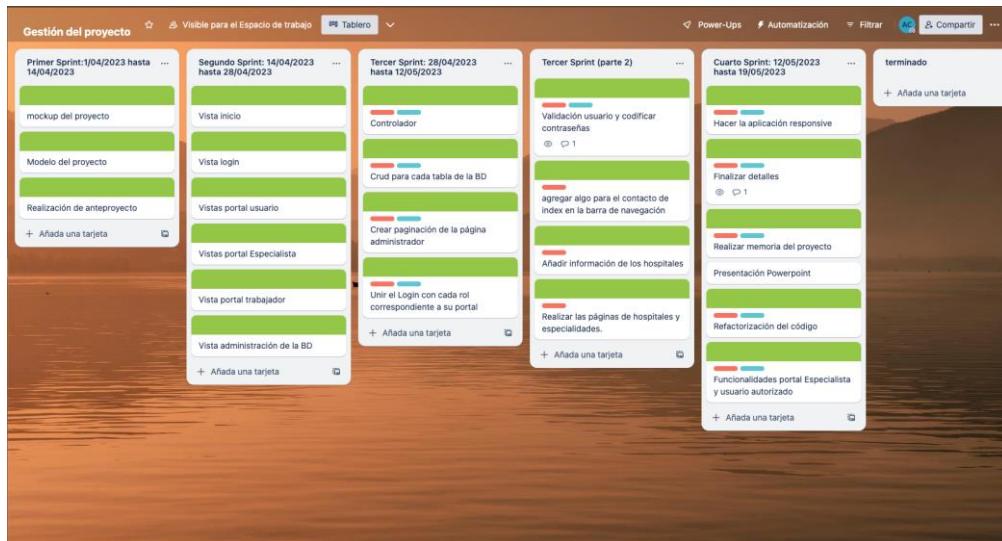
16. Retrospectiva Sprint 3

Las consultas podrían haberlas realizado como el CRUD con un patrón y así no tener que hacer una para cada una, pero con el corto tiempo que hay para la aplicación no me ha sido posible ya que para el patrón del CRUD y la unión de las tablas M:M me ha llevado más tiempo de lo que esperaba.

Para el siguiente sprint, realizar las consultas para las funcionalidades, en los apartados pedir cita del portal Familiar, Usuario autorizado y Especialista, se realizará el mismo código, mediante el Dni del paciente se le dará cita.

Por último, comentar el código y finalizar detalles.

17. Sprint 4 Backlog



En este sprint se ha realizado las funcionalidades para el portal del especialista y el portal del usuario autorizado.

También se iba a realizar el portal del familiar adjunto al usuario, pero por falta de tiempo se quedará para mejoras futuras y así poder ofrecer más servicios al usuario.

Por lo que en el portal del usuario también se han realizado modificaciones con la tarjeta familiar, ha sido eliminada, quedando el portal de la siguiente forma.



La finalización de detalles como redirigir a cada usuario a su portal correspondiente, realizar la aplicación responsive y refactorización del código.

18. Descripción del trabajo en Sprint 4

Portal Especialista

La vista de este portal ha sido modificada, anteriormente en la parte izquierda mostraba donde el especialista podía consultar la agenda introduciendo la fecha y en la parte derecha un iframe con donde poder mostrar los datos del usuario, pues bien, el iframe ha sido eliminado y además he añadido una tarjeta en la que cuando realiza la acción del evento onclick le lleva redirección a la página dar cita.

Para la realización de este portal, en la parte izquierda he creado una tabla, el usuario introduce una fecha y mediante esta fecha le proporciona los pacientes que debe atender en ese día.

Para buscar la fecha que se encuentra en la vista "cita", en primer lugar, mediante la variable de sesión que contiene el idUsuario obtengo el dni del usuario que está como clave primaria en la vista usuario.

```
//se obtienen el especialista por id usuario guardado en el login y se coje el DNI
$usuario = crud_select('usuario', 'idUsuario', $_SESSION['idUsuario']);
$_SESSION['dni'] = $usuario[0]['Dni'];
```

Una vez obtenido el DNI del usuario obtengo el idPersonal de la vista “personal”, mediante la creación de una consulta.

```
function id_Personal($dni){
    try {
        // Conectar a la base de datos utilizando PDO
        $dsn = 'mysql:host=localhost;dbname=hospital';
        $username = 'root';
        $password = '';
        $pdo = new PDO($dsn, $username, $password);
        //realiza la consulta

        $sql = "SELECT * from personal
        where personal.Dni = '$dni';";

        $stmt = $pdo->prepare($sql); //prepara la consulta
        $stmt->execute(); //ejecuta

        // Retornar los resultados
        return $stmt->fetchAll(PDO::FETCH_ASSOC); //devuelve todos los registros
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
}
```

Realizo la llamada a la función y ya obtengo el idPersonal.

```
require('../modelo/especialista_agenda.php');
$personal=id_Personal($_SESSION['dni'] = $usuario[0]['Dni']);
foreach($personal as $campo){
    $idPersonal=$campo['idPersonal'];
}
```

para poder obtener las citas de la vista “Cita” que esta relacionada con la tabla Personal mediante el campo ‘idPersonal’, realizo otra función a la que le entra la fecha y el idPersonal.

```
if(isset($_POST['fecha'])){
    $fecha = $_POST['fecha'];
    if (isset($_POST['buscar_agenda'])) {
        try{
            // $arreglo_fecha = date('Y-m-d', strtotime($fecha));
            $busqueda=obtener_cita_por_fecha($fecha,$idPersonal);
            foreach ($busqueda as $resultado) {
                // al hacer onclick muestra el nombre y el apellido del usuario seleccionado
                echo ' <tr onclick="seleccionar"' . $resultado['idCita'] . ',' . $resultado['nombre'] . ',' . $resultado['apellido'] . '>';
                echo " <td>" . $resultado['nombre'] . "</td>";
                echo " <td>" . $resultado['apellido'] . "</td>";
                echo " <td>" . $resultado['dni'] . "</td>";
                echo " <td>" . $resultado['fecha'] . "</td>";
            }
        }catch(PDOException $e) {
            echo 'Error en la búsqueda de la ciudad: ' . $e->getMessage();
        }
    }
}
```

De este modo al introducir la fecha la muestra en la parte izquierda del portal.

Nombre	Apellidos	Dni	Hora
Ana	López Suarez	88756952N	2023-05-31 03:00:00
Ana	Martínez Guerra	77777777L	2023-05-31 11:20:00

HTML de la parte izquierda.

```


<div class="contenedor_personal_autorizado_paciente columna_1">
    <div>
      <!--div para elegir día y que el especialista vea los pacientes de el día en concreto-->
      <form method="post" action="#">
        <div class="texto_titulo">
          <h1>Agenda</h1>
          <input type="date" class="form-control" id="fecha" name="fecha"
            aria-describedby="fecha-ayuda">
          <button class="btn btn-primary" style="margin-top:10px" type="submit" id="buscar_agenda"
            name="buscar_agenda">Buscar</button>
        </div>
      </form>
      <div class="con_tabala_agenda">
        <!--Bucle para mostrar todos los usuarios que tengan cita el día seleccionado----->
        <table class="tab_especialista texto">
          <tr>
            <th></th>
            <th>Pacientes</th>
            <th></th>
            <th></th>
          </tr>
          <tr>
            <td>
              <tr>
                <th>Nombre</th>
                <th>Apellidos</th>
                <th>Dni</th>
                <th>Hora</th>
              </tr>
            </td>
          </tr>
        </table>
      </div>
    </div>
  </div>


