

# Assista

---

APLICACIÓN WEB PARA EL BUFFETE DE ABOGADOS  
“SONIA PARRAS LAW OFFICE”

Nicolás Figueras Parras

10/06/2019 | 2º DESARROLLO DE APLICACIONES WEB

## Índice

Descripción general del proyecto .....	2
Análisis de las áreas .....	4
Requisitos de las áreas de gestión .....	5
Modelo Entidad / Relación .....	8
Paso a tablas y normalización.....	9
Implantación de la base de datos .....	13
Creación.....	13
Código SQL de la inserción de datos .....	18
Diseño web .....	20
Guía de colores .....	24
Programación de la web.....	25
Admin .....	27
Client .....	29
Worker .....	30
Lawer.....	32
Casos de uso .....	34
Manual de usuario .....	35
Trabajador .....	35
Cliente .....	41

## Descripción general del proyecto

En este proyecto se va a tratar la construcción de una aplicación web sobre una oficina de abogados. En esta oficina trabajan una serie de abogados del área de inmigración. Cuando un cliente solicita los servicios de esta oficina se le asigna un abogado que llevará su caso. Esta oficina necesita de una aplicación que le ayude a gestionar los datos de los propios abogados, los datos de los clientes y sus casos, la gestión de pagos y tarifas por los servicios, además de los datos y tareas administrativas de los distintos trabajadores. Es importante tener en cuenta que la oficina se encuentra en Chicago (EEUU)

Esta aplicación, deberá gestionar 4 tipos de usuarios: cliente, oficinista, abogado y administrador.

Cuando accede a la aplicación un usuario no registrado sólo verá información sobre el trabajo que hace esta oficina.

En caso de que quiera contratar algún servicio, deberá acudir físicamente a la oficina para que un abogado le informe del proceso a seguir. Si el cliente está de acuerdo con todo lo que le planteen en la oficina, un oficinista le dará de alta para gestionar todos los aspectos de su caso.

En este momento, será cuando se le proporcionará al cliente el nombre de usuario y la contraseña de acceso personalizado a la aplicación.

A partir de ese momento, cuando el cliente acceda a la aplicación lo primero que verá será una página de inicio donde tendrá un menú lateral con distintas opciones.

Desde su perfil, el usuario podrá modificar sus datos, así como su usuario y contraseña.

Cuando accede a la aplicación un oficinista, lo primero que verá será la lista de tareas pendientes para ese día. Hay que tener en cuenta que la lista de tareas pendientes es común para todos los oficinistas. Si quiere ver la lista de tareas finalizadas o crear una, lo hará desde la sección de tareas.

Los estados que puede tener una tarea son:

- Pendiente
- Realizada

Además de esto, el oficinista es el encargado de las siguientes tareas:

1. Dar de alta a los clientes
2. Modificar los datos de los clientes
3. Gestionar la unión de los clientes con los abogados
4. Gestionar los estados de los casos

Cuando accede a la aplicación un abogado, verá un mensaje de bienvenida y un menú lateral desde el que podrá gestionar los clientes que tiene asignados (crear, listar, modificar, asignarles un caso y añadir un cobro. También podrá ver los casos que lleva, así como crearlos. En último lugar, podrá ver las tareas como lo hace un trabajador, al igual que marcarlas como finalizadas.

Según todo esto, mi aplicación se va a basar en las siguientes áreas:

1. Área de gestión de clientes.
2. Área de gestión de tareas.
3. Área de gestión de trabajadores
4. Área de gestión de casos.
5. Área de gestión de abogados.

## Análisis de las áreas

1. Área de gestión de clientes
  - a. Finalidad: insertar clientes, listar clientes, modificar los datos de clientes, dar de baja a clientes, consultar factura de clientes, consultar caso.
  - b. Elementos: clientes, casos.
2. Área de gestión de tareas
  - a. Finalidad: insertar tareas, listar tareas, modificar tareas, eliminar tareas, asignar tareas.
  - b. Elementos: tareas.
3. Área de gestión de casos
  - a. Finalidad: crear casos, modificar casos, borrar casos, listar casos, asignar casos.
  - b. Elementos: casos.
4. Área de gestión de abogados
  - a. Finalidad: crear abogados, asignar casos a abogados, asignar clientes a abogados, borrar abogados, editar abogados, listar casos, editar casos.
  - b. Elementos: abogados, casos, clientes.
5. Área de gestión de trabajadores
  - a. Finalidad: crear trabajadores, asignar tareas a trabajadores, realizar cuenta del dinero pagado y a deber del cliente, editar trabajadores, asignar horario a trabajadores, modificar casos.
  - b. Elementos: trabajadores, casos, clientes.

## Requisitos de las áreas de gestión

1. Área de gestión de clientes
  - a. Requisitos de datos
    - I. ID\_cliente
    - II. Nombre
    - III. Apellido
    - IV. Fecha de nacimiento
    - V. Teléfono de contacto
    - VI. Dirección de email
    - VII. Nombre de usuario
    - VIII. Contraseña
    - IX. Cuenta
  - b. Requisitos funcionales
    - I. Insertar cliente
    - II. Modificar datos de cliente
    - III. Listar clientes
    - IV. Dar de baja a clientes
    - V. Consultar cuenta de clientes
    - VI. Asignar caso a cliente
  - c. Requisitos semánticos
    - I. No puede haber dos clientes con el mismo nombre y apellido
    - II. El teléfono de contacto no puede existir previamente
    - III. El nombre de usuario debe ser único para cada cliente
    - IV. La cuenta empieza a 0
2. Área de gestión de trabajadores
  - a. Requisitos de datos
    - I. ID\_trabajador
    - II. Nombre
    - III. Apellido
    - IV. Fecha de nacimiento
    - V. Teléfono de contacto
    - VI. Dirección de email
    - VII. Nombre de usuario
    - VIII. Contraseña
    - IX. Sueldo
  - b. Requisitos funcionales
    - I. Crear trabajadores
    - II. Asignar tareas a trabajadores
    - III. Realizar cuenta del dinero pagado y a deber del cliente
    - IV. Editar trabajadores
    - V. Modificar casos.

- c. Requisitos semánticos
  - I. Todos los trabajadores deben ser mayores de edad
  - II. No puede haber dos trabajadores con el mismo nombre y apellido
  - III. El nombre de usuario debe ser único para cada trabajador
  - IV. El teléfono de contacto no puede existir previamente
- 3. Área de gestión de tareas
  - a. Requisitos de datos
    - I. ID\_tarea
    - II. Título
    - III. Descripción
    - IV. Fecha en la que se manda
    - V. Fecha de finalización
    - VI. Trabajador asociado
    - VII. Abogado que la manda realizar
    - VIII. Estado
  - b. Requisitos funcionales
    - I. Insertar tareas
    - II. Listar tareas
    - III. Modificar tareas
    - IV. Eliminar tareas
    - V. Asignar tareas
  - c. Requisitos semánticos
    - I. Cada tarea tiene asociado un único trabajador y un único abogado
- 4. Área de gestión de casos
  - a. Requisitos de datos
    - I. ID\_casos
    - II. Título
    - III. Descripción
    - IV. ID\_abogado
    - V. ID\_cliente
    - VI. Tipo de caso
  - b. Requisitos funcionales
    - I. Crear casos
    - II. Modificar casos
    - III. Borrar casos
    - IV. Listar casos
    - V. Asignar casos
  - c. Requisitos semánticos
    - I. No puede haber un mismo caso para dos clientes, ni un mismo caso para dos abogados

## 5. Área de gestión de abogados

## a. Requisitos de datos

- I. ID\_abogado
- II. Nombre
- III. Apellido
- IV. Fecha de nacimiento
- V. Teléfono de contacto
- VI. Dirección de email
- VII. Nombre de usuario
- VIII. Contraseña
- IX. Sueldo

## b. Requisitos funcionales

- I. Crear abogados
- II. Asignar casos a abogados
- III. Borrar abogados
- IV. Editar abogados
- V. Listar casos
- VI. Editar casos

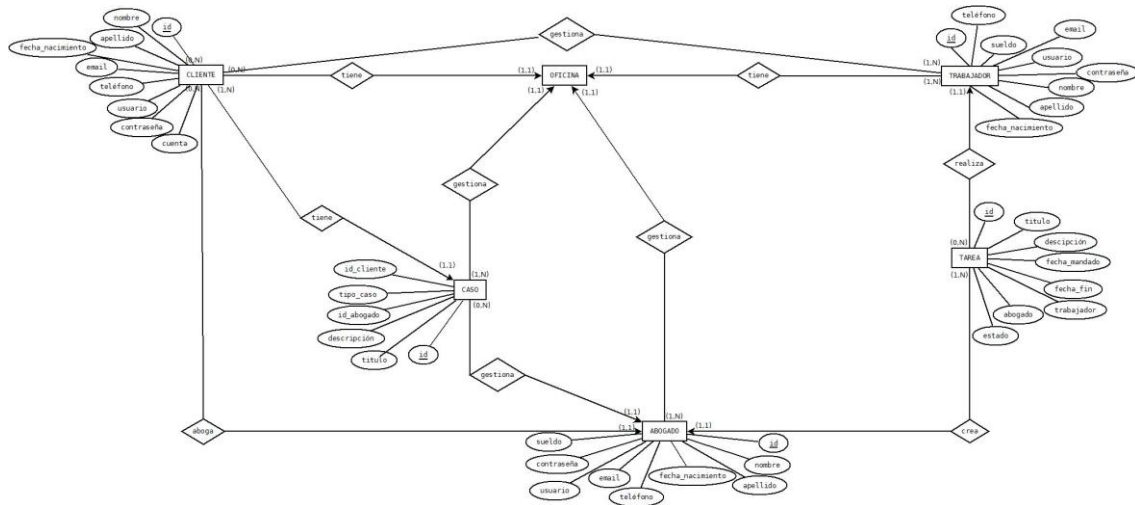
## c. Requisitos semánticos

- I. No puede haber dos abogados que se tengan el mismo nombre y apellido.