```

Para la parte derecha, mediante la función onclick, llamo a la función JavaScript seleccionar para que cuando el especialista realice un click encima del nombre del usuario este aparezca en los inputs el nombre y el apellido.

```

function seleccionar(idCita, nombre, apellido) {
  document.getElementById('nombre').value = nombre;
  document.getElementById('apellido').value = apellido;
  document.getElementById('idCita').value = idCita;

}

//al hacer onclick muestra el nombre y el apellido del usuario seleccionado
echo " <tr onclick='seleccionar(\"" . $resultado['idCita'] . "\",\" " . $resultado['nombre'] . "\",\" " . $resultado['apellido'] . "\");'>";
echo " <td>" . $resultado['nombre'] . "</td>";

```

Nombre	Apellidos	Dni	Hora
Ana	López Suarez	88756952N	2023-05-31 03:00:00
Ana	Martínez Guerra	77777777L	2023-05-31 11:20:00

Mediante el campo idCita que ya he obtenido con la consulta obtener_cita_por_fecha(), realizo una modificación en el registro de dicho id para añadir el informe que el especialista escribirá en cuadro de texto para finalizar la cita.

```
// se obtiene el id del personal mediante el Dni
if (isset($_POST['registrar'])) {
    $idcita = $_POST['idcita'];
    $informe = $_POST['informe'];

    try { //actualiza los datos por el id de provincia que se ha guardado en el hidden
        crud_actualizar('cita', array('informe' => $informe), "idCita = $idCita");
    } catch (PDOException $e) {
        echo 'Error al editar la cita: ' . $e->getMessage();
    }
}
```

Por último, el ícono de dar cita, redirige a la página dar_cita.php.

The 'Agenda' page displays a table of patient appointments:

Nombre	Apellidos	Dni	Hora
Ana	López Suarez	88756952N	2023-05-31 03:00:00
Ana	Martínez Guerra	77777777L	2023-05-31 11:20:00

The 'Informe del paciente' page shows a form with fields for Name and Surname, and a note area containing: "La paciente presenta fuertes dolores de cabeza".

Portal usuario autorizado

La vista de este portal también ha tenido modificaciones, ahora me parece más dinámica y más manejable por el usuario.

Le he creado tarjetas en las que el usuario, en este caso el usuario con el rol “usuario autorizado”, mediante el evento onclick le mostrará la acciones que puede realizar en el portal.

Cada tarjeta contiene un div oculto mediante la clase de CSS “oculta” y mediante el evento onclick se muestra cada funcionalidad que realiza.

```
.oculta{
    display:none;
}
```

Mediante JavaScript pongo a la espera el contenido de cada div relacionado con cada tarjeta.

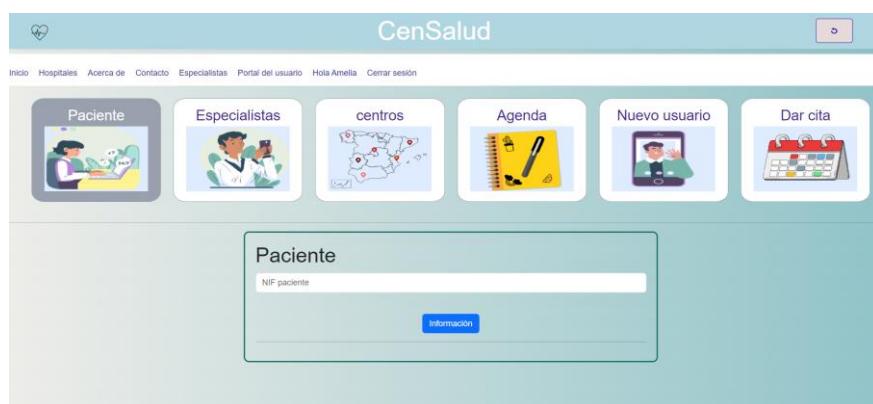
```
function mostrar_datos (div){
    var continformación_paciente = document.getElementById('información_paciente');
    if (!continformación_paciente.classList.contains('oculta')) {
        continformación_paciente.classList.toggle('oculta');
    }
    var contEspecialistas_centros = document.getElementById('Especialistas_centros');
    if (!contEspecialistas_centros.classList.contains('oculta')) {
        contEspecialistas_centros.classList.toggle('oculta');
    }
    var contagenda_especialista = document.getElementById('agenda_especialista');
    if (!contagenda_especialista.classList.contains('oculta')) {
        contagenda_especialista.classList.toggle('oculta');
    }
    var continformacion_centros = document.getElementById('informacion_centros');
    if (!continformacion_centros.classList.contains('oculta')) {
        continformacion_centros.classList.toggle('oculta');
    }
    var contenedor_formulario_datos=document.getElementById(div);
    contenedor_formulario_datos.classList.toggle("oculta");
}
```

La tarjeta paciente se relaciona con el div con identificación "información_paciente".

```
<div class="menu-tarjetas">
    <div class="tarjeta" onclick="mostrar_datos ('información_paciente')">
        Paciente <br>
        
    </div>
```

Este div solo contiene un input de tipo text donde el especialista introduce el DNI del paciente y un botón de tipo submit para buscar la información.

```
<div class="contenedor_personal_autorizado <?php if (isset($_POST['btn_informacion'])) { echo '';} else { echo 'oculta'; } ?>" id="información_paciente">
    <h1>Paciente</h1>
    <form method='post' action="#">
        <div class="texto_titulo">
            <input placeholder="NIF paciente" required type="text" class="form-control" id="dni_usuario" name="dni_usuario" aria-describedby="basic-addon3"><br>
            <button type="submit" name="btn_informacion" id="btn_informacion" class="btn btn-primary">
                Información
            </button>
        <br>
        <hr>
    </div>
```



Al introducir el DNI obtiene las citas de este usuario, sus informes clínicos.

```
<?php
require_once('.../modelo/personal_autorizado/consulta_autorizado_paciente.php');//citas e informes de paciente

if(isset($_POST['dni_usuario'])){
    $dni=$_POST['dni_usuario'];
    //busca el nombre del usuario según el dni para mostrarlo en la cabecera de la búsqueda
    $usu=crud_select('Usuario', 'Dni', $dni);
    foreach($usu as $campo){
        $nom_usu=$campo['Nombre'];
        $apel_usu=$campo['Apellido'];
    }
    echo " $nom_usu " . " . " . "$apel_usu ";

    if (isset($_POST['btn_informacion'])) [
        try{
            $busqueda=obtener_citas_por_dni('cita',$dni);
            $busqueda2=obtener_citas_por_dni('hospital.familiar',$dni);

        }catch(PDOException $e) {
            echo 'Error en la búsqueda de las citas: ' . $e->getMessage();
        }
        echo "<table class='table table-hover tab_especialista texto'>";
        echo "<thead><tr><th>Cita</th><th>Informe Clínico</th></tr></thead>";
        echo "<tbody>";
        foreach ($busqueda as $resultado) {
            echo "<tr class='celdas'><td>" . $resultado['Hora'] . "</td>";
            echo "<td>" . $resultado['Informe'] . "</td></tr>";
        }

        echo "</tbody>";
        echo "</table>";
    ]
}
```



Como ya he comentado anteriormente esta información se obtiene mediante consulta sql, en este caso, si que he realizado una de consulta para mostrar las consultas que contiene el idUsuario que se relaciona con la vista “cita”.

```

<?php
//obtener citas del paciente
function obtener_citas_por_dni($tabla,$dni){
    try {
        // Conectar a la base de datos utilizando PDO
        $dsn = 'mysql:host=localhost;dbname=hospital';
        $username = 'root';
        $password = '';
        $pdo = new PDO($dsn, $username, $password);
        //realiza la consulta
        $sql = "SELECT $tabla.* from $tabla
inner join usuario on usuario.idUsuario=$tabla.idUsuario
where usuario.Dni = '$dni';";
        $stmt = $pdo->prepare($sql);//prepara la consulta
        $stmt->execute();//ejecuta

        // Retornar los resultados
        return $stmt->fetchAll(PDO::FETCH_ASSOC); //devuelve todos los registros
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
}

```

La tarjeta especialista muestra los especialistas que contiene un centro según el nombre del centro, la tarjeta centros muestra todos los centros que se encuentran en la provincia introducida y la tarjeta agenda muestra la agenda de una fecha en concreto introducida de un especialista que se busca mediante el nombre y apellido, la búsqueda en la base de datos se realiza de forma similar a la mencionada en el portal del especialista.

Estas tarjetas siguen el mismo proceso que la tarjeta paciente.

La tarjeta nuevo usuario redirige a la página registro.php y la tarjeta citas redirige a la página pedir_cita.php.

Finalización de detalles

Para la finalización de detalles he realizado pruebas para comprobar que todo funciona correctamente.

En el portal del usuario autorizado cuando selecciona la tarjeta nuevo usuario o citas para dar cita, le lleva a la página registro.php y pedir_cita.php, el botón volver según el usuario que sea le devuelve al portal del rol que entra por la variable de sesión, esto también se ha realizado para el especialista cuando selecciona la tarjeta pedir cita.

```

//se obtienen el especialista por id usuario guardado en el login y se coje el DNI
$usuario = crud_select('usuario', 'idUsuario', $_SESSION['idUsuario']);
$_SESSION['dni'] = $usuario[0]['Dni'];
$_SESSION['rol'] = $usuario[0]['Rol'];

```

```
<div class="button-container">
    <?php if($rol =='Usuario'){//si el usuario su rol es usuario le lleva al portal del usuario
        echo '<a href="portal_usuario.php" class="c-button user-button"></a>';
    }
    if($rol =='Usuario_autorizado'){//si el usuario su rol es usuario_autorizado le lleva al portal del personal autorizado
        echo '<a href="Personal_autorizado/portal_usuario_autorizado.php" class="c-button user-button"></a>';
    }
    if($rol =='Especialista'){//si el usuario su rol es especialista le lleva al portal del especialista
        echo '<a href="Especialista/portal_especialista.php" class="c-button user-button"></a>';
    }
    ?>
</div>
```

Además, usando la variable de sesión en la página pedir_cita, según el usuario que entre le mostrará un div con un input para que introduzcan el DNI y a través del DNI el especialista y el usuario autorizado pueda dar cita al usuario.

Si el rol que entra es distinto a usuario mostrará el div, ya que si es usuario directamente mediante la variable de sesión con idUsuario le proporcionara la cita.

```
<?php if($rol !='Usuario'){
    echo '<input type="text" class="form-control" id="dni" name="dni"
    aria-describedby="fecha-ayuda" placeholder="DNI">';
}?
```

Por último, en la confirmación de la cita para ver la cita, según el rol le lleva a su portal.

```
</div>
<?php if($rol =='Usuario'){//según el rol le lleva a su portal
echo '<form action="mis_citas.php" method="post" name="mostrar-datos-usuario">
    <button type="submit" id="confirmar" name="confirmar" value="confirmar"
    class="btn btn-primary ms-3">Mis Citas</button></form>';
} if($rol =='Usuario_autorizado'){
echo'<a href="Personal_autorizado/portal_usuario_autorizado.php">
    <button type="submit" id="confirmar" name="confirmar" value="confirmar"
    class="btn btn-primary ms-3">Mis Citas</button>
</a>';
}
if($rol =='Especialista'){
    echo'<a href="Especialista/portal_especialista.php">
        <button type="submit" id="confirmar" name="confirmar" value="confirmar"
        class="btn btn-primary ms-3">Mis Citas</button>
    </a>';
}
```

Y para concluir con la descripción se ha realizado el código responsive con media queries desde CSS.

19. Documentación Incremento 4

Portal especialista

Se pone la fecha en la parte izquierda y se le da a buscar directamente aparecen los pacientes.

The screenshot shows the CenSalud portal interface. On the left, there is an 'Agenda' section with a date input field ('dd/mm/aaaa') and a 'Buscar' button. Below it, a table titled 'Pacientes' lists columns: Nombre, Apellidos, Dni, and Hora. On the right, there is an 'Informe del paciente' section with fields for Nombre and Apellido, and a 'Dar cita' button.

The screenshot shows the same CenSalud portal interface as above, but now the 'Agenda' section displays a list of patients with their details: Ana López Suarez, Ana Martínez Guerra, Carmen López Bravo, and another entry partially visible. The 'Informe del paciente' section remains the same.

Al seleccionar un paciente mediante el evento onclick aparece el nombre y apellido en la parte derecha de la pantalla.

The screenshot shows the CenSalud portal interface. The 'Agenda' section lists patients. The row for 'Carmen López Bravo' is highlighted in green. The 'Informe del paciente' section on the right shows the selected patient's name ('Carmen López Bravo') and has a 'Dar cita' button.

Se escribe el informe y se registra.

Como se puede observar no aparece nada en esa cita en concreto.

Result Grid					
idCita	Hora	idPersonal	idUsuario	idTipoCita	Informe
22	2023-05-19 00:00:00	8	7	3	NULL
23	2023-05-19 00:00:00	8	7	3	NULL
24	2023-05-25 00:00:00	8	7	3	NULL
25	2023-05-31 03:00:00	10	7	3	Informe del paciente: inform°1 ...
26	2023-05-17 00:00:00	8	7	3	NULL
27	2023-05-31 11:20:00	10	8	3	NULL
28	2023-05-25 00:00:00	8	5	3	NULL
29	2023-05-19 00:00:00	8	5	3	NULL
30	2023-05-20 20:00:00	8	23	3	NULL
31	2023-05-24 00:00:00	8	8	3	NULL
32	2023-05-31 16:33:00	12	8	2	NULL
33	2023-05-31 17:12:00	10	1	2	NULL
34	2023-05-31 18:19:00	10	2	1	NULL
*	NULL	NULL	NULL	NULL	NULL

Y al registrar este se guarda en la base de datos.

idCita	Hora	idPersonal	idUsuario	idTipoCita	Informe
22	2023-05-19 00:00:00	8	7	3	NULL
23	2023-05-19 00:00:00	8	7	3	NULL
24	2023-05-25 00:00:00	8	7	3	NULL
25	2023-05-31 03:00:00	10	7	3	Informe del paciente: inform°1 ...
26	2023-05-17 00:00:00	8	7	3	NULL
27	2023-05-31 11:20:00	10	8	3	NULL
28	2023-05-25 00:00:00	8	5	3	NULL
29	2023-05-19 00:00:00	8	5	3	NULL
30	2023-05-20 20:00:00	8	23	3	NULL
31	2023-05-24 00:00:00	8	8	3	NULL
32	2023-05-31 16:33:00	12	8	2	NULL
33	2023-05-31 17:12:00	10	1	2	NULL
34	2023-05-31 18:19:00	10	2	1	NULL
*	NULL	NULL	NULL	NULL	NULL

Al pulsar el botón cita, le lleva a la página dar cita

Seleccione los datos para elegir su cita:

Paso 1: Selecciona una fecha

DNI: 77456212L 07/06/2023 12:37 Lo antes posible

Selecciona una fecha disponible: [Siguiente](#)

Como se puede comprobar aparece el input para introducir el dni del paciente

Seleccione los datos para elegir su cita:

Paso 2: Selecciona una especialidad hospitalaria

Urología

[Anterior](#) [Siguiente](#)

Seleccione los datos para elegir su cita:

Paso 3: Selecciona una localidad

Marbella

[Anterior](#) [Siguiente](#)

Selección de los datos para elegir su cita:

Paso 4: Selecciona un centro

Bienestar

Anterior Siguiente

Selección de los datos para elegir su cita:

Paso 5: Selecciona el tipo de cita

Revisión

Anterior Confirmar

Por último, se confirma la cita y como se puede comprobar se ha añadido en la base de datos.

	idCita	Hora	idPersonal	idUserario	idTipoCita	Informe
•	24	2023-05-25 00:00:00	8	7	3	NULL
	25	2023-05-31 03:00:00	10	7	3	Informe del paciente: informº1 ...
	26	2023-05-17 00:00:00	8	7	3	NULL
	27	2023-05-31 11:20:00	10	8	3	NULL
	28	2023-05-25 00:00:00	8	5	3	NULL
	29	2023-05-19 00:00:00	8	5	3	NULL
	30	2023-05-20 00:00:00	8	23	3	NULL
	31	2023-05-24 00:00:00	8	8	3	NULL
	32	2023-05-31 16:33:00	12	8	2	NULL
	33	2023-05-31 17:12:00	10	1	2	La paciente recibe el alta. ...
	34	2023-05-31 18:19:00	10	2	1	NULL
	35	2023-06-02 00:00:00	8	1	3	NULL
	36	2023-06-07 00:00:00	8	1	2	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL

Se ha confirmado su cita:

Cita Confirmada:

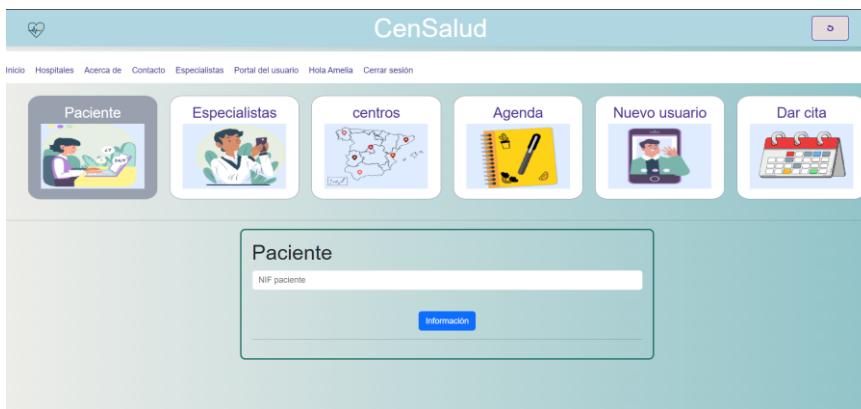
Mis Citas

Al pulsar el botón le vuelve a llevar al portal del especialista.

Portal personal autorizado



Al seleccionar la tarjeta paciente aparece un menú donde el usuario autorizado escribirá el DNI del paciente y le mostrará la información.



Si pulsa la tarjeta especialista, le aparece un menú para introducir el nombre del centro y le muestra todos los especialistas que tiene el centro.

The screenshot shows the CenSalud website with a light blue header bar. Below it is a navigation menu with links: Inicio, Hospitales, Acerca de, Contacto, Especialistas, Portal del usuario, Hola Amelia, and Cerrar sesión. A purple ribbon-like graphic is positioned above the main content area. The main content area features a row of six icons: Paciente, Especialistas, centros, Agenda, Nuevo usuario, and Dar cita. Below this is a search box labeled 'Especialistas' with a dropdown menu showing 'Centro' and a 'Buscar' button.

This screenshot shows the search results for 'Especialistas'. The search term 'Bienestar' is entered in the search bar. The results table has columns: Especialista, Especialidad, and Centro. One result is listed: Isabel, Urología, Bienestar.

Especialista	Especialidad	Centro
Isabel	Urología	Bienestar

Si pulsa tarjeta centros, aparece un menú en el que mediante la provincia le mostrará los centros que se encuentra en esa provincia.

The screenshot shows the CenSalud website with the same header and menu as the previous screenshots. The 'centros' icon is highlighted in grey. Below it is a search box labeled 'Centros' with a dropdown menu showing 'Provincia' and a 'Buscar' button.

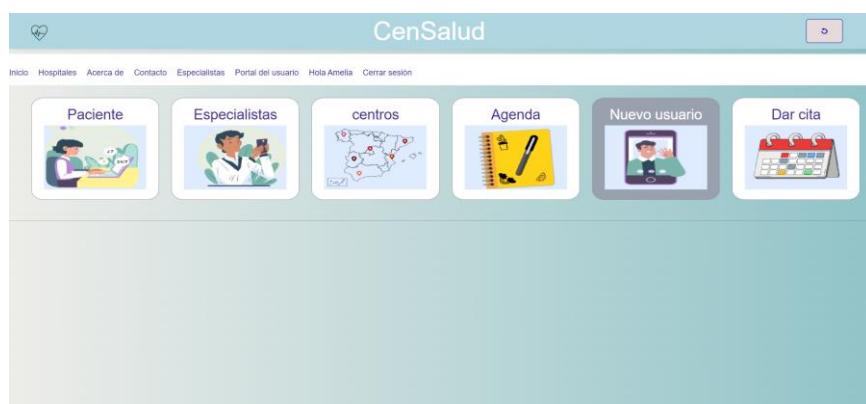
This screenshot shows the search results for 'Centros'. The search term 'Malaga' is entered in the search bar. The results table has columns: Ciudad, Dirección, and Teléfono. Three entries are listed: Torremolinos (C/ hueva, 17, 952223344), Torremolinos (C/Traviata, 20, 952336644), and Marbella (C/Federico, 17, 954987741).

Ciudad	Dirección	Teléfono
Torremolinos	C/ hueva, 17	952223344
Torremolinos	C/Traviata, 20	952336644
Marbella	C/Federico, 17	954987741

Si pulsa la tarjeta agenda mediante el nombre y apellido del especialista junto con la fecha del día que quiere consultar le muestra la agenda del especialista, es igual al lo que se realiza en el portal del especialista para mostrar la agenda.



La tarjeta nuevo usuario le lleva a generar un nuevo registro.



CenSalud

Registro de usuario

Nombre
Apellidos
Documento (DNI)
Fecha de Nacimiento
Teléfono
Email
Contraseña

Enviar

Por último, la tarjeta dar cita, realiza la misma funcionalidad que en el portal del especialista.



Seleccione los datos para elegir su cita:

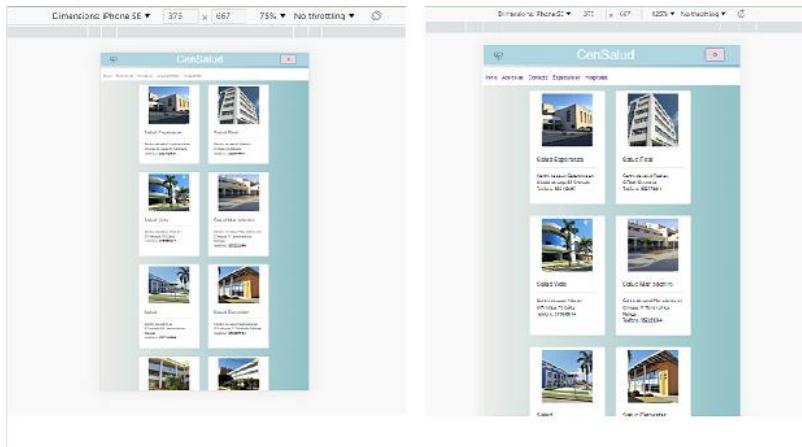
Paso 1: Selecciona una fecha

DNI	dd/mm/aaaa --::--	<input type="button" value=""/>
-----	-------------------	---------------------------------

Selecciona una fecha disponible.

Responsive

Para hacer la aplicación responsive he usado media queries en CSS.



20. Retrospectiva Sprint 4

En este Sprint he realizado el funcionamiento del portal del Especialista, Portal usuario autorizado y finalizado detalles en el código.