## Modelo Entidad / Relación

El modelo entidad relación de la base de datos es el siguiente:



Las principales entidades son los distintos usuarios de la aplicación: cliente, abogado y trabajador. El resto es lo que les une:

- Los acuden a la oficina para gestionar un **caso** judicial.
- El **trabajador** les da de alta y les pone en contacto con un **abogado**.
- A partir de esto, un **trabajador** adquiere nuevas **tareas**.

## Paso a tablas y normalización

Tablas que surgen de las entidades:

- CLIENTE (id\_cliente, nombre, apellido, fecha\_nacimiento, teléfono, email, usuario, contraseña, cuenta)
- CASO (id\_caso, título, descripción, id\_abogado, id\_cliente, tipo)
- ABOGADO (id\_abogado, nombre, apellido, fecha\_nacimiento, teléfono, email, usuario, contraseña, sueldo)
- TAREA (id\_tarea, título, descripción, fecha\_mandado, fecha\_fin, id\_abogado, id\_trabajador)
- TRABAJADOR (id\_trabajador, nombre, apellido, fecha\_nacimiento, teléfono, email, usuario, contraseña, salario)

Tablas que surgen de las relaciones:

TABLA	CARDINALIDAD	RESULTADO
Tiene ( <u>id_cliente</u> , <u>id_caso</u> )	N:1	Se mantiene
Crea ( <u>id_abogado</u> , <u>id_tarea</u> )	1:N	Desaparece ABOGADO( <u>id_abogado</u> )====>TAREA
Gestiona ( <u>id_caso</u> , <u>id_abogado</u> )	N:1	Se mantiene
Aboga ( <u>id_cliente</u> , <u>id_abogado</u> )	N:1	Se mantiene
Gestiona ( <u>id_caso</u> , <u>id_trabajador</u> )	N:N	Se mantiene
Realiza ( <u>id_tarea</u> , <u>id_trabajador</u> )	1:N	Desaparece TAREA( <u>id_tarea</u> )====>TRABAJADOR

Conjunto final de tablas:

- CLIENTE (id\_cliente, nombre, apellido, fecha\_nacimiento, email, teléfono, cuenta)
- CASO (id\_caso, id\_cliente, tipo\_caso, id\_abogado, descripción, título)
- ABOGADO (id\_abogado, nombre, apellido, fecha\_nacimiento, teléfono, email, sueldo)
- TAREA (id\_tarea, título, descripción, fecha\_mandado, fecha\_fin, id\_abogado, id\_trabajador)
- TRABAJADOR (id\_trabajador, teléfono, sueldo, email, nombre, apellido, fecha\_nacimiento)

## CLIENTE

### 1FN

Una tabla está en primera forma normal si todos sus atributos no primos dependen funcionalmente de la clave principal.

Esta tabla ya está en 1FN porque todos sus atributos no primos dependen funcionalmente de la clave principal que es el DNI

### 2FN

Una tabla está en segunda forma normal si, está en 1FN y, todos los atributos no primos tienen una dependencia funcional completa de la clave principal.

Esta tabla ya está en 2FN porque está en 1FN y además la clave principal es simple.

### 3FN

Una tabla está en tercera forma normal si, está en 2FN y, no existen dependencias funcionales transitivas.

Esta tabla ya está en 3FN porque está en 2FN y además no hay dependencias funcionales transitivas

## CASO

### 1FN

Una tabla está en primera forma normal si todos sus atributos no primos dependen funcionalmente de la clave principal.

Esta tabla ya está en 1FN porque todos sus atributos no primos dependen funcionalmente de la clave principal que es el ID

### 2FN

Una tabla está en segunda forma normal si, está en 1FN y, todos los atributos no primos tienen una dependencia funcional completa de la clave principal.

Esta tabla ya está en 2FN porque está en 1FN y además la clave principal es simple.

### 3FN

Una tabla está en tercera forma normal si, está en 2FN y, no existen dependencias funcionales transitivas.

Esta tabla ya está en 3FN porque está en 2FN y además no hay dependencias funcionales transitivas

## ABOGADO

### 1FN

Una tabla está en primera forma normal si todos sus atributos no primos dependen funcionalmente de la clave principal.

Esta tabla ya está en 1FN porque todos sus atributos no primos dependen funcionalmente de la clave principal que es el ID

### 2FN

Una tabla está en segunda forma normal si, está en 1FN y, todos los atributos no primos tienen una dependencia funcional completa de la clave principal.

Esta tabla ya está en 2FN porque está en 1FN y además la clave principal es simple.

### 3FN

Una tabla está en tercera forma normal si, está en 2FN y, no existen dependencias funcionales transitivas.

Esta tabla ya está en 3FN porque está en 2FN y además no hay dependencias funcionales transitivas

## TAREA

### 1FN

Una tabla está en primera forma normal si todos sus atributos no primos dependen funcionalmente de la clave principal.

Esta tabla ya está en 1FN porque todos sus atributos no primos dependen funcionalmente de la clave principal que es el ID

### 2FN

Una tabla está en segunda forma normal si, está en 1FN y, todos los atributos no primos tienen una dependencia funcional completa de la clave principal.

Esta tabla ya está en 2FN porque está en 1FN y además la clave principal es simple.

### 3FN

Una tabla está en tercera forma normal si, está en 2FN y, no existen dependencias funcionales transitivas.

Esta tabla ya está en 3FN porque está en 2FN y además no hay dependencias funcionales transitivas

## TRABAJADOR

### 1FN

Una tabla está en primera forma normal si todos sus atributos no primos dependen funcionalmente de la clave principal.

Esta tabla ya está en 1FN porque todos sus atributos no primos dependen funcionalmente de la clave principal que es el ID

### 2FN

Una tabla está en segunda forma normal si, está en 1FN y, todos los atributos no primos tienen una dependencia funcional completa de la clave principal.

Esta tabla ya está en 2FN porque está en 1FN y además la clave principal es simple.

### 3FN

Una tabla está en tercera forma normal si, está en 2FN y, no existen dependencias funcionales transitivas.

Esta tabla ya está en 3FN porque está en 2FN y además no hay dependencias funcionales transitivas.

## Implantación de la base de datos

### CREACIÓN

CASES:

```
CREATE TABLE `cases` (  
  `case_ID` bigint(20) UNSIGNED NOT NULL,  
  `title` varchar(200) NOT NULL,  
  `description` varchar(1000) NOT NULL,  
  `lawer_ID` bigint(20) NOT NULL,  
  `client_ID` bigint(20) NOT NULL,  
  `type` varchar(50) NOT NULL  
)
```

La tabla “cases” almacena los datos de los casos que lleva la oficina. Todos los datos son obligatorios, por lo que no se permite ninguna columna nula. Cuenta con un ID autonumérico que asigna un número identificativo a cada caso, con un título de extensión máxima de 200 caracteres, junto con su descripción, que asciende a 1000 caracteres. Las columnas “lawer\_ID” y “client\_ID” hacen referencia a la clave primaria por la que podemos hacer referencia a un abogado y a un cliente. La última columna “type” almacenará el tipo de caso del que se trata.

## CLIENTS:

```
CREATE TABLE `clients` (  
  `client_ID` bigint(20) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `surname` varchar(70) NOT NULL,  
  `birth_date` date NOT NULL,  
  `phone` int(11) NOT NULL,  
  `email` varchar(70) NOT NULL,  
  `username` varchar(15) NOT NULL,  
  `password` varchar(25) NOT NULL,  
  `bill` float NOT NULL  
)
```

La tabla “clients” almacena la información de todos los clientes que están dados de alta en la oficina. Como la tabla anterior, cuenta con un ID autonumérico, además de ser obligatorio insertar todos los datos para crear un nuevo cliente, ya que no permite ningún dato nulo. El nombre y el apellido del cliente los almacena en las columnas “name” y “surname”. La fecha de nacimiento, en la columna de tipo dato “birth\_date”. Los datos de contacto los almacena en las columnas “phone” y “email”. También se almacena el usuario y contraseña con el que dichos clientes acceden a la aplicación con las columnas “username” y “password”. Por último, “bill” que hace referencia a la cantidad de dinero que debe a la oficina.

## LAWERS:

```
CREATE TABLE `lawers` (  
  `lawer_ID` bigint(20) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `surname` varchar(70) NOT NULL,  
  `birth_date` date NOT NULL,  
  `phone` int(11) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `username` varchar(15) NOT NULL,  
  `password` varchar(30) NOT NULL,  
  `salary` float NOT NULL  
)
```

Esta tabla tiene una estructura bastante similar a la que almacena la información de los clientes, solo que “lawers” lo almacena de los abogados. La principal diferencia está en que, en lugar de una columna llamada “bill”, se usa “salary”, que referencia al salario del abogado.



## WORKERS:

```
CREATE TABLE `workers` (  
  `worker_ID` bigint(20) NOT NULL,  
  `name` varchar(30) NOT NULL,  
  `surname` varchar(30) NOT NULL,  
  `birth_date` date NOT NULL,  
  `phone` int(11) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `username` varchar(15) NOT NULL,  
  `password` varchar(30) NOT NULL,  
  `salary` float NOT NULL  
)
```

Esta tabla almacena la información de los trabajadores. Tiene la misma estructura que la tabla de los abogados.