Para las funcionalidades que se realizan a través de la base de datos, se han realizado consultas específicas para esa funcionalidad, con más tiempo para el desarrollo de la aplicación, se podría haber refactorizado un poco más esta parte.

En conclusión, en este sprint he finalizado la realización de la aplicación, siempre se le pueden realizar mejoras y añadir funcionalidades, pero con más tiempo ya que me parece un proyecto muy ambicioso.

21. Manual de uso y puesta en marcha

Puesta en marcha

El portal del Administrador ha sido creado para el mantenimiento de la base de datos, de una forma sencilla y dinámica.

Cada centro se encuentra en una ciudad y cada ciudad está situada en una provincia, por lo que la vista de la BD “ciudad” contiene el campo idProvincias como FK y ese campo lo une con la vista “provincias” que como PK tiene el campo idProvincias, por este motivo si se quiere crear una ciudad primero debe de existir la provincia donde está situada esa ciudad.

Provincia		
Málaga	<button>Borrar</button>	
Jaén	<button>Borrar</button>	
1 2 3 4 Siguiente		
Nombre:	<input type="text" value="Albacete"/>	<button>Registrar</button>

Lo único que tiene que hacer el administrador es escribir el nombre de la provincia y registrarla

Provincia	
Albacete	<button>Borrar</button>

Anterior 1 ... 3 4 5

Nombre: Registrar

Una vez obtenida la provincia, ya se puede crear la ciudad, escribiendo el nombre de la ciudad y seleccionando a la provincia a la que pertenece.

Ciudades	Provincia	Borrar
Torremolinos	Málaga	<button>Borrar</button>
Martos	Jaén	<button>Borrar</button>

1 2 3 ... 5 Siguiente

Ciudad: Almansa Provincia: Albacete Málaga Jaén Almería Granada Sevilla Huelva Cádiz Córdoba Albacete

Se registra y ya se obtiene la ciudad con su provincia correspondiente.

Ciudades	Provincia	Borrar
Almansa	Albacete	<button>Borrar</button>

Anterior 1 ... 4 5 6

Ciudad: Provincia: Albacete

Ya creada la ciudad donde se situará el centro se puede crear el centro perfectamente.

Nombre	Direccion	Telefono	Ciudad	Borrar
Mar adentro	C/nueva,17	952223344	Torremolinos	<button>Borrar</button>
Salud	C/Traviata,20	952336644	Torremolinos	<button>Borrar</button>

1 2 3 ... 5 Siguiente

Centro: Salud Almansa Dirección: C/ Pedreria,45 Telefono: 937159247 Ciudad: Almansa

Se registra y ya tenemos el centro guardado en la base de datos.

Nombre	Direccion	Telefono	Ciudad	Borrar
Salud Almansa	C/ Pedreria,45	937159247	Almansa	<button>Borrar</button>

Anterior 1 ... 4 5 6

Centro: Dirección: Telefono: Ciudad: Almansa

Ahora bien, cada centro contiene departamentos, pero que pasa que un centro puede contener muchos departamentos y los mismos departamentos pueden estar en distintos centros, para la unión la vista “centro” y “departamento” se genera una vista intermedia “centro_departamento” donde contiene dos campos como FK uno de ellos es idCentro que lo une con la vista “centro” que contiene el campo idCentro como PK y el campo idDepartamento que lo une con la vista “departamento” la cual contiene un campo como PK idDepartamento.

Por lo que se crea un departamento en la vista “departamento”, si este no existe.

Nombre	Descripción	Borrar
Endocrinología	Nutrición	<button>Borrar</button>
Podología	Pies	<button>Borrar</button>

1 2 3 ... 9 Siguiente

Nombre: Psicología Descripción: Problemas mentales

Nombre	Descripción	Borrar
Hematología	Enfermedades sanguíneas	<button>Borrar</button>
Psicología	Problemas mentales	<button>Borrar</button>

Anterior 1 ... 7 8 9

Y una vez creado el departamento y el centro, en la sección centros del portal del administrador se relacionan ambos registros, el nombre del centro y el nombre del departamento.

Centro	Departamento
Mar adentro	Podología
Mar adentro	Dermatología
Salud	Oftalmología
Salud	Dentista
Bienestar	Urología

Centro: Salud Almansa Departamento: Psicología

Centro	Departamento	
Mar adentro	Podología	<button>Borrar</button>
Mar adentro	Dermatología	<button>Borrar</button>
Salud	Oftalmología	<button>Borrar</button>
Salud	Dentista	<button>Borrar</button>
Bienestar	Urología	<button>Borrar</button>
Salud Almansa	Psicología	<button>Borrar</button>

Centro: Salud Almansa Departamento: Psicología Registrar departamento en centro

Cada departamento tiene su correspondiente personal, por lo que pasa un caso similar al anteriormente explicado, donde cada departamento puede tener mucho personal y cada personal puede tener varios departamentos, para la unión de estas vistas se genera la vista “departamento_personal” con dos campos como FK, uno idPersonal que lo une con la vista “Personal” que contiene el campo idPersonal como PK, el campo idDepartamento de la vista “departamento_personal” como FK lo une con la vista “departamento” mediante el campo idDepartamento que contiene como PK.

Por lo que si no existe el personal que se quiere relacionar con el departamento se crea.

Personal

Nombre	Apellido	Telefono	Borrar
Isabel	Montes Alpes	125444215	<button>Borrar</button>
Eustaquio	Sánchez Ramírez	965412541	<button>Borrar</button>

1 2 3 ... 10 Siguiente

Nombre: Lourdes Apellido: Benitez Telefono: Sanz
 Fecha de nacimiento: 26/05/1995 Email: Lourdes@gmail.com
 Contraseña: **** DNI: 66315920G Usuario:
 Especialista
 Usuario Autorizado

El personal, también puede ser creado mediante la sección usuario que más adelante se detallara.

Personal

Nombre	Apellido	Telefono	Borrar
Lourdes	Benitez	Sanz	<button>Borrar</button>

Anterior 1 ... 9 10 11

Una vez creado el personal en el apartado departamentos del portal del administrador, se relaciona el personal con su respectivo departamento.

Psiquiatría	Estefanía Aguilera Monzon	Borrar
Neurología	Víctor Cruz López	Borrar
Hematología	Pablo Ruiz Sánchez	Borrar
Departamento: Psicología ▾ Personal: Lourdes Benitez ▾ Registrar personal en departamento		

Ya se puede comprobar que el registro se ha realizado correctamente.

Psicología	Lourdes Benitez	Borrar
------------	-----------------	---------------

Por otro lado, los departamentos contienen citas, estas citas pueden ser de tipo rutinarias, revisión, etc.

Y vuelve a ocurrir lo anteriormente mencionado, que un departamento puede tener varios tipos de citas y los tipos de citas pueden tener varios departamentos, por lo que genera la vista “departamento_tipoCita”, que contiene el campo idTipoCita como FK el cual se une con la vista “tipoCitas” mediante el campo idTipoCita que contiene como PK y el campo idDepartamento como FK que lo une con la vista “departamento”.

Si no existe el tipo de cita que se quiere usar, este se crea en el apartado tipo de cita.

Nombre	Duracion	Borrar
Rutinaria	5	Borrar
Anterior 1 2		
Nombre: Seguimiento	Duracion: <input type="text" value="10 minutos"/>	Registrar
Nombre: Duracion: <input type="text" value="10 minutos"/> Registrar		
Nombre	Duracion	Borrar
Rutinaria	5	Borrar
Seguimiento	10	Borrar
Anterior 1 2		
Nombre:	Duracion:	Registrar

En este mismo apartado se creará la relación del tipo de cita con el departamento.

Departamento	Tipo de Cita	
Urología	Primera consulta	Borrar
Urología	Revisión	Borrar
Urología	Rutinaria	Borrar
Departamento:	Psicología	
Tipo de cita:	Seguimiento	
		Registrar tipo de cita en departamento

Departamento	Tipo de Cita	
Urología	Primera consulta	Borrar
Urología	Revisión	Borrar
Urología	Rutinaria	Borrar
Psicología	Seguimiento	Borrar

Llegado a este punto, la vista “cita” se une mediante el campo idTipoCita que contiene como FK con la vista “tipoCitas”, además contiene el campo idPersonal también como FK que se une con la vista “personal” y el campo “idUsuario” también como FK que se une con la vista “usuario” mediante el campo “idUsuario” que contiene esta vista como PK.

Por lo que se escoge el día y hora de la cita, usuario, el especialista y tipo de cita.

Hora	Usuario	Personal	Tipo Cita	Borrar
2023-05-25 04:54:00	Amelia Ruiz Rodriguez	Eustaquio Sánchez Ramírez	Rutinaria	Borrar
2023-05-30 04:55:00	Matiasss Benítez Pérez	Mará Trinidad Los Santos Vega	Revisión	Borrar
1 2 3 ... 10 Siguiente				
Hora:	26/05/2023 20:56		Seguimiento	Registrar

De este modo la cita ya estaría generada.

Hora	Usuario	Personal	Tipo Cita	Borrar
2023-05-22 00:00:00	ana ana	Isabel Montes Alpes	Rutinaria	Borrar
2023-05-26 20:56:00	Carmen López Bravo	Lourdes Benitez	Seguimiento	Borrar
Anterior 1 ... 8 9 10				

Por último, la vista “usuario” en este apartado del portal del administrador es donde se crean a los usuarios y se les proporciona el rol correspondiente a cada usuario.

Nombre:	Fernando	Apellidos:	Armenteros Peña
DNI:	77521422R	Fecha de nacimiento:	26/01/2002
Teléfono:	670222111	Email:	Fernando@gmail.com
Contraseña:	****	Rol:	Usuario Administrador Especialista Usuario Autorizado
			<input type="button" value="Registrar"/>

Dni	Nombre	Apellidos	Teléfono	Email	Rol	Fecha_Nacimiento	
66315920G	Lourdes	Benitez Sanz	963333258	Lourdes@gmail.com	Especialista	26-05-1995	<input type="button" value="Borrar"/>
77521422R	Fernando	Armenteros Peña	670222111	Fernando@gmail.com	Usuario	26-01-2002	<input type="button" value="Borrar"/>

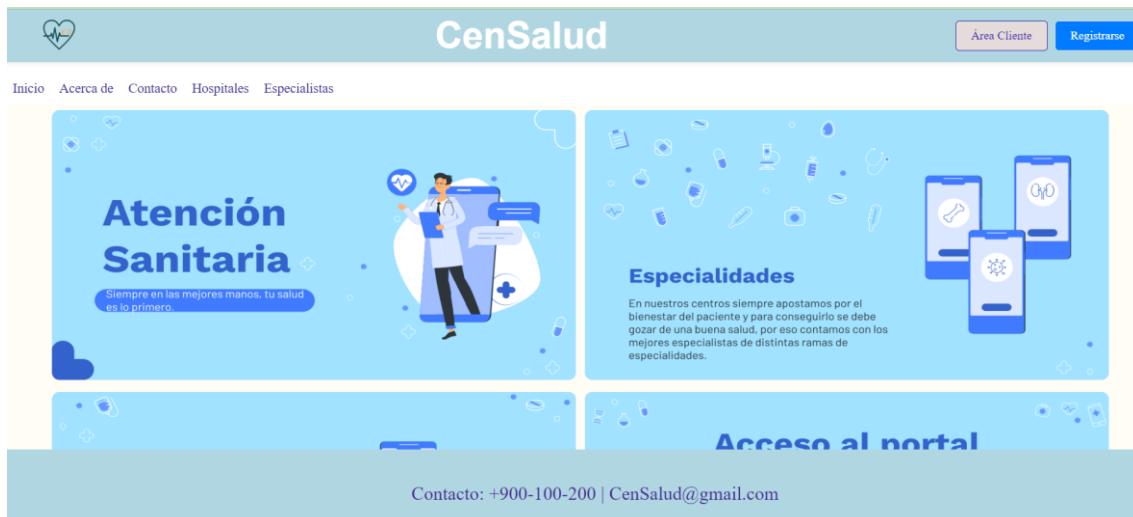
[Anterior](#) [1](#) ... [12](#) [13](#) **14**

Para eliminar un registro simplemente pulsar el botón “Borrar”, hay que tener en cuenta que algunas vistas están relacionadas con otras por lo que primero se ha de borrar la relación con esa vista y después el registro.

Modificar los registros, haciendo click encima del registro que se quiere modificar, mostrará los campos en el formulario y aparecerá un botón “Modificar” al pulsarlo directamente se modifica con los nuevos datos introducidos.

Página de inicio

Para ver el contenido de la página de inicio, el usuario no necesita estar logado ni estar registrado.



- El ícono lleva a la página de inicio en todas las páginas de la aplicación.
- Acerca de mostrará una página hablando de nuestros centros.

Nuestras raíces

El centro de salud CenSalud se fundó en el año 2000 con el objetivo de brindar atención médica de calidad y accesible para todos los habitantes de la comunidad. Desde sus inicios, el centro de salud contó con un equipo de médicos altamente capacitados y especializados en diferentes áreas, como dermatología, ginecología, pediatría y muchas más. Con el tiempo, el centro de salud se convirtió en uno de los lugares más populares para recibir atención médica en la ciudad, debido a su enfoque en la atención médica centrada en el paciente y en la contratación de médicos altamente capacitados y especializados en diferentes áreas. Los pacientes confiaban en el centro de salud para brindarles la atención médica que necesitaban, y siempre se sentían en buenas manos.

- Contacto, según la ciudad introducida mostrará los datos para ponerse en contacto con algún centro específico que se deseé.

Selección Ciudad

Contacto: +900-100-200 | CenSalud@gmail.com

Ciudad Buscar

- Hospitales, muestra todos los centros que hay.

Salud Esperanza Centro de salud Esperanza en C/Lope de Vega, 63 Granada Teléfono: 953112407	Salud Real Centro de salud Real en C/Real, 42 Almería Teléfono: 952774411	Salud Vida Centro de salud Vida en C/Principal, 70 Cádiz Teléfono: 975556314	Salud Mar adentro Centro de salud Mar adentro en C/Nueva, 17 Torremolinos Málaga Teléfono: 952223344
--	--	---	--

- Especialistas, muestra todos los especialistas, destacar que en todas las páginas se muestra un ícono para volver a la página anterior.

Urología Isabel Montes Alpes Da servicio en el centro de Salud Mar adentro en C/Nueva, 17 Torremolinos	Oftalmología Eustaquio Sanchez Ramírez Da servicio en el centro de Salud en C/Triavala, 20 Torremolinos	Ginecología María Trinidad Los Santos Vega Da servicio en el centro de Salud en C/Triavala, 20 Torremolinos	Dentista Francisco Gutierrez Sanchez Da servicio en el centro de bienestar en C/Federico, 17 Marbella
---	--	--	--

- Área cliente, te lleva al login del usuario.

- Registrarse lleva al formulario para crear un nuevo usuario, por defecto el usuario que se registre desde esta página tendrá el rol de “usuario”.

Área cliente

- Barra de navegación, muestra las páginas indicadas anteriormente en la página principal.
- Mi perfil, muestra un formulario con los datos del usuario.

CenSalud

Inicio Hospitales Especialistas Portal del usuario Mi perfil

FOTO: Cambiar foto

NOMBRE: ana

APELLIDOS: ana

FECHA DE NACIMIENTO: 01-01-1970

TELÉFONO: 999999999

EMAIL: lala@la.com

DNI: 55622350B

Restablecer datos

Restablecer contraseña

Al hacer click en restablecer datos, estos se pueden modificar.