## TASKS:

```
CREATE TABLE `tasks` (  
  `task_ID` bigint(20) NOT NULL,  
  `title` varchar(200) NOT NULL,  
  `description` varchar(1000) NOT NULL,  
  `start_date` date NOT NULL,  
  `end_date` date NOT NULL,  
  `worker_ID` bigint(20) NOT NULL,  
  `lawer_ID` bigint(20) NOT NULL,  
  `status` int(1) NOT NULL  
)
```

“Tasks” almacena los datos de las tareas que se envían y realizan entre trabajadores y abogados. Como todas las anteriores, no permite ningún dato nulo. El título y la descripción se almacenan en las columnas denominadas “title” y “description”. La fecha de inicio y de finalización de la tarea se almacenan en “start\_date” y “end\_date”. “worker\_ID” y “lawer\_ID” hacen referencia al abogado y el trabajador responsables de la creación y finalización de una tarea. Para saber si una tarea está finalizada o no, se utiliza la columna “status” con un número entero, el cual posteriormente se interpreta desde el código de la aplicación.

## CÓDIGO SQL DE INSERCIÓN DE DATOS

```
INSERT INTO `cases` (`case_ID`, `title`, `description`, `lawer_ID`,
`client_ID`, `type`) VALUES
(1, 'Infracción por conducción temeraria', 'El cliente pide asesoría
legal para mantener su ciudadanía tras haber cometido una infracción de
tráfico', 1, 2, 'Tráfico'),
(2, 'Permiso laboral', 'El cliente desea obtener el permiso laboral en el
país', 2, 1, 'Permiso laboral'),
(3, 'Obtención de la nacionalidad', 'El cliente desea obtener la
nacionalidad por medio de un familiar que reside en el país', 1, 4,
'Nacionalidad');
```

```
INSERT INTO `clients` (`client_ID`, `name`, `surname`, `birth_date`,
`phone`, `email`, `username`, `password`, `bill`) VALUES
(1, 'Nicolás', 'Figueras Parras', '1999-03-22', 639941992,
'nicolasfiguerasparras@gmail.com', 'nicoClient', 'nicoClient', 850),
(2, 'Fernando', 'Juanez ', '1998-12-03', 626451200, 'fernando@gmail.com',
'fernando', 'fernando', 0),
(3, 'Jose Luis', 'Jimenez Marquez', '1980-05-06', 621545898,
'joseluis@gmail.com', 'joseluis', 'joseluis', 0),
(4, 'Luis', 'Vigo Pérez', '1996-07-02', 722154535, 'luisrubio@gmail.com',
'luis', 'rubio', 0),
(5, 'Patricia Elvira', 'Jimenez Martín', '2000-05-24', 645512466,
'mesosaurus_zen@gmail.com', 'patricia', 'mesosaurus', 0);
```

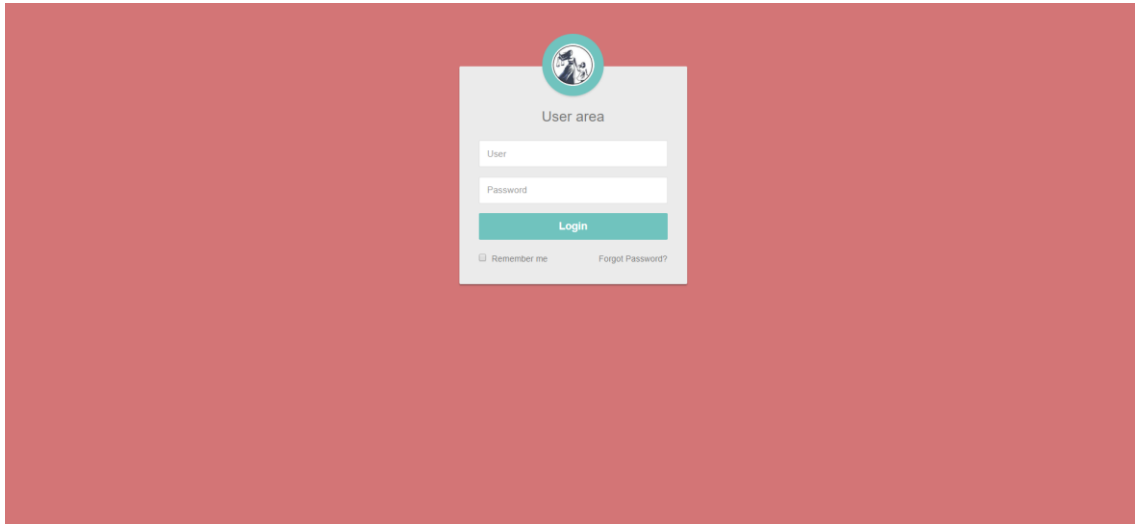
```
INSERT INTO `lawers` (`lawer_ID`, `name`, `surname`, `birth_date`,
`phone`, `email`, `username`, `password`, `salary`) VALUES
(1, 'Sonia', 'Parras', '1976-03-25', 639941992, 'soniaparras@law.com',
'soniaparras', 'lawoffice', 3500),
(2, 'Noemí', 'Casas', '1978-03-21', 649310891, 'noemi@law.com', 'noemi',
'parrasCasas', 1500),
(3, 'Mark', 'Konrad', '1984-08-15', 712245165, 'mark@law.com', 'mark',
'Konrad', 3400);
```

```
INSERT INTO `tasks` (`task_ID`, `title`, `description`, `start_date`,
`end_date`, `worker_ID`, `lawer_ID`, `status`) VALUES
(1, 'Llamada', 'Llamar a todos los clientes que tuvieron cita ayer',
'2019-06-02', '2019-06-11', 1, 1, 0),
(2, 'Recopilar datos', 'Pedir datos a los clientes para registrar en la
aplicación', '2019-01-01', '2019-06-10', 1, 3, 0),
(3, 'Recordatorio de cita', 'Llamar a Olivia Piñero Viñolo para
recordarle su juicio del viernes 14/06', '2019-06-10', '2019-06-11', 2,
2, 0),
(4, 'Organizar facturas', 'Ordenar todas las facturas para pasarlas a la
aplicación', '2019-06-01', '2019-06-20', 3, 2, 0);
```

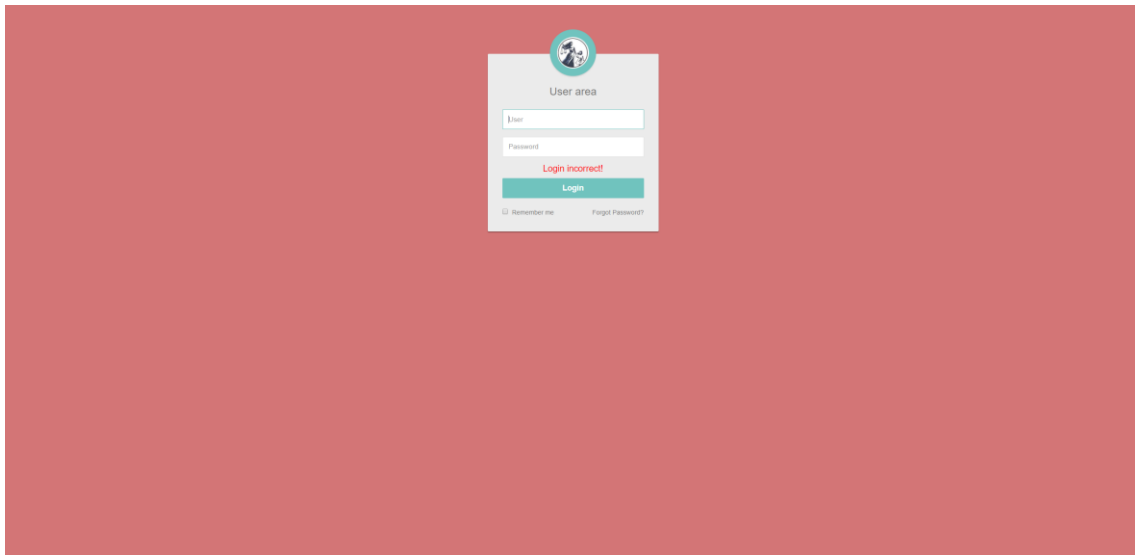
```
INSERT INTO `workers` (`worker_ID`, `name`, `surname`, `birth_date`,
`phone`, `email`, `username`, `password`, `salary`) VALUES
(1, 'Raúl', 'Méndez Núñez', '1990-03-22', 639941992, 'raul@worker.com',
'raulmendez', 'mendeznuñez', 800),
(2, 'Tania', 'Rodríguez Gutiérrez', '1980-03-06', 61542451,
'tania@worker.com', 'tania', 'rodriguez', 1200),
(3, 'Olivia', 'Piñero Viñolo', '1977-11-14', 615124351,
'olivia@worker.com', 'olivia', 'piñero', 2500),
(4, 'Alicia', 'Martín Vicente', '1988-09-14', 655255135,
'alicia@worker.com', 'alicia', 'martin', 1000);
```