CenSalud

Nombre: ana
Apellido: Aguilar Lara
DNI: 952125587
Fecha de nacimiento: 01-01-1970
Teléfono: 55622350B
Email: lala@la.com

Restablecer contraseña

Y los datos quedarán modificados en la base de datos.

CenSalud

Nombre: ana
Apellido: Aguilar Lara
DNI: 952125587
Fecha de nacimiento: 01-01-1970
Teléfono: 55622350B
Email: lala@la.com

Para restablecer contraseña, se realiza el mismo proceso, un click en restablecer contraseña y muestra el formulario de modificación.

CenSalud

Inicio Hospitales Especialistas Portal del usuario Mi perfil

FOTO: Cambiar foto

NOMBRE: Ana

APELLIDOS: Aguilar Lara

FECHA DE NACIMIENTO: 01-01-1970

TELÉFONO: 952125587

Restablecer datos

Restablecer contraseña

Contraseña actual:

Nueva contraseña:

Verificar nueva contraseña:

Enviar

El portal da la bienvenida al usuario para hacerlo más personal.



- Pedir Cita: Seleccionar día y hora para continuar.

Seleccione los datos para elegir su cita:

Paso 1: Selecciona una fecha

dd/mm/aaaa --:--

Selecciona una fecha disponible.

Seleccione los datos para elegir su cita:

Paso 1: Selecciona una fecha

26/05/2023 12:58

mayo de 2023 ▾							↑	↓	08	55
L	M	X	J	V	S	D			09	56
1	2	3	4	5	6	7			10	57
8	9	10	11	12	13	14			11	58
15	16	17	18	19	20	21			12	59
22	23	24	25	26	27	28			13	00
29	30	31	1	2	3	4			14	
5	6	7	8	9	10	11				
<input type="button" value="Borrar"/> Hoy										

En el paso 2, seleccionar la especialidad y continuar.

Seleccione los datos para elegir su cita:

Paso 2: Selecciona una especialidad hospitalaria

Portal del usuario Pedir Cita

Selec... u cito:

Podología	
Dermatología	
Oftalmología	
Dentista	
Ginecología	
Urología	
Traumatología	
Médico de familiar	
Oftalmología	
Areología	
Oncología	
Dietista	
Psiquiatría	
Neurología	
Cardiología	
Hematología	
Psicología	
Psicología	<input type="button" value="▼"/>
<input type="button" value="Anterior"/> <input type="button" value="Siguiente"/>	

En el paso 3, seleccionar localidad y continuar.

Seleccione los datos para elegir su cita:

Paso 3: Selecciona una localidad

Almansa ▾

Anterior Siguiente

Seleccione los datos para elegir su cita:

Paso 3: Selecciona una localidad

Almansa ▾

- Selecciona una localidad...
- Torremolinos
- Martos
- Luque
- Marbella
- Córdoba
- Huelva
- Sevilla
- Cádiz
- Almería
- Granada
- Almansa

En el paso 4, seleccionar centro de la localidad y continuar.

Seleccione los datos para elegir su cita:

Paso 4: Selecciona un centro

Salud Almansa ▾

Anterior Siguiente

Seleccione los datos para elegir su cita:

Paso 4: Selecciona un centro

Salud Almansa ▾

- Selecciona un centro...
- Mar adentro
- Salud
- Bienestar
- Esperanza
- Salud Real
- Salud Vida
- Salud Oliva
- Salud bienestar
- Salud Al-Andalus
- Salud Sevilla
- Salud Almansa

En el paso 5, elegir el tipo de cita y confirmar.

Seleccione los datos para elegir su cita:

Paso 5: Selecciona el tipo de cita

Seguimiento ▾

Anterior **Confirmar**

Seleccione los datos para elegir su cita:

Paso 5: Selecciona el tipo de cita

Seguimiento ▾

Selecciona un tipo de cita...
Primera consulta
Revisión
Rutinaria
Seguimiento

Pulsar el botón “Mis Citas”.

Se ha confirmado su cita:

Cita Confirmada:

Mis Citas

Esto te llevará a la pestaña Mis citas del área de cliente.

- Mis citas: Muestra las citas programadas.

Próxima cita
Tu proxima cita es Rutinaria con el especialista Isabel Montes Alpes el 2023-05-22 00:00:00 **Borrar**

Próxima cita
Tu proxima cita es Seguimiento con el especialista Lourdes Benitez el 2023-05-26 00:00:00 **Borrar**

- Historial: Muestra los informes clínicos del paciente.

Rutinaria Isabel 2023-05-22 00:00:00

Seguimiento Lourdes 2023-05-26 00:00:00

- Salud: Muestra consejos de salud.

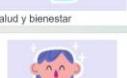


CenSalud





Salud y bienestar



Investigación



Prevención

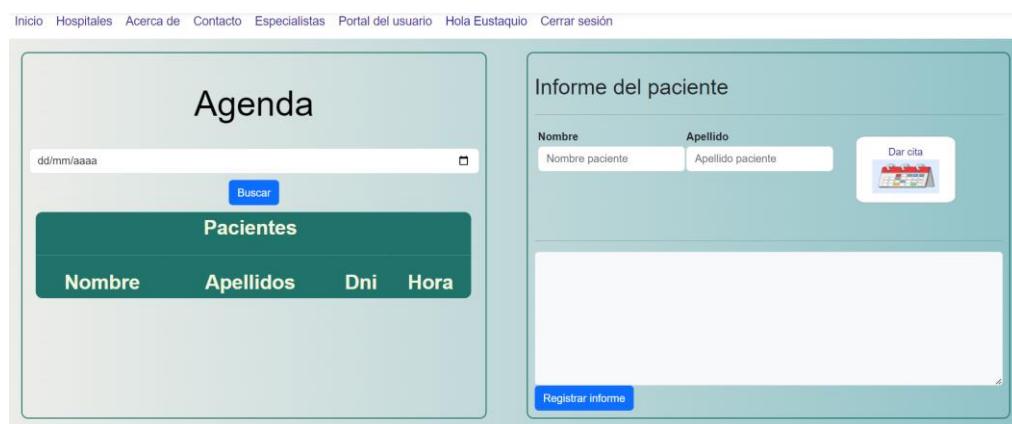
Salud y bienestar

Gozar de una buena salud es de suma importancia, ya que mejora nuestra calidad de vida y poder vivir una vida plena, sea cual sea la edad en la que nos encontramos.

Llevar una vida saludable no es tan fácil ya que por costumbres, trabajo, ir deprisa, etc. Nuestros hábitos son poco saludables y aun siendo consciente de ello nos lleva a tener problemas de salud en un futuro. Una motivación para cambiar nuestros hábitos saludables es ver los beneficios y ser conscientes de lo que nos pueden aportar en nuestro estado físico, mental y afectivo.

- Previene problemas de sobrepeso, nos ayuda a perder peso.
- Aumenta los niveles de energía.
- Mejora la acción respiratoria.
- Ayuda a tener un desarrollo y crecimiento más saludable.
- Aumenta el rendimiento en general: fuerza, velocidad, resistencia.
- Hay una gran mejoría de la coordinación.

Portal Especialista



En la parte Agenda, seleccionar fecha y buscar.

Pacientes				
Nombre	Apellidos	Dni	Hora	
Ana	López Suarez	88756952N	2023-05-31 03:00:00	
Ana	Martínez Guerra	77777777L	2023-05-31 11:20:00	
Carmen	López Bravo	77455212L	2023-05-31 17:12:00	
Matiasss	Benítez Pérez	88546963M	2023-05-31 18:19:00	

Al hacer click en cualquier registro los datos se muestran en la parte Informe del paciente.