## DISEÑO WEB

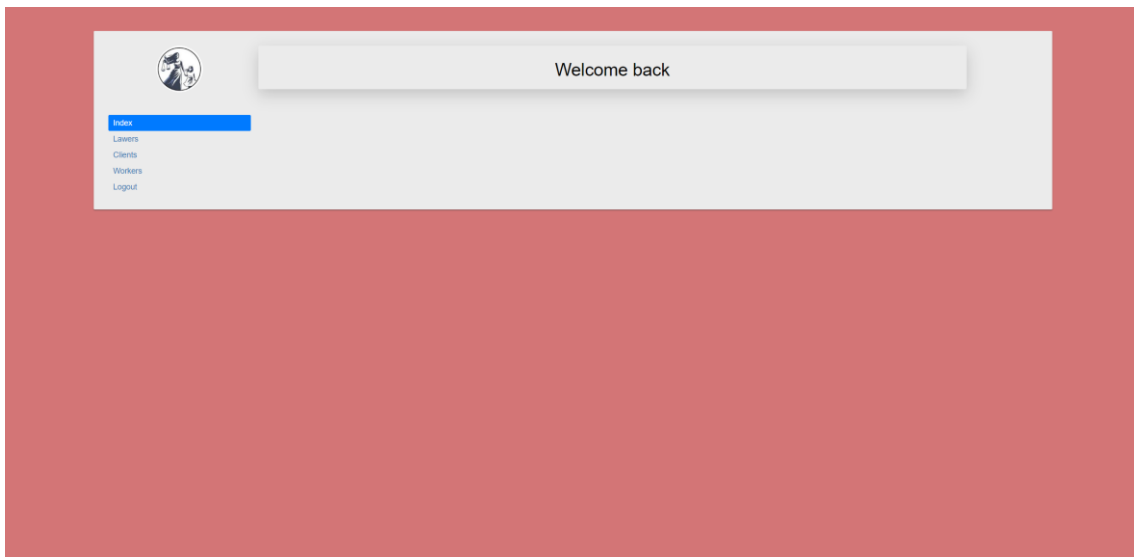
Al comenzar a diseñar la aplicación, busqué una plantilla de login usando la librería de Bootstrap y la modifiqué a mi gusto. A partir de ahí, comencé a desarrollar el resto tomando la misma como guía.



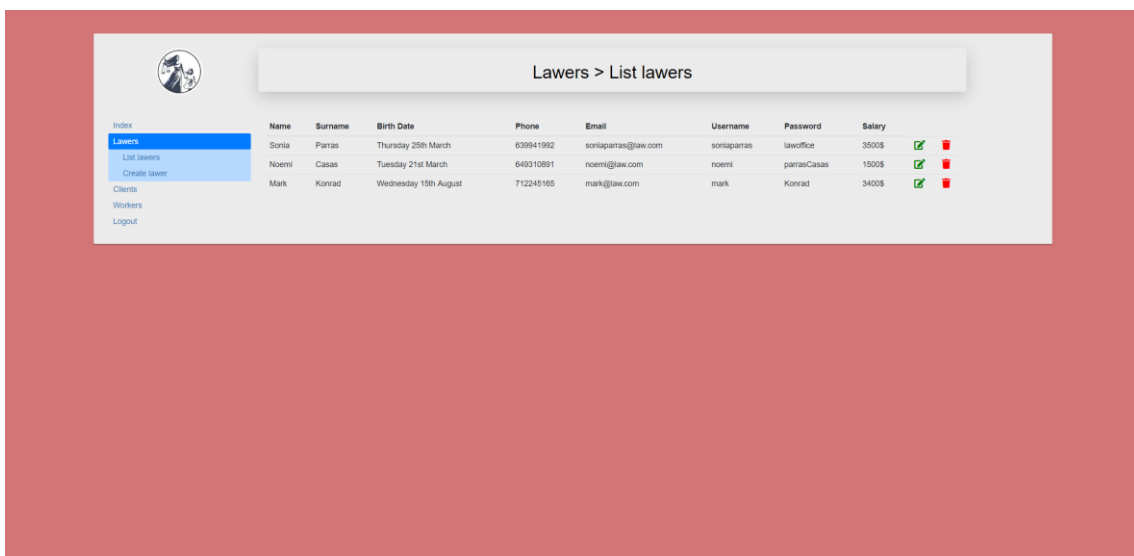
El diseño se basa en un fondo con el color #d47677 y, sobre él, una caja con un color de fondo #ececcec con sombra en los bordes laterales e inferior. En la parte superior de la misma se ubica el logo de la empresa, que aparecerá en el área personal de cada usuario y como imagen de pestaña. Cuando se pincha en un casillero del formulario, este cambia sus bordes de #e3e3e3 a #70c5c0.



En caso de introducir un usuario y/o contraseña incorrectos, nos aparecerá un mensaje de error en rojo.



En caso de introducir correctamente los datos, accederemos al área personal. En concreto, éste es el área personal del administrador. En la esquina superior izquierda de la caja interior tendremos el logotipo de la empresa. Justo debajo se encuentra el menú que nos permite navegar por todas las acciones que podemos realizar. El menú es de la clase “*nav flex-column nav-pills*”, de la librería de Bootstrap.



Al acceder a un apartado, se desplegará una lista de sub-acciones en un fondo de color #b8daff. Para conocer en qué apartado nos encontramos, se indicará en una caja con sombreado, ubicada en la parte superior del contenido. Las tablas son de clase “table”. Los iconos utilizados son de la librería de “fontawesome”.

Lawers > Modify lawyer

Index	First name	Last name	Salary	Birth date
Lawers	Sonia	Parras	3000	25/03/1975
List lawers	Phone number	Email	Username	Password
Create lawyer	639941992	soniaparras@lee.com	soniaparras	*****
Clients	Modify			
Workers				
Logout				

Los posibles formularios a los que accede el usuario usan el mismo estilo que el login.

```
.form-control {
    min-height: 41px;
    background: #fff;
    box-shadow: none !important;
    border-color: #e3e3e3;
}
.form-control:focus {
    border-color: #70c5c0;
}
.form-control, .btn {
    border-radius: 2px;
}
.login-form input[type="checkbox"] {
    margin-top: 2px;
}
.login-form .btn {
    font-size: 16px;
    font-weight: bold;
    background: #70c5c0;
    border: none;
    margin-bottom: 20px;
}
```

```
.login-form .btn:hover, .login-form .btn:focus {
    background: #50b8b3;
    outline: none !important;
}
.login-form a {
    color: #fff;
    text-decoration: underline;
}
.login-form a:hover {
    text-decoration: none;
}
.login-form form a {
    color: #7a7a7a;
    text-decoration: none;
}
.login-form form a:hover {
    text-decoration: underline;
}
```

Dado que la aplicación es completamente personalizada, no he encontrado una competencia que pudiese referenciar.

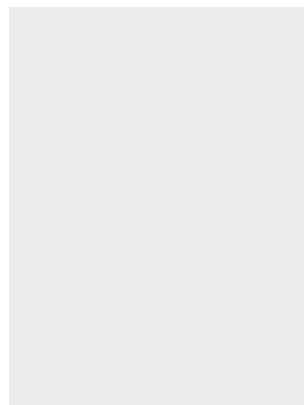


## GUÍA DE COLORES

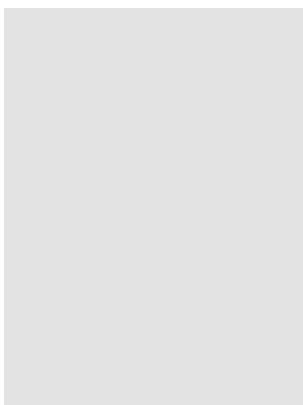
#d47677



#ececec



#e3e3e3



#70c5c0



## PROGRAMACIÓN DE LA WEB

Lo primero que se ha tenido que programar ha sido el sistema de acceso identificado a la aplicación. El código es el siguiente:

```
<!-- Form action -->
<?php
if(isset($_POST['submit'])){
    // Collect user and password on $_POST
    $user = $_POST['username'];
    $password = $_POST['password'];

    // Create query for all possible login
    $queryClient = mysqli_fetch_array(mysqli_query($db, "SELECT * FROM clients WHERE username = '$user' and password = '$password'"));
    $queryWorker = mysqli_fetch_array(mysqli_query($db, "SELECT * FROM workers WHERE username = '$user' and password = '$password'"));
    $queryLawer = mysqli_fetch_array(mysqli_query($db, "SELECT * FROM lawers WHERE username = '$user' and password = '$password'"));

    // Possible actions
    if($queryClient != ""){
        $_SESSION['login_ok'] = true;
        $_SESSION['user'] = $user;
        $_SESSION['id_user'] = $queryClient['client_ID'];

        // Codificate session and save in cookie if openSession exist
        $dataSession = session_encode();

        if(isset($_POST['openSession'])){
            setcookie("session", $dataSession, time()+(60*60*60), "/");
        }

        header("Location: ../personalArea/Client/");
    }elseif($queryWorker != ""){
        $_SESSION['login_ok'] = true;
        $_SESSION['user'] = $user;
        $_SESSION['id_user'] = $queryWorker['worker_ID'];

        // Encode session and save in cookie if openSession exist
        $dataSession = session_encode();

        if(isset($_POST['openSession'])){
            setcookie("session", $dataSession, time()+(60*60*60), "/");
        }

        header("Location: ../personalArea/Worker/");
    }elseif($queryLawer != "" && $password == 'admin'){
        $_SESSION['login_ok'] = true;
        $_SESSION['user'] = "admin";
        $_SESSION['id_user'] = 0;

        // Codificate session and save in cookie if openSession exist
        $dataSession = session_encode();

        if(isset($_POST['openSession'])){
            setcookie("session", $dataSession, time()+(60*60*60), "/");
        }

        header("Location: ../personalArea/Admin/");
    }else{
        header("Location:index.php?error=true");
    }
}
?>
<!-- /Form action -->
```

Este fragmento de código valida el formulario y redirige al usuario que accede a su área personal, creando los datos de sesión pertinentes. Además, en caso de que el usuario seleccione la casilla habilitada para mantener la sesión abierta, se crea una cookie con dicha finalidad.