The screenshot shows two side-by-side panels. The left panel is titled 'Agenda' and contains a search bar ('dd/mm/aaaa') and a 'Buscar' button. Below this is a table titled 'Pacientes' with columns: Nombre, Apellidos, Dni, and Hora. It lists two entries: Ana López Suarez (DNI 88756952N, Hora 03:00:00) and Ana Martínez Gómez (DNI 77777777L, Hora 11:20:00). The right panel is titled 'Informe del paciente' and shows a form with 'Nombre' (Ana) and 'Apellido' (López Suarez). A 'Dar cita' button is visible next to the patient information. At the bottom of the right panel is a 'Registrar informe' button.

Escribir informe y registrar informe.

This screenshot shows a large central area labeled 'Informe del paciente ...' which is currently empty. At the bottom of this area is a blue 'Registrar informe' button.

Si el especialista necesitara dar una cita, pulsar en la tarjeta “Dar cita”, esta le llevará a la página pedir cita, pero con un campo adicional para introducir el DNI del paciente y poder darle cita a través del DNI, el proceso es igual al explicado en el área de usuario.

Portal Usuario autorizado

The screenshot shows the main menu of the CenSalud application for authorized users. The top navigation bar includes links for Inicio, Hospitales, Acerca de, Contacto, Especialistas, Portal del usuario, Hola Amelia, and Cerrar sesión. Below the navigation are five main menu items: 'Paciente' (with a person at a computer icon), 'Especialistas' (with a doctor icon), 'centros' (with a map of Spain icon), 'Agenda' (with a calendar icon), and 'Nuevo usuario' (with a person on a smartphone icon). A 'Dar cita' button with a calendar icon is also present.

- Paciente: Mediante el DNI del paciente el personal autorizado puede ver sus datos.

Paciente

Información

Paciente

Información

Ana López Suarez

Cita	Informe Clínico
2023-05-23 04:55:00	La paciente presenta dolores de cabeza

- Especialista: Haciendo click en especialista, introduciendo el nombre del centro y buscar, se mostrarán todos los especialistas de ese centro.

Especialistas

Buscar

Especialistas

Buscar

Especialista	Especialidad	Centro
Isabel	Urología	Bienestar
Francisco	Urología	Bienestar

- Centros: Mediante el nombre de la provincia se mostrará todos los centros que se encuentran en ella.

Centros

Provincia

Buscar

Centros

Ciudad	Dirección	Teléfono
Torremolinos	C/nueva,17	952223344
Torremolinos	C/Traviata,20	952336644
Marbella	C/Federico,17	954987741

- Agenda: Mediante el nombre, apellido y fecha mostrará la agenda del especialista seleccionado.

Especialista

Nombre

Apellido

dd/mm/aaaa

Buscar

Especialista

Paciente	Fecha
Ana Martínez Guerra	2023-05-31 16:33:00

- Nuevo registro: redirige a la página crear un nuevo usuario, comentada en la descripción de la página principal.
- Dar Cita: Sigue el mismo proceso mencionado en el portal del Especialista.

22. Conclusiones globales

Con la realización de este proyecto he comprendido la unión de la parte cliente con la parte servidor, además de cómo hacer uso de las bases de datos.

Ha sido una experiencia muy buena ya que he practicado bastante tanto con HTML, CSS y JavaScript como con PHP, aunque le he puesto más horas de las necesarias creo que ha sido un buen proyecto. Todo el desarrollo desde cero hasta su fin ha sido creación propia y me ha permitido dejar volar mi imaginación.

He tenido momentos en que no sabía cómo solucionar alguna implementación, pero buscando información e indagando he conseguido solucionar el problema, eso me ha servido para afianzar el aprendizaje, como por ejemplo la realización de un patrón del CRUD, esto fue una idea que se me ocurrió pero no sabía si se podía hacer ni como se hacía, con esto he conseguido hacer un buen uso de las bases de datos, aunque me queda mucho por aprender ya que para algunas funciones he realizado las consultas sin patrón, por lo que considero que ha sido un buen principio.

Debido al corto tiempo creo que la mayoría de las funcionalidades que me había propuesto las he realizado, me hubiera gustado hacer más funcionalidades como son:

- Crear el portal del familiar adjunto al usuario.
- Huecos para citas en agendas por especialista.
- Cuando el usuario busque en todos los especialistas, que con un click le lleve a pedir cita con ese especialista en concreto.
- Al igual que con el centro cuando busque un centro al click le salga las especialidades que contiene el centro y pida cita directamente.
- Que el informe del paciente se guarde en la base de datos mediante un PDF y el usuario pueda descargarlo directamente.
- Añadir un botón para que el usuario pueda cambiar su imagen.
- Que el usuario pueda realizar una modificación del email.
- Si el usuario ha olvidado su contraseña, en el login un enlace “Has olvidado tu contraseña” y mediante un correo pueda actualizar la contraseña.
- El botón búsqueda rápida para el índice.

Aunque solo he puesto estas funcionalidades como mejoras futuras siempre se pueden seguir realizando más.

Me hubiera gustado realizar pruebas de otros usuario tester, ya que con estas pruebas se puede obtener las sensaciones que causa en otros usuarios, además de proporcionar mejoras y encontrar errores, en este caso las únicas pruebas que se han realizado han sido las que yo he realizado para ver su correcto funcionamiento.

Por último, aclarar que los datos son todos ficticios, si hay alguna coincidencia es pura casualidad.

23. Webgrafía

<https://stackoverflow.com/questions/19572763/echo-value-into-hidden-field>

<https://stackoverflow.com/questions/195951/how-can-i-change-an-elements-class-with-javascript>

<https://www.php.net/manual/es/index.php>

<https://www.neoguias.com/tutorial-crud-php/>

https://www.youtube.com/watch?v=xIPnc0iUick&ab_channel=Socket%27sService

https://www.youtube.com/watch?v=j7bml8EQpIk&ab_channel=Develoteca

<https://www.baulphp.com/unir-dos-tablas-y-buscar-con-php-mysql/>

<https://www.forosdelweb.com/f18/unir-datos-dos-tablas-php-mysql-1149490/>

<https://www.baulphp.com/paginacion-usando-php-mysqli-bootstrap/>

https://www.youtube.com/watch?v=hgS9vX7REbw&ab_channel=CodersFree

login

https://www.youtube.com/watch?v=LL66QTP3txE&ab_channel=VidaMRR-Programacionweb

https://www.youtube.com/watch?v=Tb-_cEAsp4s&ab_channel=VidaMRR-Programacionweb

https://www.youtube.com/watch?v=AWjCxj9D_Ko&ab_channel=GABRIELHDS

Reemplazar div

https://www-tutorialspoint-com.translate.goog/replace-a-child-node-with-a-new-node-in-javascript?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq#:~:text=To%20replace%20a%20child%20node,are%20features%20of%20DOM%20Level1.

<https://www.pildorasinformaticas.es/course/javascript-desde-0/>

php

<https://www.pildorasinformaticas.es/course/php-mysql/>

imágenes

<https://slidesgo.com/es/centro-medico?order=recent#rs=collections-medical-collection>

<https://icons.getbootstrap.com/?q=magnifying+glass>

<https://express.adobe.com/es-ES/sp/>

<https://labs.openai.com/e/4lp5U8yZlcqyUMMeCt1zG565>

<https://imageresizer.com/es>

aplicación responsive

https://www.youtube.com/watch?v=ZiwEpJshXQc&ab_channel=Formaci%C3%B3nGr%C3%A1fica1fica

https://developer.mozilla.org/es/docs/Learn/Common_questions/Design_and_accessibility/Common_web_layouts

https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries/Using_media_queries

<https://crocreativo.com/blog/diseno-web-responsive/>

GIT <https://github.com/anaicm/Hospital>