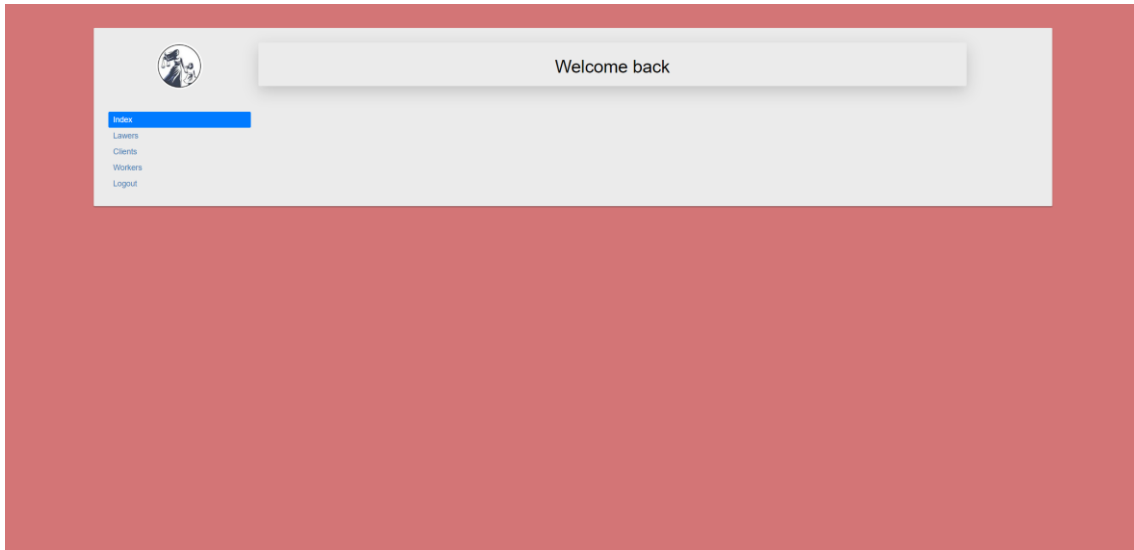
Además, se ha tenido que programar cada opción que tiene disponible cada usuario dentro de su área personal.

El archivo "connectDB.php" contiene la función de conexión con la base de datos. El código es el siguiente:

```
<?php
function connectDB() {
    $db = mysqli_connect("127.0.0.1", "root", "", "proyectoFinal");
    if (!$db) {
        echo "Error: No se pudo conectar a MySQL." . PHP_EOL;
        echo "Error de depuración: " . mysqli_connect_errno() . PHP_EOL;
        echo "Error de depuración: " . mysqli_connect_error() . PHP_EOL;
        exit;
    }
    mysqli_set_charset($db, 'utf8');
    return $db;
}
?>
```





## ADMIN

En caso de que el usuario se identifique como administrador, accederá a la raíz de su área personal, en el que se le mostrará el index de la susodicha, donde tendrá acceso a todas sus acciones.



### Lawers

Los archivos que se encuentran en el interior de la carpeta “Lawers” son los siguientes:

-  create.php
-  delete.php
-  index.php
-  modify.php

El archivo “index.php” nos mostrará un listado con todos los abogados, junto con iconos para modificar o borrar.

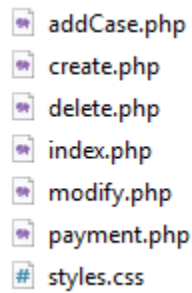
El archivo “create.php” muestra un formulario en el que insertar los datos suficientes para crear un abogado y, posteriormente, los inserta en la base de datos como un usuario nuevo.

El archivo “delete.php” elimina el abogado en el que se haya pinchado en el icono del listado de “index.php”.

El archivo “modify.php” nos muestra el mismo formulario que “create.php” solo que, en este caso, nos rellena las casillas con los datos del abogado en el que se ha pinchado en el icono de la lista de “index.php”.

## Clients

Los archivos de la carpeta “Clients” son los siguientes:



A vertical list of files with icons to their left: a purple document icon for .php files and a blue hash icon for .css files.

- addCase.php
- create.php
- delete.php
- index.php
- modify.php
- payment.php
- styles.css

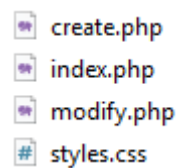
Los archivos “index.php”, “create.php”, “delete.php” y “modify.php” desempeñan la misma función que los de “Lawers”, solo que en este caso enfocado a los clientes.

El archivo “addCase.php” añade un caso a la base de datos, enlazándolo con un cliente y un abogado.

El archivo “payment.php” inserta un pago/cobro en la cuenta del cliente.

## Workers

Los archivos de la carpeta “Workers” son:



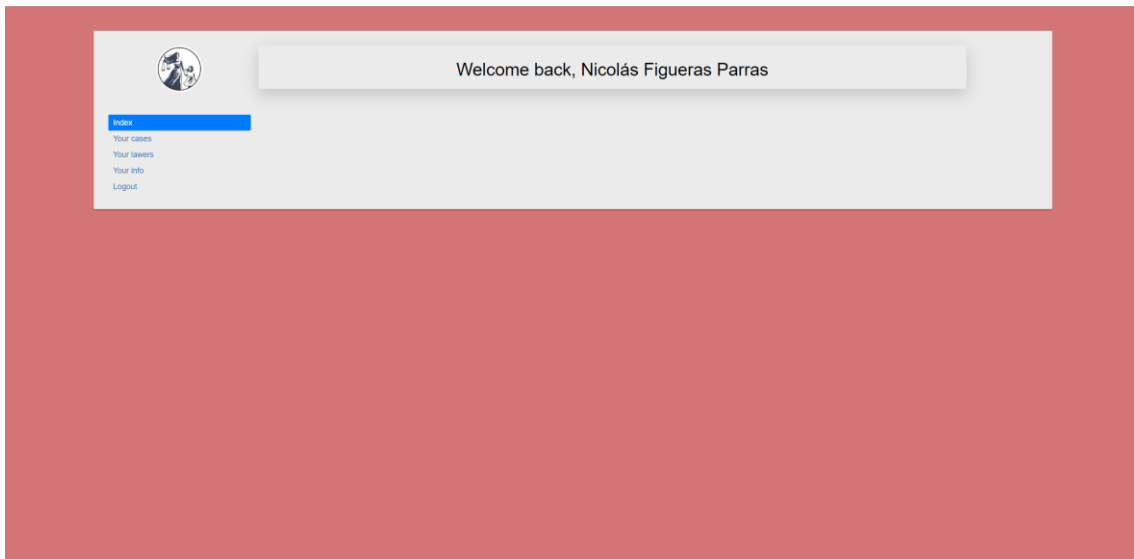
A vertical list of files with icons to their left: a purple document icon for .php files and a blue hash icon for .css files.

- create.php
- index.php
- modify.php
- styles.css

Tienen la misma función que los archivos de “Lawers” solo que, en este caso, aplicados a los trabajadores.

## CLIENT

Al acceder, verá lo siguiente:



### Your cases

La carpeta "Cases" contiene un archivo llamado "index.php", donde se listan todos los casos correspondientes al cliente identificado.

### Your lawyers

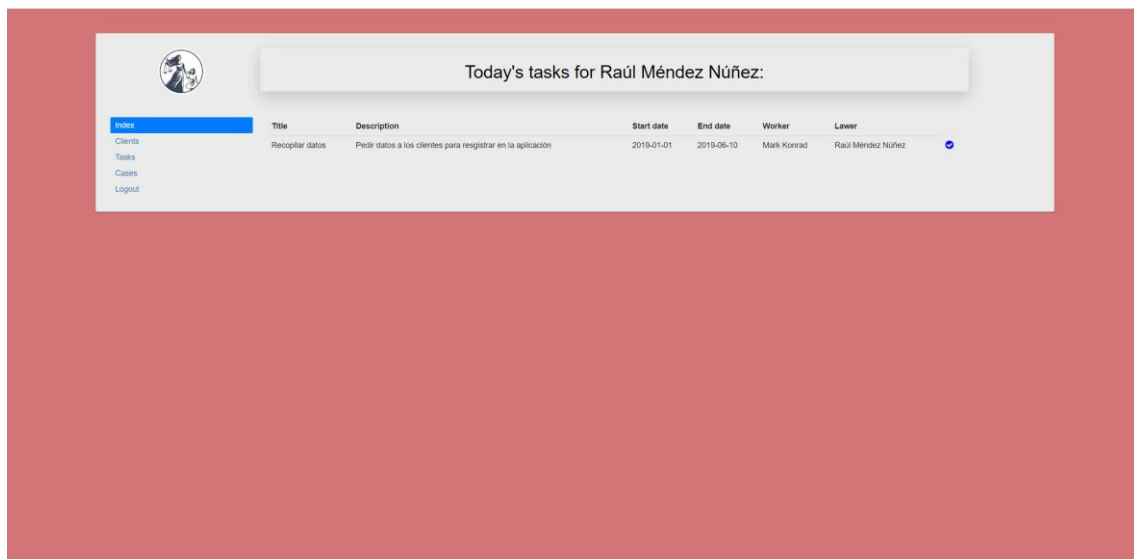
Contiene un archivo "index.php" que muestra los datos de cada abogado que lleva sus casos.

### Your info

La carpeta "Info" contiene los archivos "index.php" y "modify.php", los cuales listan la información del cliente y le permiten editarla.







## WORKER

Cuando un trabajador accede, lo primero que ve son las tareas que finalizan en el mismo día:



## Clients

En el interior de la carpeta “Clients” se encuentran los siguientes archivos:

-  addCase.php
-  create.php
-  delete.php
-  index.php
-  modify.php
-  payment.php







Los archivos “index.php”, “create.php”, “delete.php” y “modify.php” hacen lo mismo que los archivos ya mencionados con el mismo nombre en otros apartados, pero en este caso, listan, crean, modifican y eliminan clientes.

El archivo “addCase.php” añade un caso a la base de datos, enlazándolo con un cliente y un abogado.

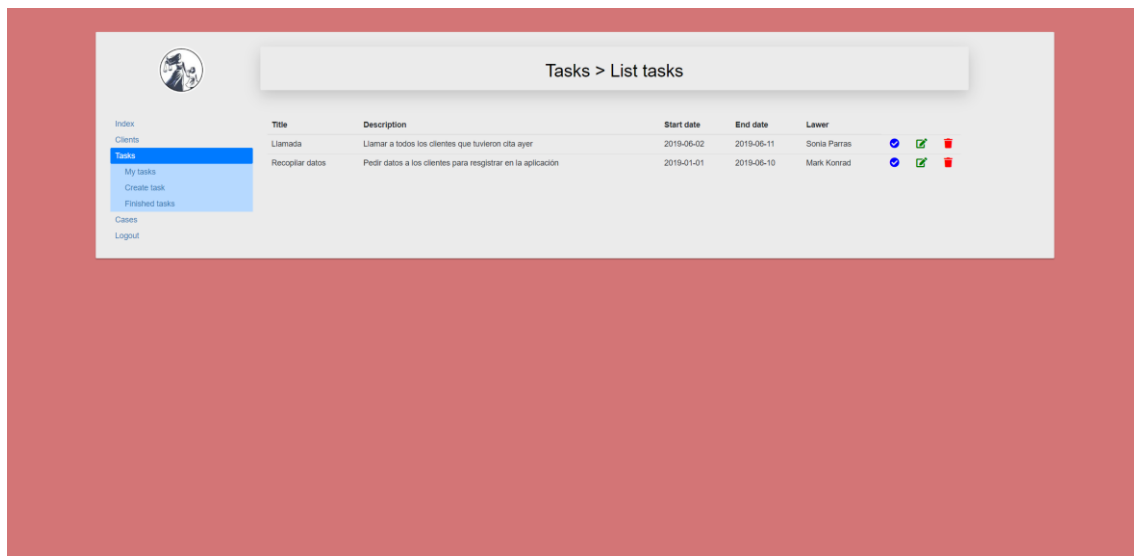
El archivo “payment.php” inserta un pago/cobro en la cuenta del cliente.

## Tasks

La carpeta “Tasks” contiene los siguientes archivos:

 create.php  
 delete.php  
 done.php  
 finished.php  
 index.php  
 modify.php





El archivo “index.php” lista las tareas que hay asociadas al trabajador que se ha identificado.



Desde este listado, tendrá la opción de marcar una tarea como finalizada (archivo “done.php”), modificar la tarea (archivo “modify.php”) o eliminarla (“delete.php”). Desde el menú lateral también puede visualizar las tareas finalizadas (“finished.php”) y crear una tarea nueva (“create.php”).

## Cases

En esta carpeta se encuentran los siguientes archivos:

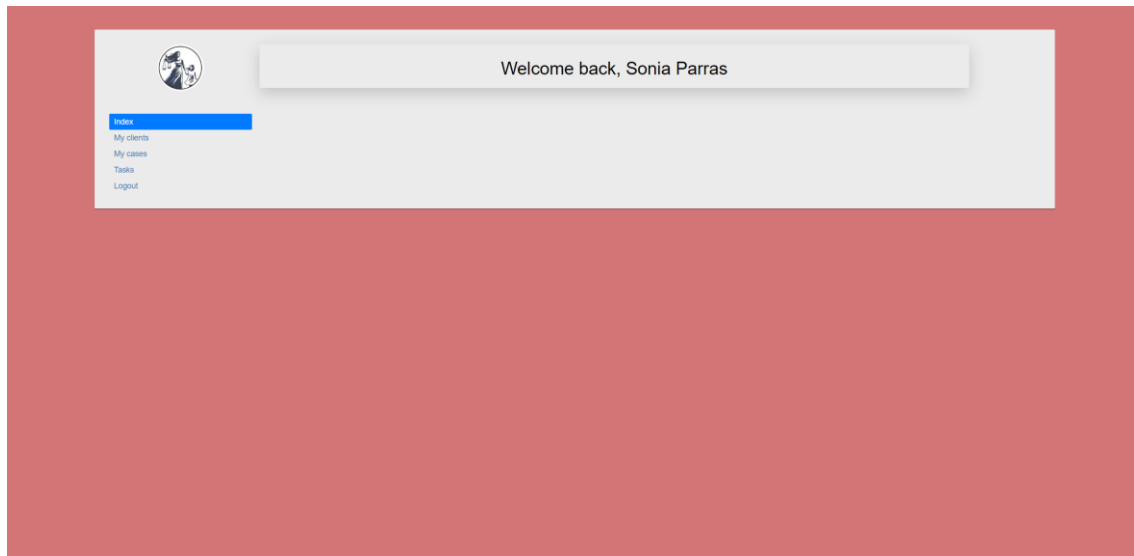
 create.php  
 delete.php  
 index.php  
 modify.php

El funcionamiento de los archivos con respecto al nombre sigue los estándares usados que se han visto anteriormente.









## LAWER

Al acceder, verá la siguiente ventana:



### My clients

En esta parte se listan todos los clientes correspondientes a los casos que lleva el abogado que se ha identificado. Los archivos son los siguientes:

-  `addCase.php`
-  `create.php`
-  `delete.php`
-  `index.php`
-  `modify.php`
-  `payment.php`

El archivo “index.php” nos mostrará un listado con todos los clientes, junto con iconos para modificar, añadir un caso a dicho cliente, modificar o borrar.

El archivo “create.php” muestra un formulario en el que insertar los datos suficientes para crear un cliente y, posteriormente, los inserta en la base de datos como un cliente nuevo.

El archivo “delete.php” elimina el cliente en el que se haya pinchado en el icono del listado de “index.php”.

El archivo “modify.php” nos muestra el mismo formulario que “create.php” solo que, en este caso, nos rellena las casillas con los datos del cliente en el que se ha pinchado en el icono de la lista de “index.php”.

El archivo “addCase.php” añade un caso a la base de datos, enlazándolo con un cliente y un abogado.

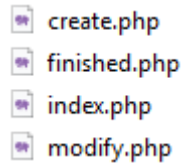
El archivo “payment.php” inserta un pago/cobro en la cuenta del cliente.

### My cases

En este directorio se encuentran los archivos “index.php”, el cual lista los casos asociados al abogado identificado; y el archivo “create.php”, cuya función es mostrar un formulario para crear un caso que, directamente, va asociado al abogado que lo crea.

### My tasks

Los archivos dentro de “Tasks” son los siguientes:



El archivo “index.php” lista las tareas que hay asociadas al abogado que se ha identificado.

Desde este listado, tendrá la opción de marcar una tarea como finalizada (archivo “done.php”), modificar la tarea (archivo “modify.php”) o eliminarla (“delete.php”). Desde el menú lateral también puede visualizar las tareas finalizadas (“finished.php”) y crear una tarea nueva (“create.php”).

## Casos de uso

Con el sistema van a interactuar tanto los clientes como los trabajadores como los abogados y el administrador.

### Cliente:

- Puede consultar sus casos, los datos de sus abogados y sus propios datos.
- Puede modificar su propia información.
- El cliente debe loguearse en el sistema para realizar cualquier operación.

### Abogado:

- Pueden consultar el listado de sus clientes, de tareas, de sus casos y consultar su propia información.
- Además, podrá crear clientes, eliminarlos, añadirles un pago/registro, un caso o modificar su información.
- También podrán crear casos y tareas.
- En último lugar, podrán ver las tareas finalizadas, modificarlas, eliminarlas o marcarlas de finalizadas.

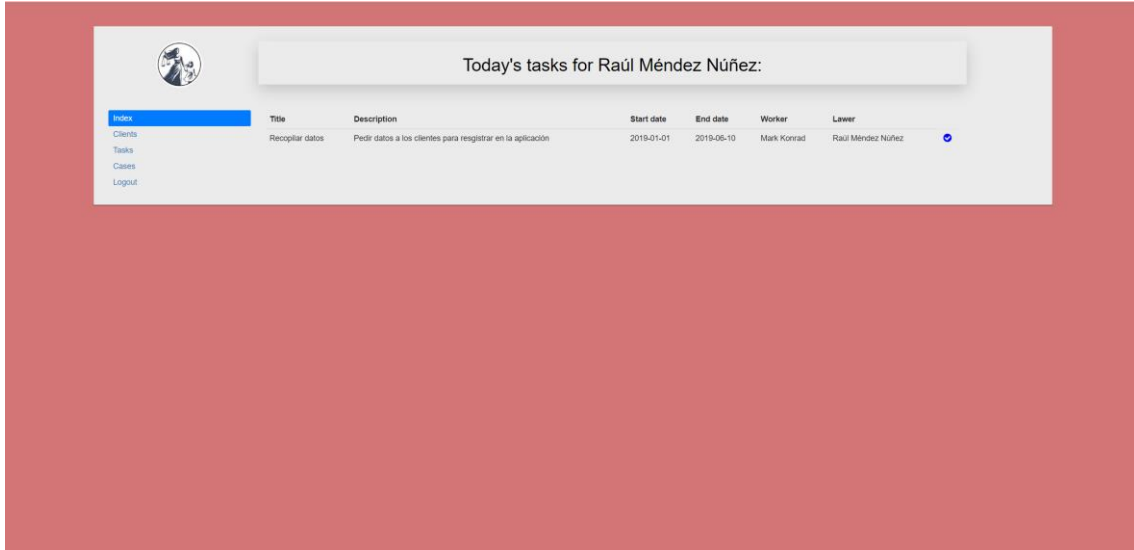
### Administrador:

- Es el encargado de gestionar los clientes, abogados y trabajadores.
- Puede modificar, crear y eliminar clientes, abogados y trabajadores.
- Además, el administrador es el único que puede crear abogados y trabajadores.
- También puede realizar las mismas acciones que un abogado sobre los clientes y los trabajadores.

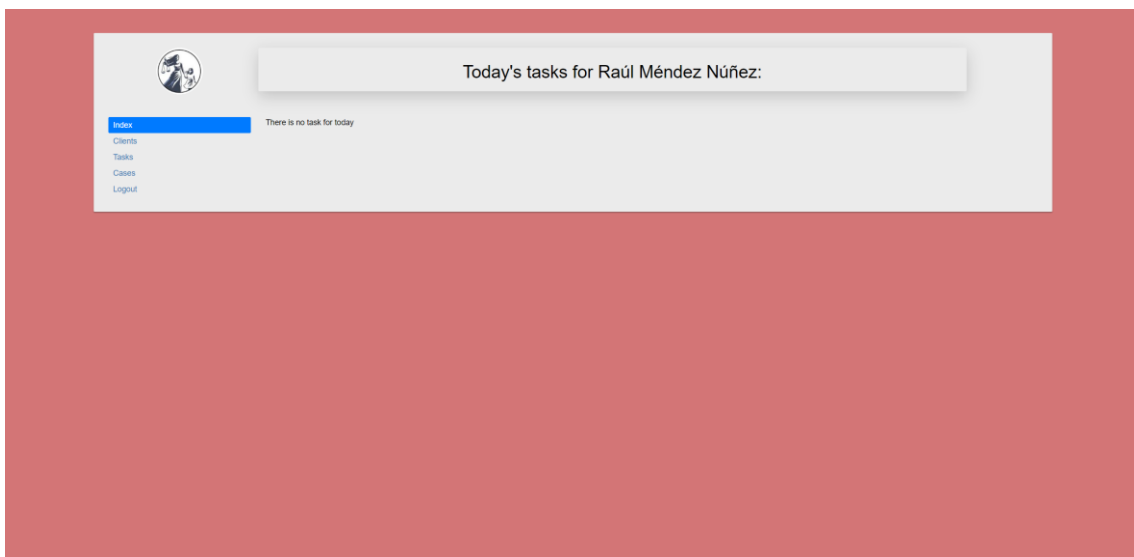
## Manual de usuario

### Trabajador

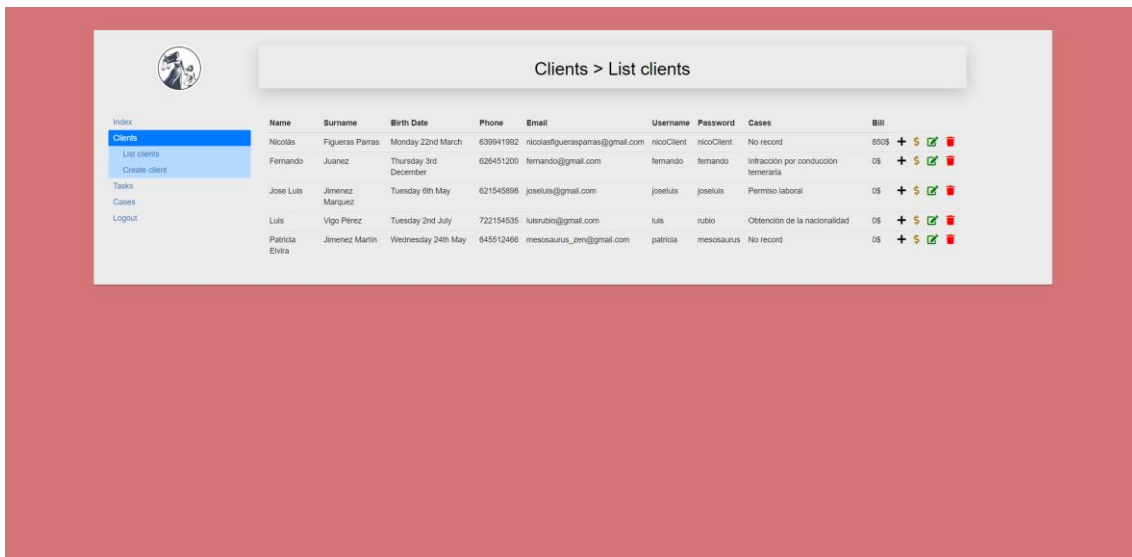
Al acceder a la aplicación, aparecerá una caja con un menú en el lateral izquierdo y una zona de contenido variable en el centro. En este caso, al principio aparecerán las tareas pendientes de realizar que finalizan en el mismo día:



En caso contrario, aparecerá un mensaje que lo indica:







Para realizar cambios o consultas sobre los clientes, accedemos a dicha sección en el menú lateral.

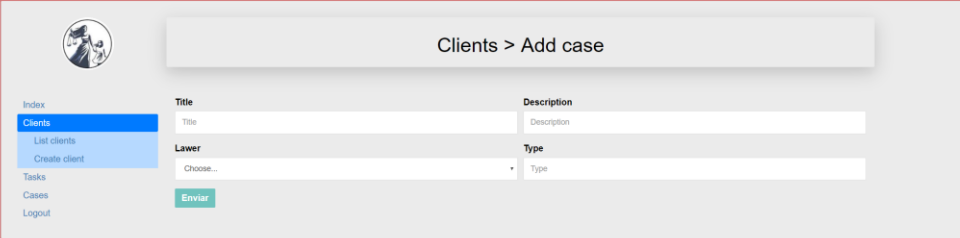


Como puede ver, se despliega el menú lateral y el contenido central cambia.

Los botones del listado tienen cada uno una función distinta:

	Éste botón permite añadir un caso y asociarlo con dicho cliente.
	Para añadir un pago o cobro a un cliente, pulse en éste símbolo.
	Sirve para modificar la información de un cliente.
	Con éste botón, eliminamos un cliente. Antes de hacerlo, el sistema nos pide confirmar la acción.

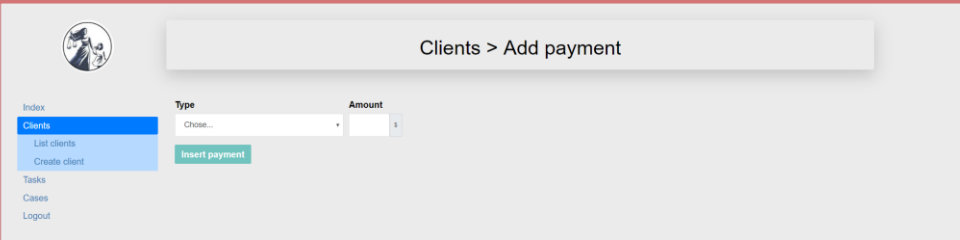
Al pulsar en el primer símbolo, tendremos la siguiente vista:



The screenshot shows a web application interface for adding a new case. On the left is a sidebar menu with a profile icon at the top, followed by 'Index', 'Clients' (highlighted in blue), 'List clients', 'Create client', 'Tasks', 'Cases', and 'Logout'. The main content area has a header 'Clients > Add case'. Below the header, there are four input fields: 'Title' and 'Description' (both text inputs), 'Lawer' (a dropdown menu with 'Choose...' selected), and 'Type' (a dropdown menu with 'Type' selected). A green 'Enviar' button is located below the 'Lawer' field.

En “Title”, insertaremos el título del caso. En “Description”, lo correspondiente. Para asignarle un abogado, tendremos un selector con el nombre de todos los abogados. En la última casilla, se escribirá el tipo de caso del que se trata.

Al pulsar el segundo botón, accederemos a la siguiente vista:



The screenshot shows a web application interface for adding a new payment. On the left is a sidebar menu with a profile icon at the top, followed by 'Index', 'Clients' (highlighted in blue), 'List clients', 'Create client', 'Tasks', 'Cases', and 'Logout'. The main content area has a header 'Clients > Add payment'. Below the header, there are two input fields: 'Type' (a dropdown menu with 'Choose...' selected) and 'Amount' (a text input with a currency symbol '€' on the right). A green 'Insert payment' button is located below the 'Type' field.

Aquí debemos escoger el tipo de pago que es y la cantidad correspondiente.

Con el tercer botón, modificaremos los datos del cliente en un formulario igual que el usado para crearlo, solo que éste vendrá relleno con los datos almacenados:

Clients > Modify client

First name	Last name		Birth date
Nicolas	Figueras Parras		22/03/1999
Phone number	Email	Username	Password
639941992	nicolasfiguerasparras@gmail.com	nicoClient	*****

[Modify](#)

Si queremos modificar o consultar la información referente a las tareas, debemos pulsar “Tasks” en el menú lateral.

Tasks > List tasks

Title	Description	Start date	End date	Lawyer	
Llamada	Llamar a todos los clientes que tuvieron cita ayer	2019-06-02	2019-06-11	Sonia Parras	<a href="#">●</a> <a href="#">✓</a> <a href="#">✖</a>
Recopilar datos	Pedir datos a los clientes para registrar en la aplicación	2019-01-01	2019-06-10	Mark Konrad	<a href="#">●</a> <a href="#">✓</a> <a href="#">✖</a>

Desde aquí, podremos marcar de finalizada una tarea con el primer botón del listado. El segundo y el tercero cumplen la misma función que los similares vistos anteriormente.

En caso de querer consultar las tareas ya finalizadas, debemos irnos a la opción del menú “Finished tasks”. Para crearla, hay que acceder a “Create task” y rellenar el formulario:

The screenshot shows the 'Tasks > Create task' form. On the left is a sidebar menu with options: Index, Cliente, Tasks (selected), My tasks, Create task, Finished tasks, Cases, and Logout. The main form area has a header 'Tasks > Create task'. Below the header, there are input fields for 'Title' (with placeholder 'Insert title') and 'Description' (with placeholder 'Insert description'). Below these are dropdown menus for 'Lawyer' and 'Worker' (both with 'Choose...' text), and date pickers for 'Start date' and 'End date' (both with 'dd/mm/aaaa' text). A green 'Submit' button is at the bottom left of the form.

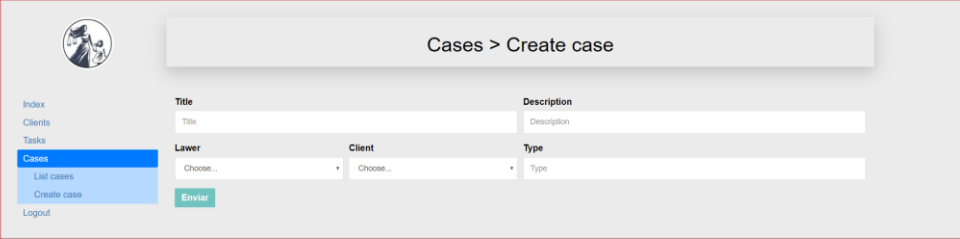
Al pulsar en la sección “Cases”, iremos a la siguiente ventana:

The screenshot shows the 'Cases > List cases' table. The sidebar menu is the same as in the previous screenshot, but 'Cases' is now selected. The table has columns: Title, Description, Lawyer, Client, and Type. Each row represents a case and includes a green checkmark icon and a red trash can icon in the rightmost column.

Title	Description	Lawyer	Client	Type	
Infracción por conducción temeraria	El cliente pide asesoría legal para mantener su ciudadanía tras haber cometido una infracción de tráfico	Sonia Parras	Fernando Juarez	Trafico	 
Permiso laboral	El cliente desea obtener el permiso laboral en el país	Noemi Casas	Jose Luis Jimenez Marquez	Permiso laboral	 
Obtención de la nacionalidad	El cliente desea obtener la nacionalidad por medio de un familiar que reside en el país	Sonia Parras	Luis Vigo Perez	Nacionalidad	 



En esta sección, podremos modificar los datos de los casos, al igual que eliminarlos. Además, accediendo a la sub-opción “Create case”, podremos crear nosotros uno:

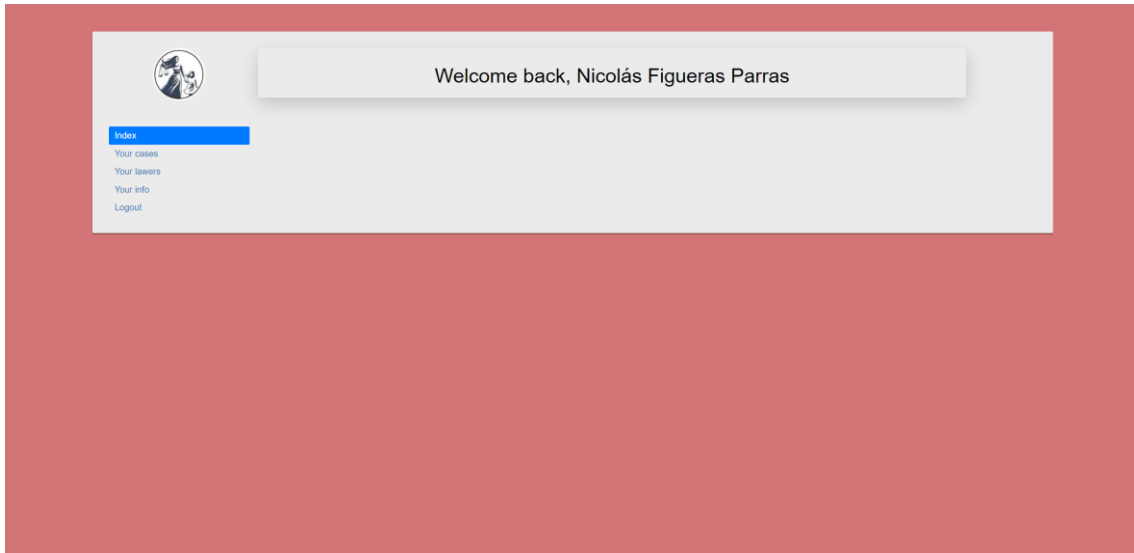


The screenshot shows a web application interface for creating a new case. The page has a red background. On the left, there is a sidebar with a profile icon and a menu with the following items: Index, Clients, Tasks, Cases (highlighted in blue), List cases, Create case (highlighted in light blue), and Logout. The main content area is titled 'Cases > Create case' in a grey header. Below the header, there is a form with the following fields: 'Title' (text input), 'Description' (text input), 'Lawyer' (dropdown menu with 'Choose...' selected), 'Client' (dropdown menu with 'Choose...' selected), and 'Type' (text input). A green 'Enviar' button is located at the bottom left of the form.

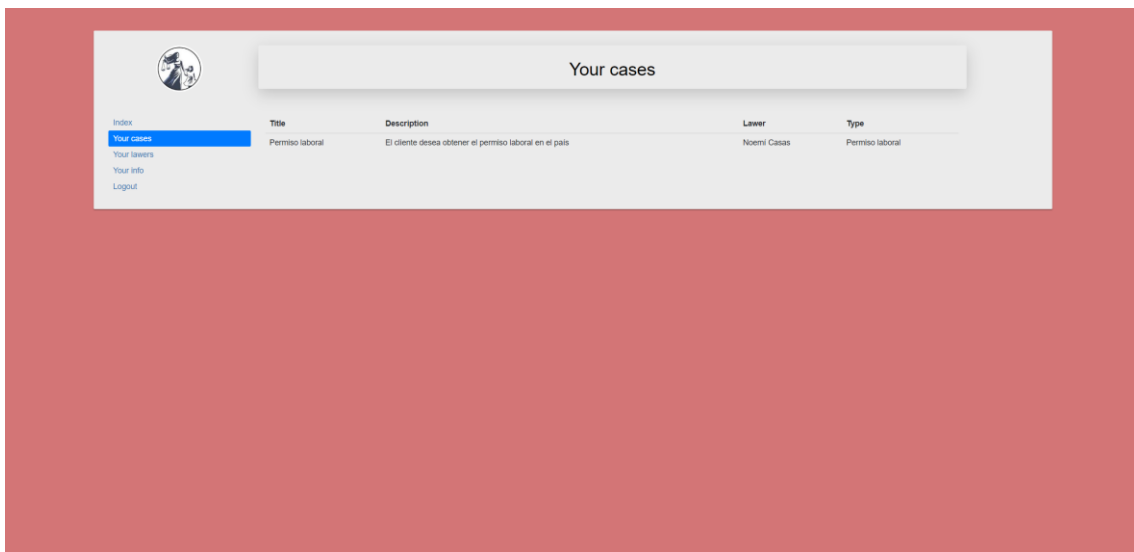
Para salir de la aplicación, hacemos click en “Logout” y volveremos a la zona de acceso.

## Cliente

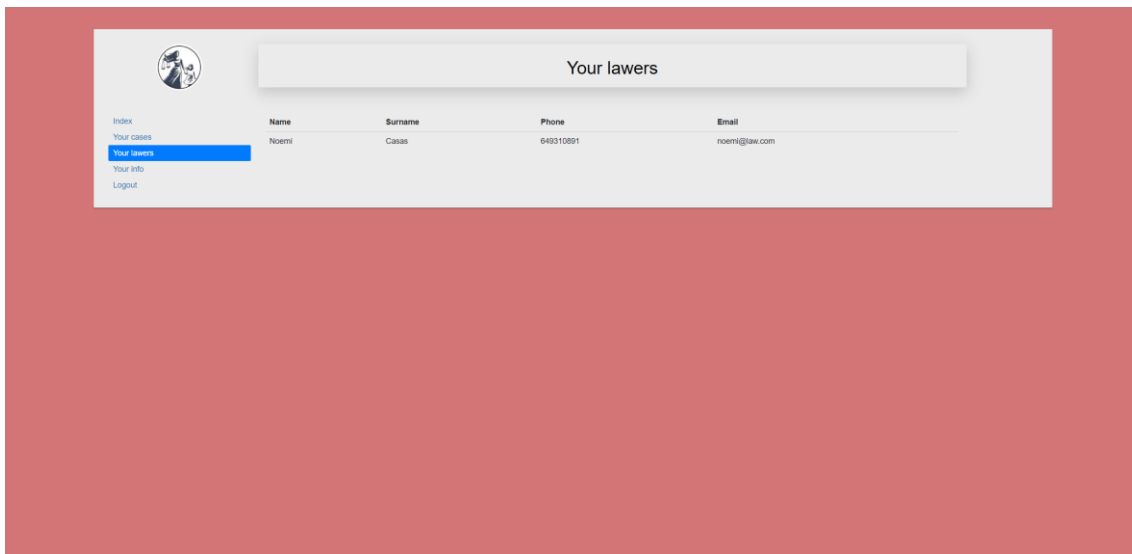
Al acceder a la aplicación, aparecerá una caja con un menú en el lateral izquierdo y una zona de contenido variable en el centro. En este caso, al principio el contenido vacío hasta que seleccionemos una opción:



La opción “Your cases” nos permite visualizar todos los casos que tenemos con los abogados de la oficina:



En “Your lawyer” accederemos a la información del abogado o de los abogados que lleven nuestros casos:



“Your info” permite visualizar la información que la oficina tiene sobre nosotros, además de poder modificar la misma:

