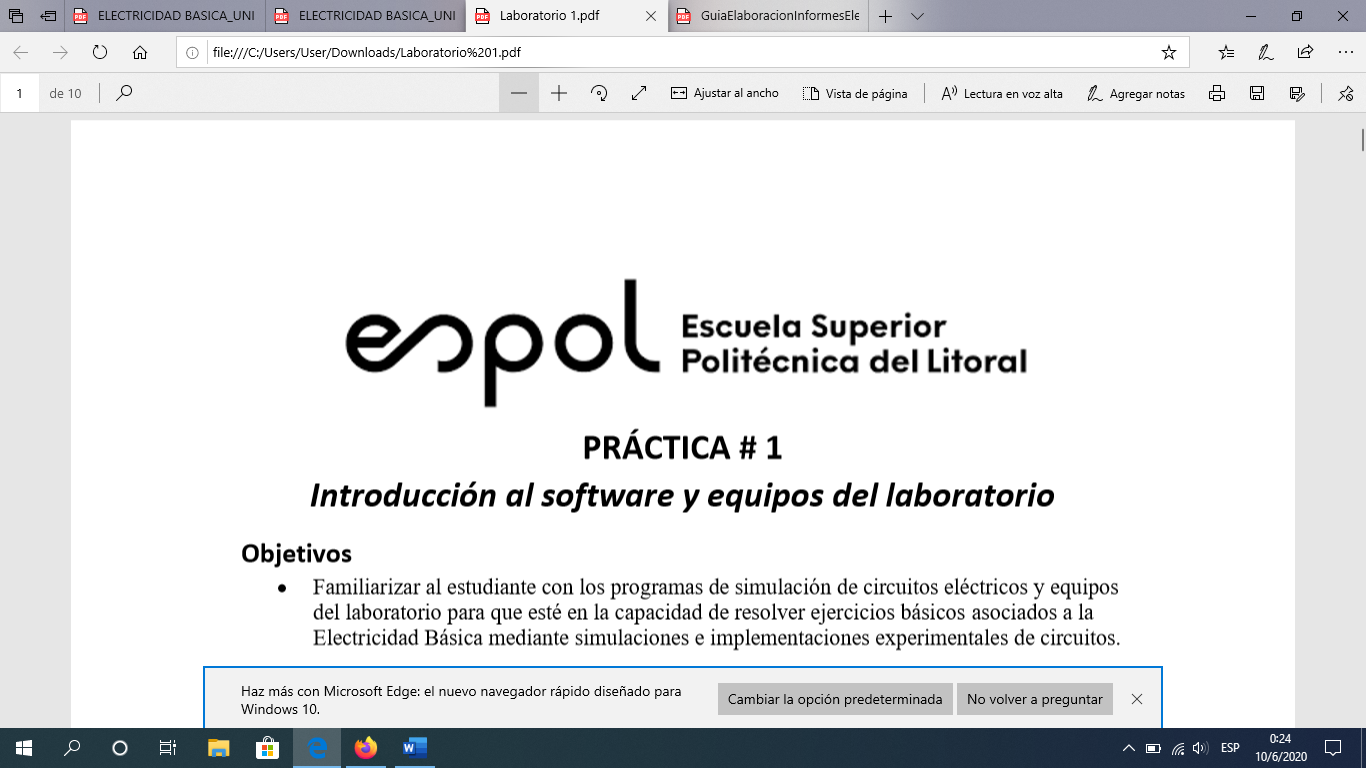
I Término 2020-2021



Programación de Sistemas Telemáticos

Informe Técnico

Tema 3: Sistema para agendamiento de turnos en establecimientos comerciales

Grupo 4

Integrantes:

José Johil Patiño Miñán

Julián EchaízArteaga

Michael QuezadaZamora

Andrés Suárez Carlín

Contenido

[1 Resumen ejecutivo 3](#_Toc49596258)

[2 Descripción del problema 3](#_Toc49596259)

[3 Objetivos específicos 3](#_Toc49596260)

[4 ¿Cómo funciona la solución? 3](#_Toc49596261)

[4.1 Para los administradores del local: 3](#_Toc49596262)

[4.2 Para los clientes: 3](#_Toc49596263)

[4.3 Firmware (Consola de conteo de persona y tiempo) 3](#_Toc49596264)

[5 ¿Qué van a construir para resolver el problema? 4](#_Toc49596265)

[6 Recursos de hardware y de software 4](#_Toc49596266)

[6.1 Recursos de hardware a utilizarse 4](#_Toc49596267)

[6.2 Recursos de software a utilizarse 4](#_Toc49596268)

[7 Explicación paso a paso de la implementación del proyecto 4](#_Toc49596269)

[8 Diagramas 4](#_Toc49596270)

[8.1 Diagramas de diseño del proyecto 4](#_Toc49596271)

[8.2 Diagrama de circuito 5](#_Toc49596272)

[8.3 Diagrama del modelo entidad-relación 5](#_Toc49596273)

[8.4 Diagrama de casos UML 5](#_Toc49596274)

[8.5 Diagrama de despliegue 5](#_Toc49596275)

[9 Descripción de los campos, tipos de datos creados en la base de datos, y el código SQL 5](#_Toc49596276)

[10 Explicación del código fuente completo desarrollado con los comentarios correspondientes 5](#_Toc49596277)

[11 Análisis de presupuesto 5](#_Toc49596278)

[12 Conclusiones 5](#_Toc49596279)

[13 Referencias bibliográficas 5](#_Toc49596280)

# Resumen ejecutivo

La finalidad del proyecto era manejo de información recopilados por la interface creada para el cliente y que permitirá el agendamiento simultaneo de horarios alojados en una base de datos y limitando el número de cliente de dicho horario. Los clientes pueden registrarse con la cedula y su nombre, una vez creado un usuario tendrá acceso a los horarios adjuntos que previamente el administrador especificó. Una vez escogido el horario la información de la aplicación se actualizará y no permitirá el agendamiento en un horario con el número de cliente este al límite del establecido por el administrador. El administrador podrá coordinar y habilitar horarios desde su propia interface. Para asegurar que el cliente este informado de que su turno está por terminar se emitirá un aviso a través de un dispositivo de sonido.

# Descripción del problema

Debido a la problemática actual provocada por el virus COVID-19 el distanciamiento social se vuelto obligatorio limitando el número máximo de clientes en locales y establecimiento. Para asegurar el control de personas se propone la creación de una aplicación para que los clientes pueda separar turnos en horarios que el administrador del local coordine para un número máximo de clientes que se encuentren en el lugar simultáneamente. Además, se debe indicar la terminación del turno del cliente a través de un dispositivo de sonido.

# Objetivos específicos

* Disponer de una base de datos que relacione los datos de los clientes con los horarios disponibles.
* Crear una aplicación web para el agendamiento simultaneo de turnos de los clientes.
* Implementar interfaces graficas para manejo de los horarios por parte de administrador y separar turnos de los clientes.

# ¿Cómo funciona la solución?

## Para el administrador del local:

Los administradores del local deben registrar su local en la base de datos. Deben especificar el nombre de su negocio, el tipo de producto o servicio que ofrecen, los horarios de atención y capacidad máxima de personas en el local. Los horarios de atención pueden ser modificados dependiendo a la necesidad del local.

**Nota:** Para la presentación del proyecto existirá únicamente un local, por lo cual no es necesario el registro del local y el administrador es único.

## Para los clientes:

Primero los clientes deben registrarse mediante su número de cédula, nombre, apellido, corre electrónico y contraseña. Una vez que estén registrados, la interfaz mostrará la lista de locales comerciales existente en la base de datos y el número de personas que existen actualmente en el local. Al dar click sobre cierto local comerciales, se desplegarán las franjas horarias disponibles para el cliente. Si en la franja horaria actual no se ha alcanzado el límite de citas en ese horario. Además, una vez que el usuario haya agendado la cita. No se permitirá que un usuario agende dos citas en el mismo horario, ni que escoja horarios que ya han vencido. Cuando un cliente ha escogido el horario, le llegará inmediata un correo recordándole la asistencia y avisándole que la alarma sonará 30 minutos antes de la hora límite. También el cliente puede cambiar su contraseña.

**Nota:** Para la simulación del proyecto el usuario podrá escoger únicamente horarios de un solo negocio.

## Firmware (Conteo de turnos disponibles y alarma)

Cada local deberá adquirir una consola que contiene el circuito mostrado posteriormente. La placa de Arduino Ethernet se encontrará directamente conectada al router del local.

Para restringir el número de personas que pueden asistir simultáneamente al local, la interface dispone de un sistema de registro, en el cual el usuario puede escoger el horario al cual desea asistir. El usuario no puede elegir el horario más de una vez y tampoco elegir horarios que ya han culminado.

Los horarios ingresados por el administrador son consultados por un archivo PHP (conect.php), el cual suministra tanto la hora actual, como la hora límite que el administrador ingresó para cierto horario. La hora límite la obtiene de la base de datos. El archivo PHP únicamente muestra la hora actual; además, le resta 30 minutos a la hora límite que el administrador ingresó. Si la hora límite (restada 30 minutos) es igual a la hora actual, entonces el archivo PHP también muestra dicha hora.

El archivo PHP es leído mediante la placa Arduino Ethernet Shield. Para lograr la comunicación de la placa, primero se establece una IP estática, de acuerdo a la red en la que se encuentre el local. Una vez logrado esto, la placa Arduino comenzará a realizar peticiones HTTP como cliente al archivo PHP. De esta manera, mediante código Arduino se comparará la hora actual comparada con la hora proporcionada por la base de datos y si faltan 30 minutos para que se termine la franja, entonces el buzzer sonará, indicando la finalización del turno.

# ¿Qué van a construir para resolver el problema?

Se planea construir una consola que cuente a los clientes que usan el local en ese momento. Para este objetivo se utilizará la información en la base de datos para la coincidencia con los horarios disponibles. Esta consola la cual tiene su respectivo circuito cumple la función de contador de personas, y el dato actual de cantidad personas será mostrado en la interfaz de los clientes. La consola será conectada al router del local (para la conexión a la red) y a un puerto USB de la computadora del local que funciona como fuente de alimentación del circuito. Se indicará si el horario escogido por el cliente va terminar por medio de un buzzer.

# Recursos de hardware y de software

## Recursos de hardware a utilizarse

* Buzzer: Elemento de salida que avisará cuando falten 10 minutos para que se termine la franja horaria
* Placa de Arduino Ethernet: Permitirá enviar y recibir datos a la base de datos.
* Conversor FTDI (Porque la placa Arduino Ethernet no tiene puerto USB)
* Cable de red (Conectado al enrutador del local y a la placa de Arduino Ethernet)

## Recursos de software a utilizarse

* Servidor MySQL
* PHPmyAdmin
* Servicio de hosting 000webhost
* IDE de Arduino

# Explicación paso a paso de la implementación del proyecto

## Estructura de la base de datos.

Se crearon 4 tablas cliente, administrador, turno y horario. Cada una con sus propios atributos. El cliente posee una identificación, su cedula, nombres, apellidos y correo electrónico; también una contraseña que el cliente escoja al registrarse con la cual podrá iniciar sesión.

El cliente podrá escoger horarios que cuya tabla tendrá los atributos: identificador, hora de inicio y finalización donde el cliente poda ingresar al local, la capacidad máxima que el local tiene permitido tener clientes simultáneamente, y los turnos separados para dicho horario.

La tabla “detalleturno” permite guardar información que el cliente separó para ese horario. Datos como: el horario escogido, el cliente del turno, identificador de turno.

El administrador podrá habilitar horarios y sus campos dentro de la tabla de la base de datos son: identificación con la contraseña de su sesión. Nos basamos en la tabla horario y sus atributos datetime para la verificación del cumplimiento de tiempo en que se acaba el turno.

## Diseño de las interfaces graficas

Antes del diseño de la interfaz gráfica se trabajó en la funcionalidad del proyecto. Luego de haber culminado con dicha tarea, se buscó y seleccionó el diseño de la página web. Se obtuvo el diseño de la página: <https://startbootstrap.com/themes/sb-admin-2/>

Para comenzar se escogió el diseño y el código fuente del diseño en el cual se creó el menú principal que te permite iniciar sesión con sus opciones para ingresar texto, tanto el usuario y la contraseña son requeridos para iniciar sesión. Se separó lo necesario del formato inicial del diseño y se empezó a reemplazar valores en los iconos del diseño, también se hizo un logo que represente nuestro proyecto y su nombre. Para el registro se uso una plantilla del mismo diseño y las entradas de texto para llenar los datos y el botón para confirmar los datos. Después se agrupo 2 campos en una misma línea para iniciar sesión del cliente. Asimismo, para crea los horarios hay entrada de texto los datos con la información requerida que el administrador puede llenar. La interface del cliente permite seleccionar el horario que desea escoger y un botón para confirmar el registro del turno. Se mostrará un mensaje tanto el registro del cliente fue exitoso como su tuno fue registrado.

La siguiente plantilla fue usada tanto para el registro del usuario, el ingreso del usuario a su perfil y e ingreso del administrador a su perfil.

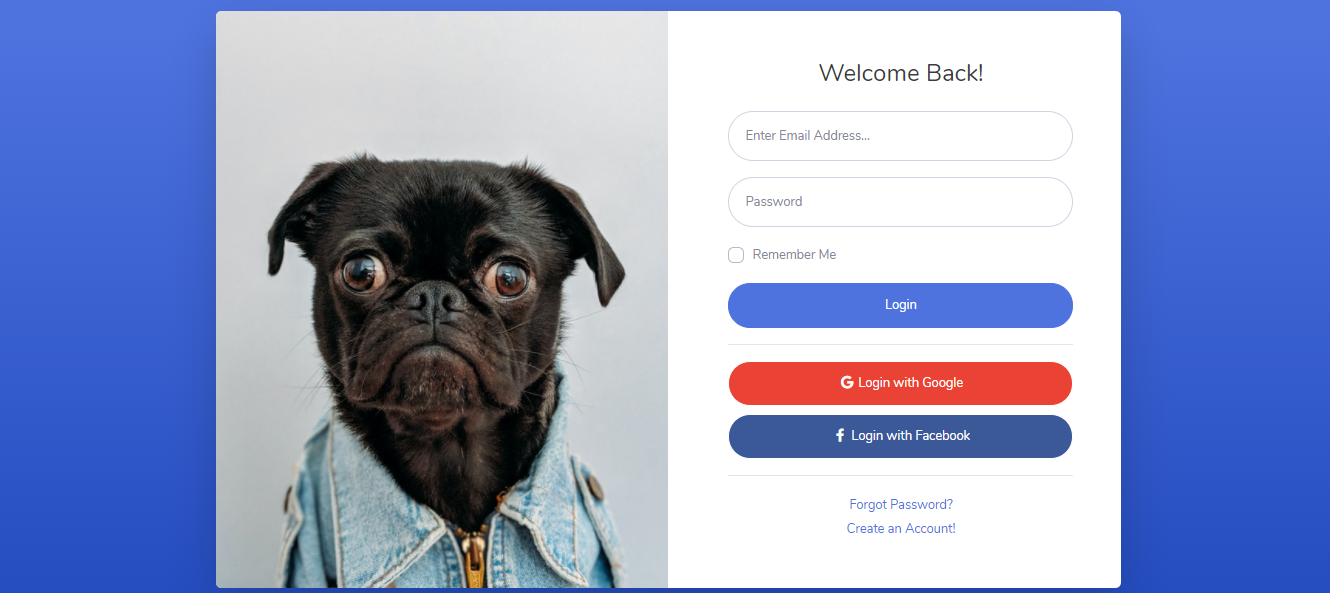


Ilustración 1 Diseño para las páginas de ingreso

La siguiente plantilla fue usada para la página principal, la lección de horarios del usuario, el registro de horarios del administrador y la visualización de citas que tiene el administrador.

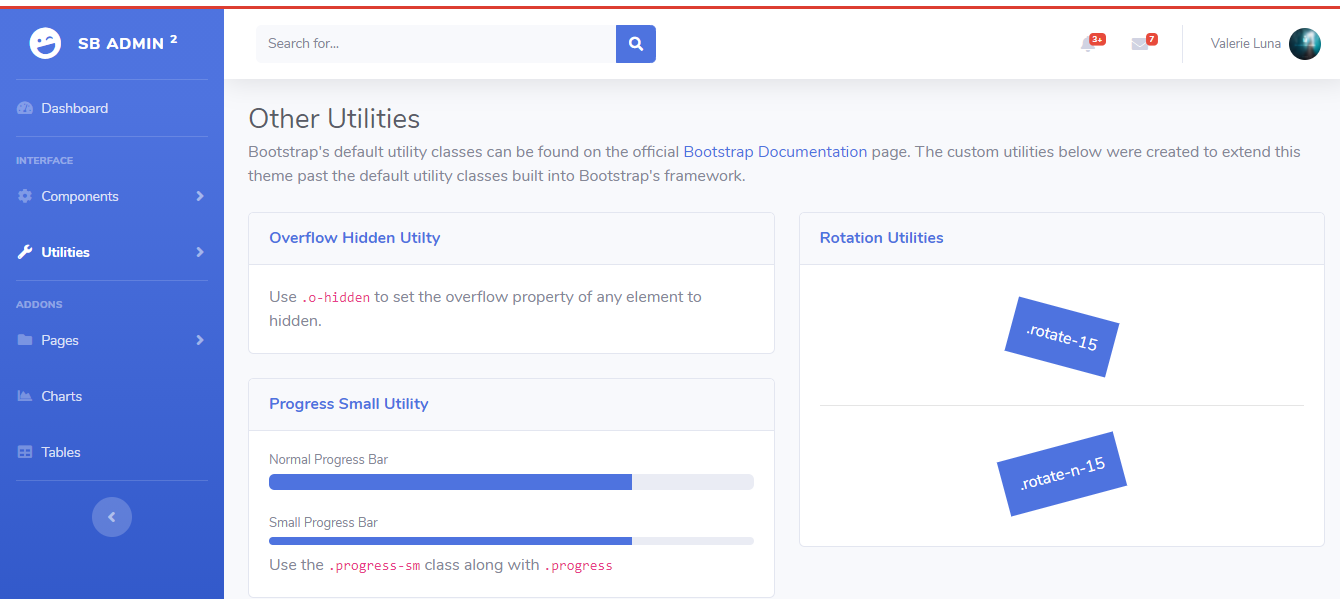


Ilustración 2 Diseño para las páginas de los perfiles de usuarios y la página principal

Algunos botones usados tienen el siguientes diseño de los botones “.btn-google” y “.btn-facebook”:

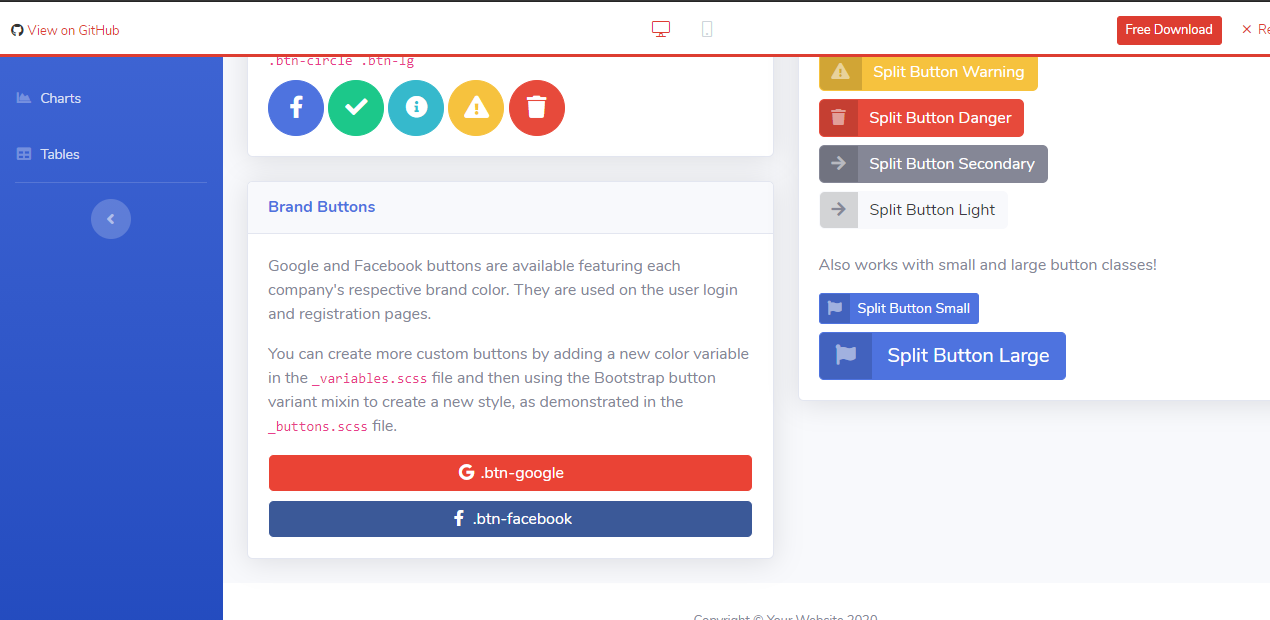


Ilustración 3 Diseño para los botones

Para las páginas que muestran mensajes de error o de ingresos o registros correctos, se empleó el mismo diseño de la Ilustración 1, pero sin botones ni entradas de texto.

## Implementación del dispositivo de sonido

Para la comunicación del Arduino de la consola se ceo un archivo PHP que sirve tanto obtener información del cliente de la base de datos como para obtener la hora actual para la verificación con el turno del cliente. Una vez obtenida la hora actual se valida con código de Arduino si el turno del cliente está por terminar, después se sube el código al Placa de Arduino Ethernet que se encuentra conectada al router del local a través del cable de red que también se verifica si hay conexión con la base de datos. Se indicará con un Buzzer cuando el turno del cliente este por terminar y se enviará un correo al cliente.

# Diagramas

## Diagramas de diseño del proyecto

Nuestro proyecto posee una base de datos que obtiene información de la interface del cliente y permite sincronizar los horarios y los espacios disponibles que se verifican a través de un dispositivo IoT y avisa al usuario que puede ingresar al local o el local tiene el máximo número de personas, a la vez que el administrador supervisa desde su interface.

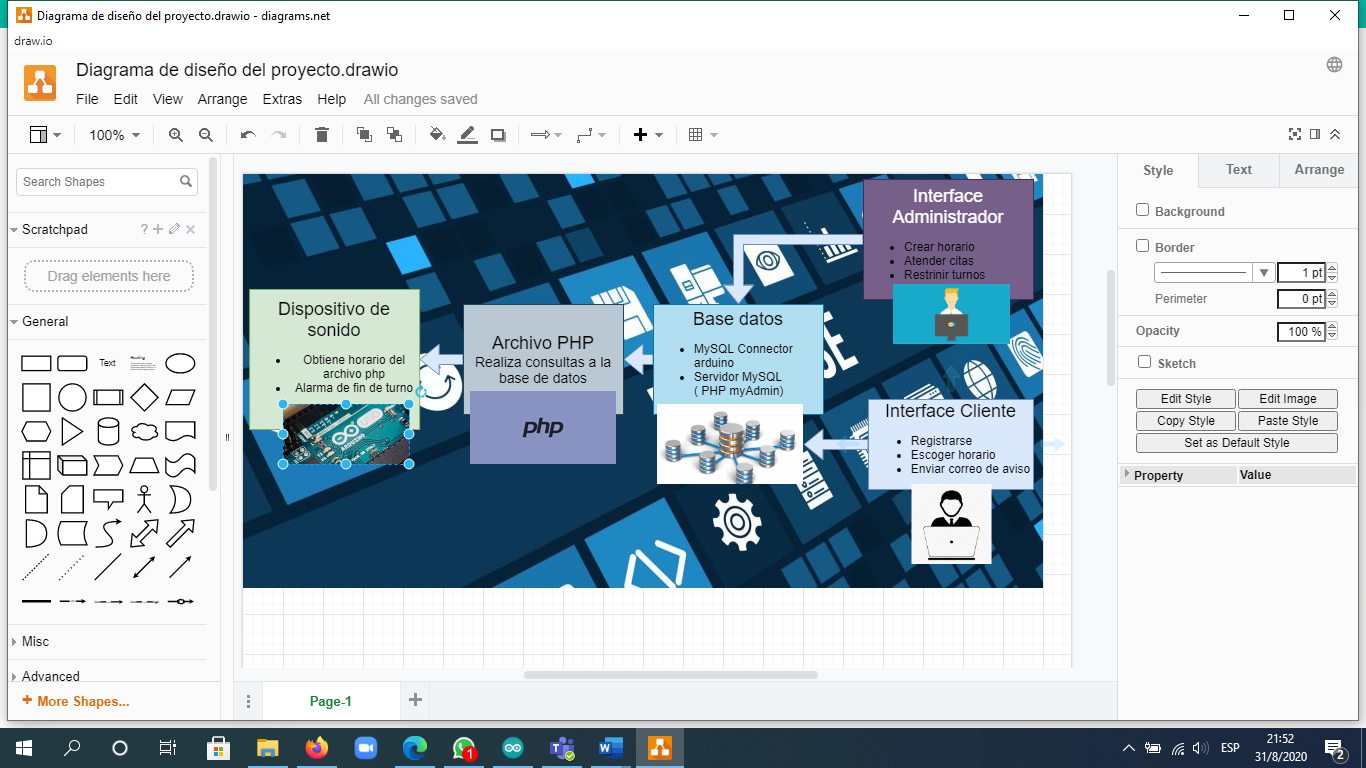


Ilustración 4 Diagrama de diseño del proyecto

## Diagrama de circuito

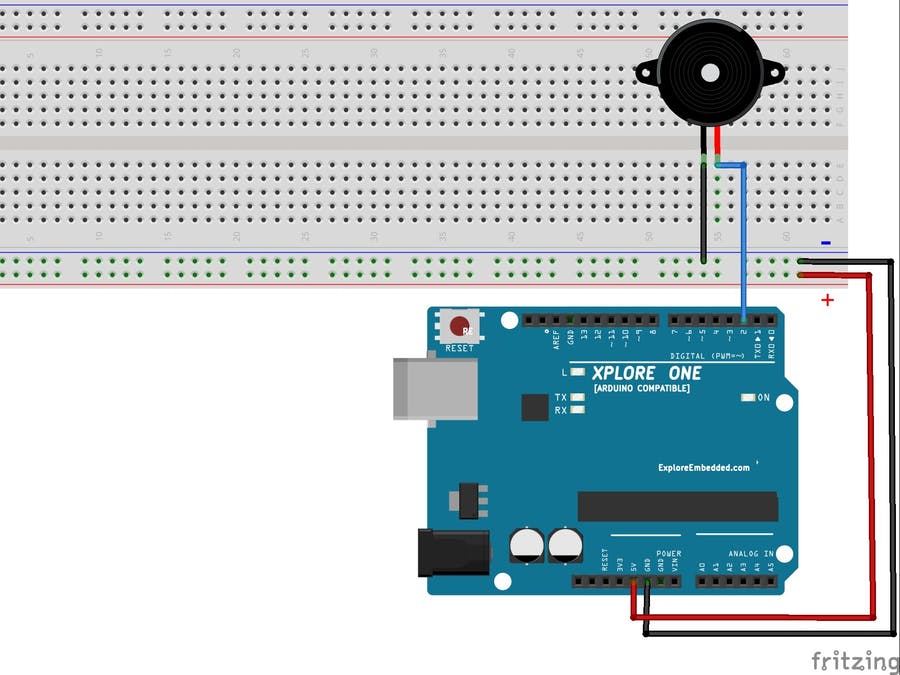


Ilustración 5 Circuito del proyecto

## Diagrama del modelo entidad-relación

### Modelo entidad-relación para varios locales

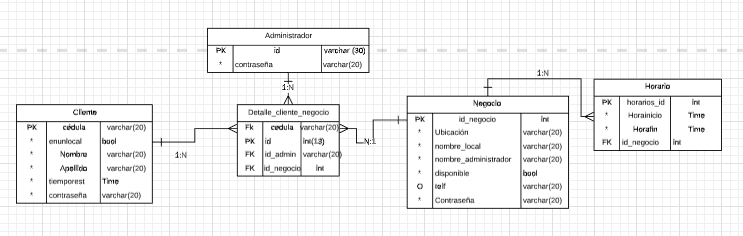


Ilustración 6 Modelo entidad relación para varios locales

### Modelo entidad-relación para un local (usado para la presentación del proyecto)

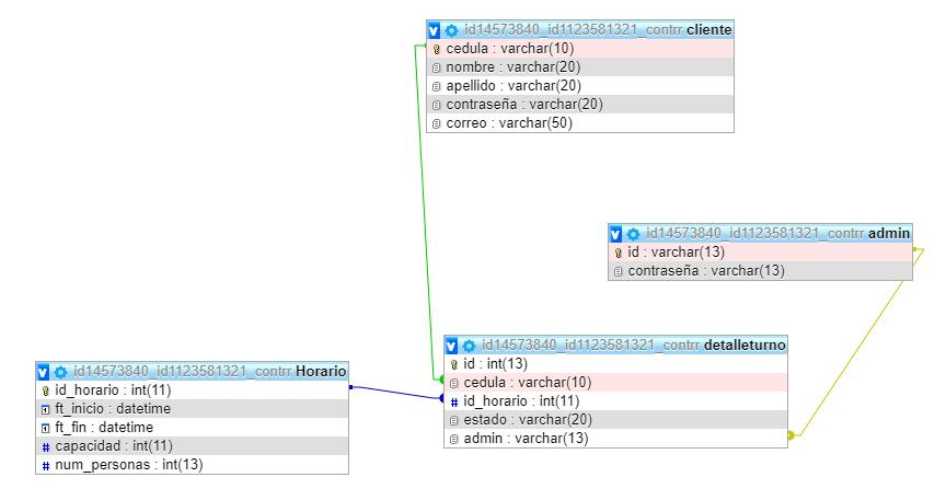


Ilustración 7 Modelo entidad relación para un solo local

## Diagrama de casos UML

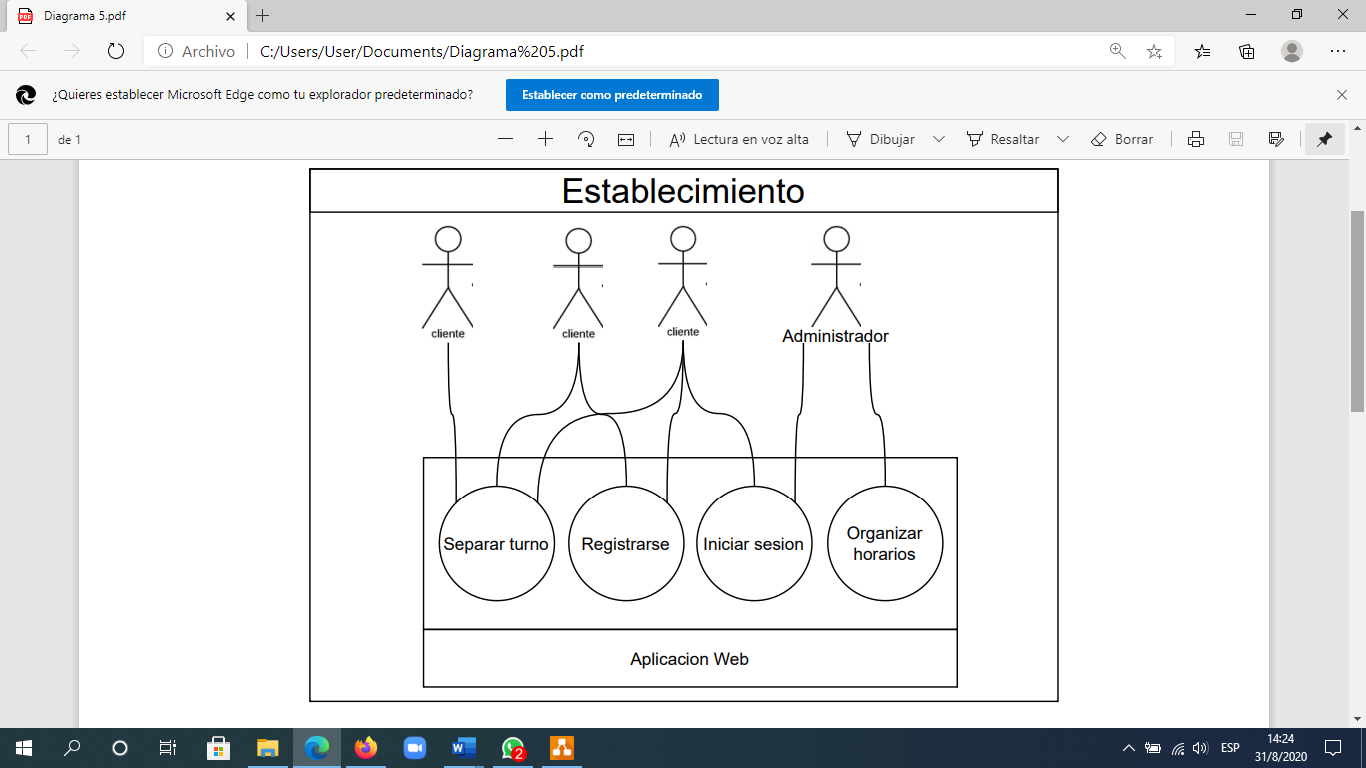


Ilustración 8 Diagrama de casos del proyecto

## Diagrama de despliegue

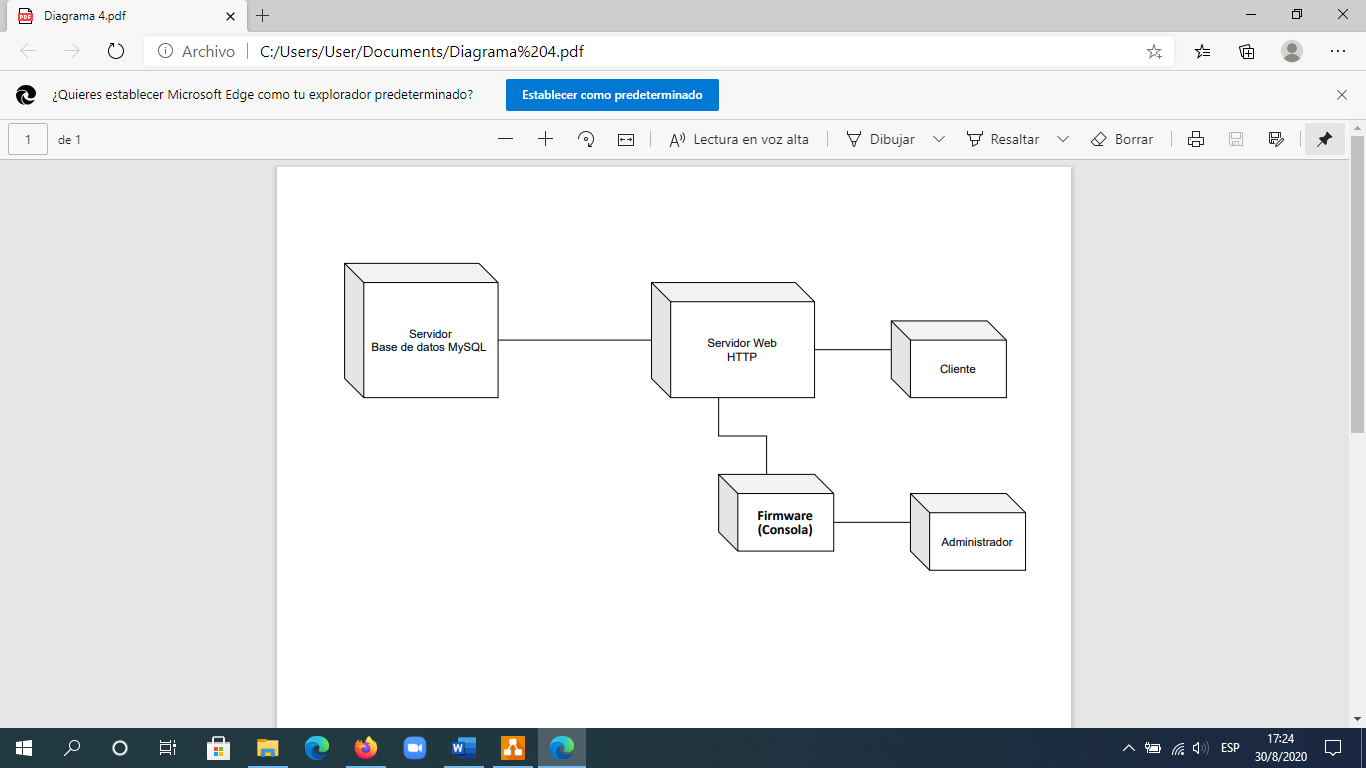
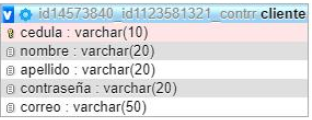


Ilustración 9 Diagrama de despliegue

# Descripción de los campos, tipos de datos creados en la base de datos, y el código SQL

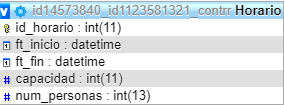
## Descripción de los campos usados en la presentación del proyecto

**Tabla cliente**

****

* **cedula**: Campo primary key, funciona como el identificador único de un cliente. La longitud y el valor numérico se restringen en el formulario de registro del usuario.
* **nombre:** Este campo almacena el nombre de los clientes.
* **apellido:** Este campo almacena el apellido de los clientes.
* **contraseña:** Este campo almacena la contraseña que los clientes ingresarán en el formulario de registro, o que modificarán en caso de que deseen cambiar su contraseña.
* **correo:** Este campo almacena el correo electrónico del usuario, que servirá posteriormente para recordar al usuario que tiene una cita en cierto horario.

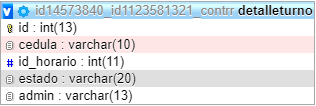
**Tabla Horario**



* **id\_horario:** Campo que permite distinguir a cierto conjunto fecha-hora como registro único. Y que permitirá al cliente escoger la hora en la que registrará una cita.
* **ft\_inicio:** Registra la hora de inicio para una franja horaria establecida por el administrador.
* **ft\_fin**: Registra la hora límite o final para una franja horaria establecida por el administrador.
* **capacidad:** Registra la capacidad máxima de turnos simultáneos que pueden agendar en cierta franja horario.
* **num\_personas:** Permite registrar el número de citas actuales para cierta hora.

Para explicar mejor la relación entre los campos capacidad y num\_persona se mencionará un ejemplo: El administrador puede definir que para cierto horario la capacidad máxima en el local será de 50 personas, 50 sería el valor para el campo capacidad en ese registro. El campo num\_persona siempre inicia en cero y comienza a incrementar la cantidad en una unidad cada vez que un cliente se registra.

**Tabla detalleturno**

****

* **id:** Campo que permite distinguir e identificar a cierto turno como registro único.
* **cedula:** Foreign key que permite detallar qué usuario está registrado en el turno.
* **Id\_horario**: Foreign key que permite detallar qué horario está registrado en el turno.
* **estado:** Permite indicar si el turno ya fue atendido o no por el administrador. El valor de inicio de este campo es “no atendido”, lo cual se controla desde el registro del cliente en el horario desde un archivo PHP. Una vez que el administrador marca como atendido esta cita, el valor del campo estado pasa a “atendido”
* **admin:** Foreign key del identificador del administrador. El administrador es único, es decir que todos los turnos serán atendidos desde una sola cuenta de administrador.

**Tabla admin**



* **id:** Permite identificar al administrador.
* **contraseña:** Permite restringir el acceso al perfil del administrador en la interfaz.

## Tipos de datos creados en la base de datos

Para las primary keys de las tablas “detalleturno” y “Horario”, se usaron tipos numéricos entero que se autoincrementan cada vez que se añade un registro. Para las primary keys de las demás tablas se utilizaron tipos de variable varchar. De la misma manera, las contraseñas de clientes y del administrador son de tipo varchar para que puedan ingresar cualquier tipo de carácter y tener una correcta seguridad de su perfil. Para los campos “ft\_fin” y “ft\_inicio”, que corresponden a los horarios de inicio y final de cierta franja horaria ingresada por el administrador, se utilizó el tipo datetime para registras fecha y horario. A continuación, se muestra una tabla que resume los campos de acuerdo con el tipo de dato usado:

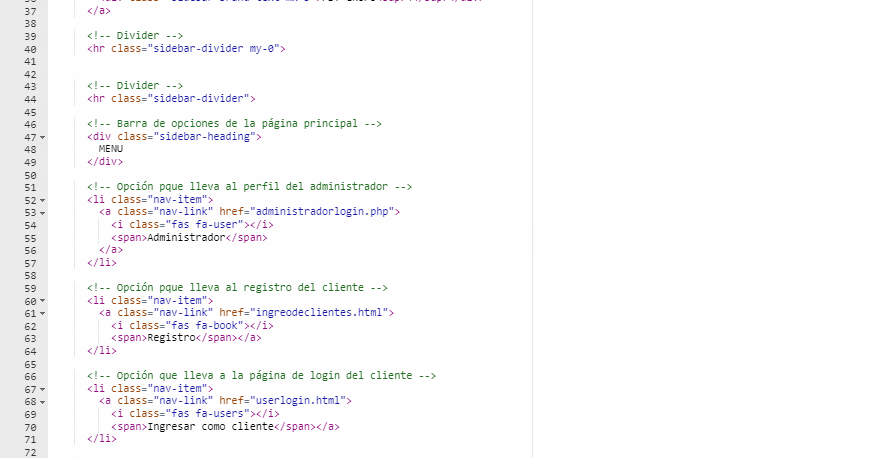
|  |  |  |
| --- | --- | --- |
| **Tipo de datos** | **Tablas** | **Campos** |
| varchar | admin | id, contraseña |
| detalle\_turno | cedula, estado, admin |
| cliente | cedula, nombre, apellido, contraseña, correo |
| datetime | Horario | ft\_inicio, ft\_fin |
| int | Horario | id\_horario, capacidad, num\_personas |
| detalle\_turno | id, id\_horario |

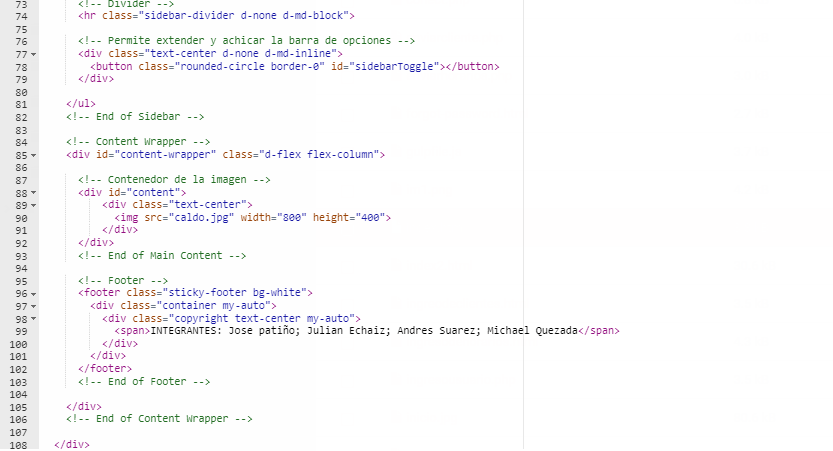
## El código SQL

|  |  |
| --- | --- |
| **Código SQL** | **Función** |
| create table admin (id primary key varchar(13), contraseña varchar(13)) | Crear tabla admin |
| create table cliente(cedula varchar(10) primary key , nombre varchar(20), apellido varchar(20), contraseña varchar(20), correo varchar(50)) | Crear tabla cliente |
| create table  detalleturno(id primary key int(13) auto\_increment, cedula varchar(13), id\_horario int(11), estado varchar(20), admin  varchar(13), FOREIGN KEY(cedula) REFERENCES cliente(cedula), FOREIGN KEY(id\_horario) REFERENCES Horario(id\_horario), FOREIGN KEY(admin) REFERENCES admin(id)) | Crear tabla detalleturno |
| create table Horario (id\_horario int(11) primary key auto\_increment, ft\_inicio datetime, ft\_fin datetime, capacidad int(11), num\_personas int(13)) | Crear tabla Horario |
| "INSERT INTO `detalleturno` (`id`, `cedula`, `id\_horario`, `estado`, `admin`) VALUES (NULL, '$cedula', '$id\_horario', 'no atendido', 'admin')"; | En el archivo agendamiento.php se usa la siguiente consulta para insertar el turno escogido por el usuario |
| "UPDATE `Horario` SET `num\_personas` = `num\_personas`+1  WHERE `Horario`.`id\_horario` = '$id\_horario'" | En el archivo agendamiento.php se usa la siguiente consulta para incrementar el número de citas que han sido tomados del horario en la tabla Horario |
| "UPDATE `admin` SET `contraseña` = '$newPass' WHERE `admin`.`id` = 'admin'" | En el archivo cambioconfadmin.php**p**ermite cambiar la contraseña del administrador |
| "INSERT INTO `cliente` (`cedula`, `nombre`, `apellido`, `contraseña`, `correo`) VALUES ('$cedula', '$nombre', '$apellido','$contrasena','$correo') | En el archivo enviarcliente.php, permite insertar un nuevo cliente en la tabla cliente |
| "INSERT INTO Horario (`id\_horario`, `ft\_inicio`, `ft\_fin`,`capacidad`,`num\_personas`) VALUES (NULL, '$dt\_in', '$dt\_fin','$capacidad','0')" | En el archivo enviarhorarios.php permite ingresar un nuevo horario elegido por el administrador |
| SELECT \* FROM `detalleturno` JOIN Horario ON detalleturno.id\_horario=Horario.id\_horario WHERE substring(Horario.ft\_inicio,1,10)='$fechahoy' | Permite seleccionar las citas existentes para el día actual, en el archivo resumen.php |
| select \* from `Horario` | En userhome.php permite mostrar al usuario los horarios ingresados por el administrador y el número de turnos disponibles. |
| UPDATE `cliente` SET `contraseña` = '$newPass' WHERE `cliente`.`cedula` = '$cedula'" | Permite que el usuario cambie su contraseña en el archivo cambioconf.php |

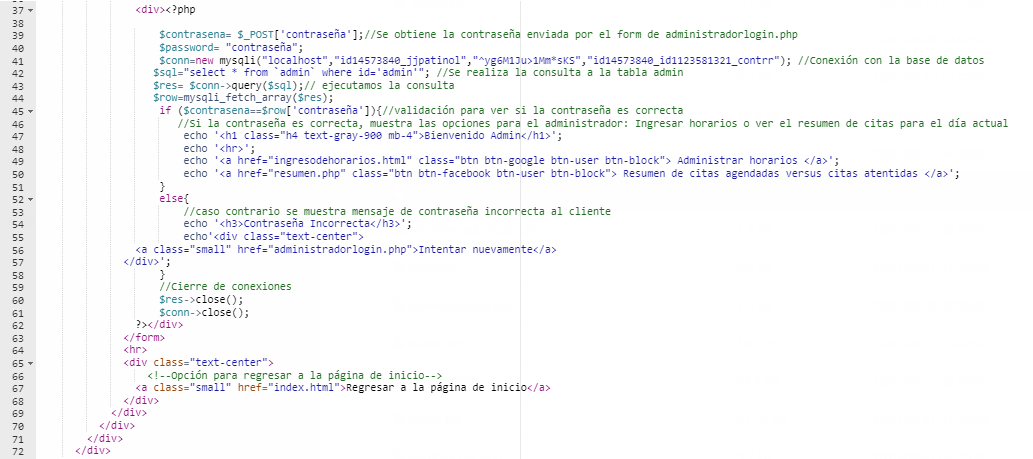
# Explicación del código fuente completo desarrollado con los comentarios correspondientes

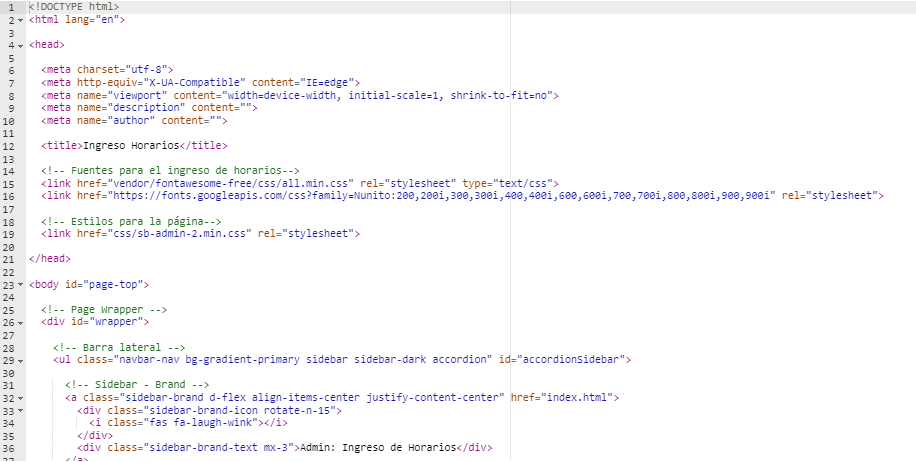
**Archivo index.html :** Muestra las opciones principales del main page: opción para ingresar como administrador, para registrarse e ingresar como cliente.



****

**Archivo admincontraseña.php :** Obtiene la contraseña ingresada por el administrador. Si es correcta muestra las opciones del administrador. Si no es correcta, muestra el rescpectivo mensaje.

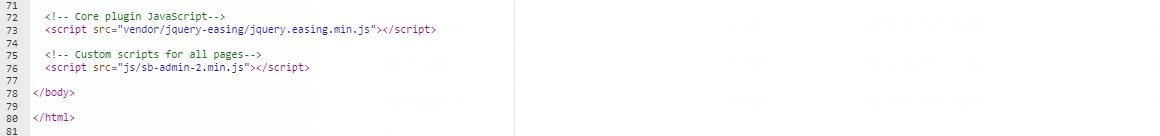


**Archivo ingresodehorarios.html :** Permite al administrador ingresar las franjas horarias, fecha y capacidad máxima de citas para esa franja.



**Archivo enviarhorarios.php :** Envía los datos sobre la franja horario ingresados por el admin hacia la base de datos y muestra mensaje de registro exitoso.

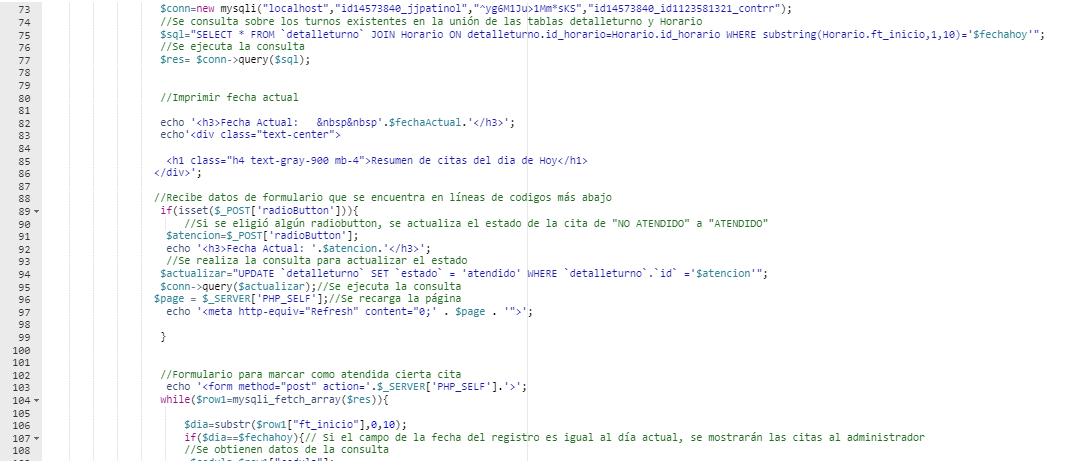


****

**Archivo resumen.php:** Permite al administrador ver e resumen de sus citas para el día presente y marcarlas como atendidas.

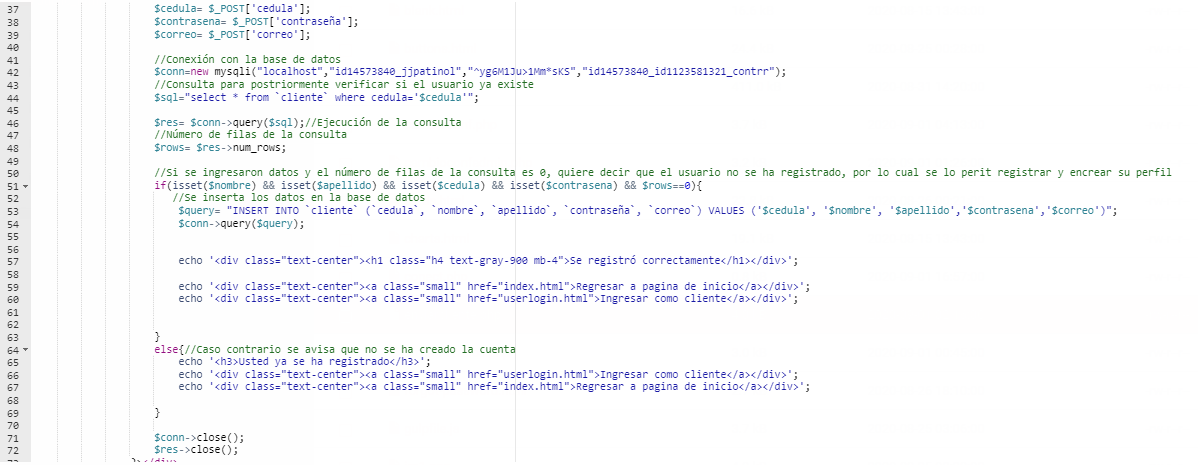




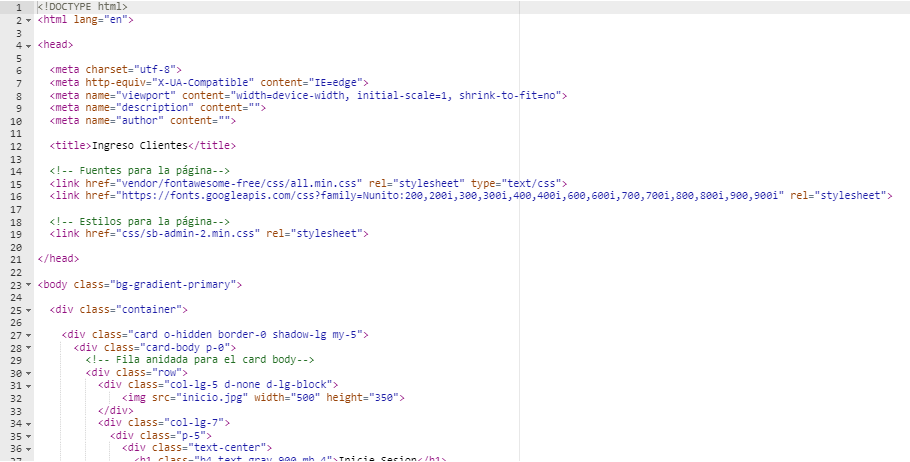




**Archivo enviarcliente.php:** Permite enviar los datos del cliente para crearle un perfil en la base datos.

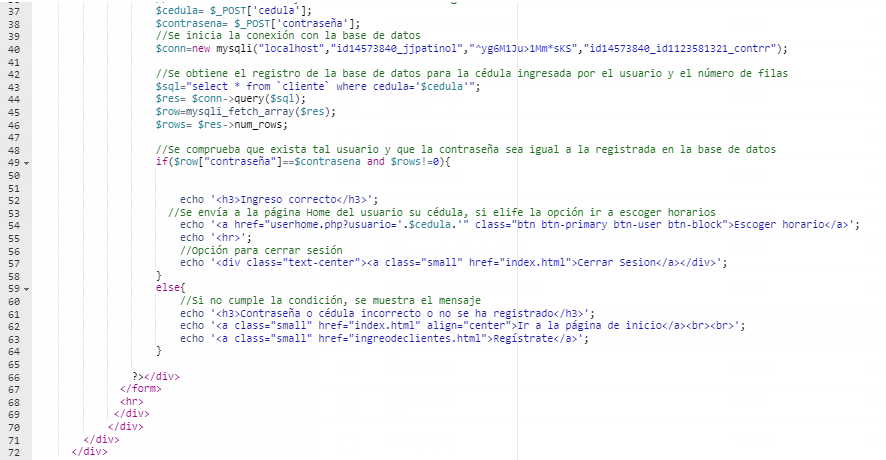


**Archivo userlogin.html:** Es la página de ingreso del usuario a su perfil. Este archivo contiene el formulario que será luego enviado a la base de datos mediante el archivo ingresousuario.php





**Archivo ingresousuario.php:** Envía los datos ingresados por el usuario en el formulario del archivo userlogin.html a la base de datos, verifica si el usuario existe y si la contraseña es correcta. Si es correcta muestra opciones propias del cliente y si no muestra un mensaje de error.



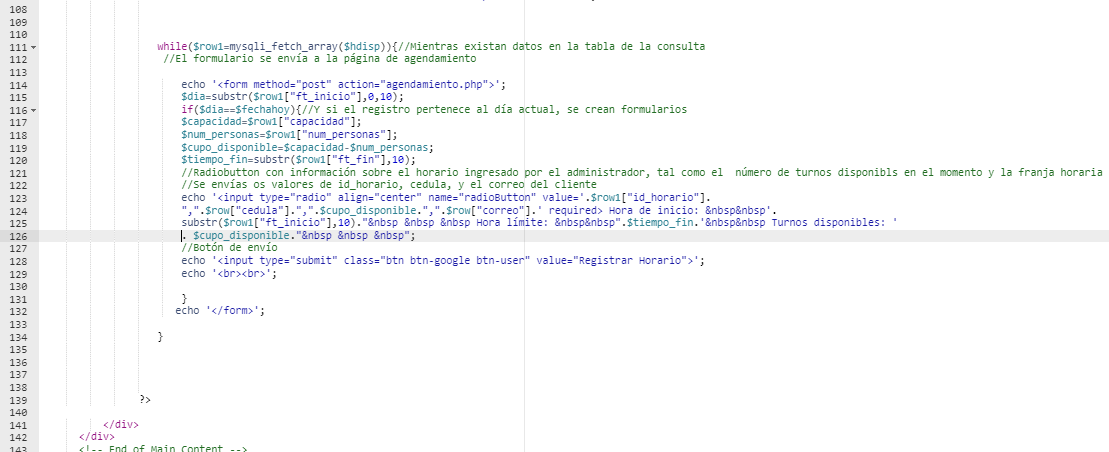


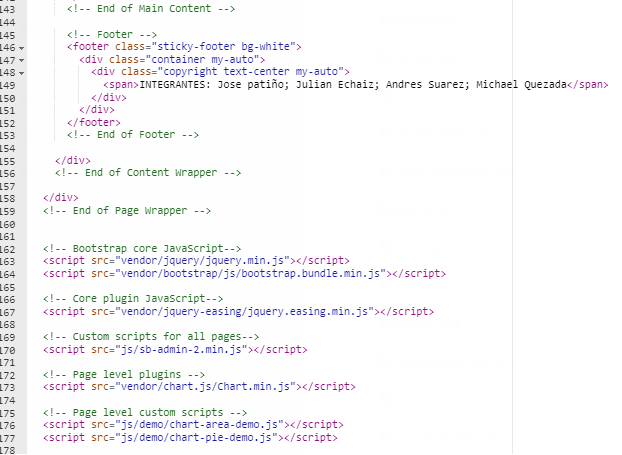
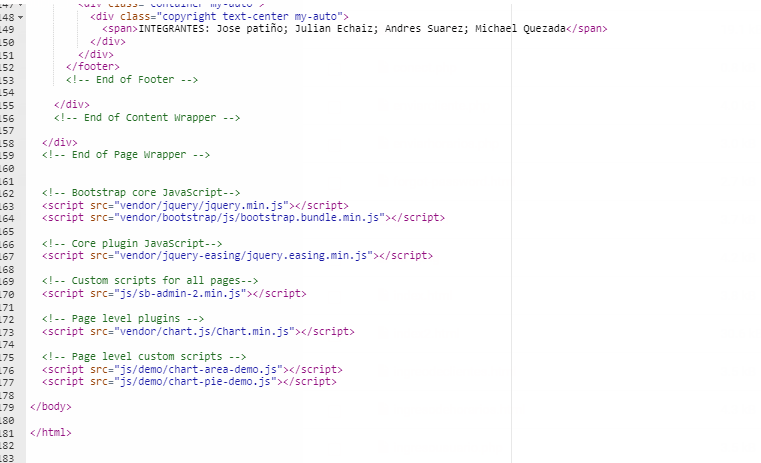
**Archivo userhome.php:** Permite al cliente elegir el horario al cual se registrará y envía los datos (correo, cédula, id\_horario y cupo disponible) al archivo agendamiento.php



****

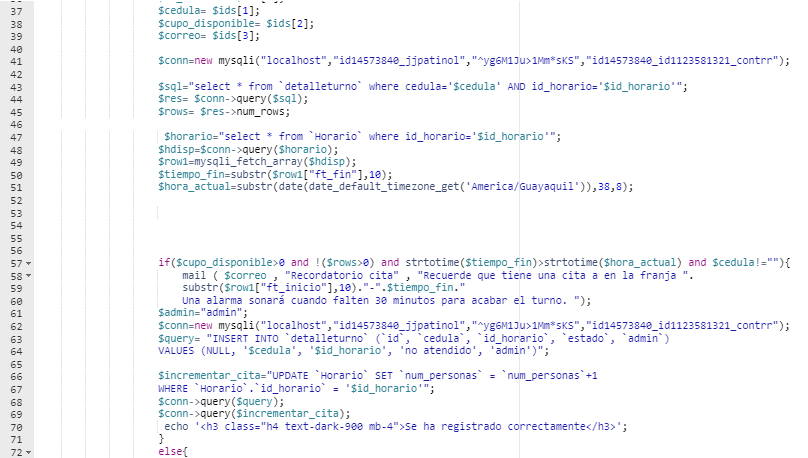
****

****

****

**Archivo agendamiento.php:** Obtiene los datos del formulario del archivo userhome.php, con los cuales se registra el turno en la tabla “detalleturno”, se envía un correo de recordatorio al usuario y se incrementa el número de personas que han reservado un turno en la tabla “Horario”.



****

****

**Archivo userpass.php:** Permite al usuario llenar un formulario para el cambio de su contraseña. Formulario con los campos: Contraseña actual, nueva contraseña y confirmar contraseña. Además, envía este formulario al archivo cambioconf.php para que registre el cambio en a base de datos.





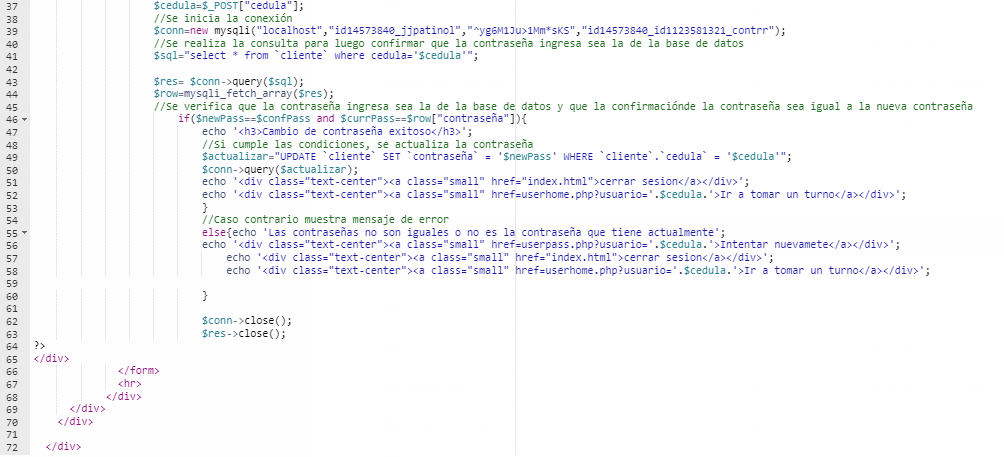






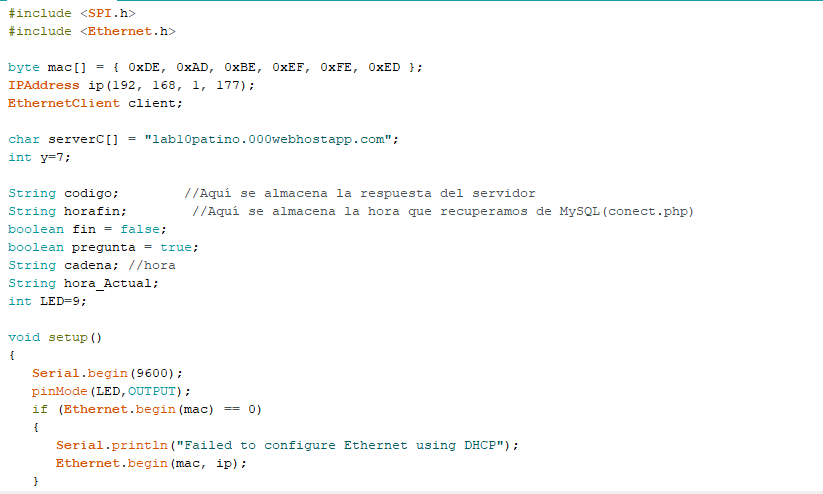
**Archivo cambioconf.php:** Envía la nueva contraseña a la base de datos, si se han cumplido las validaciones, caso contrario muestra un mensaje que indica que el cambio de contraseña ha fallado.

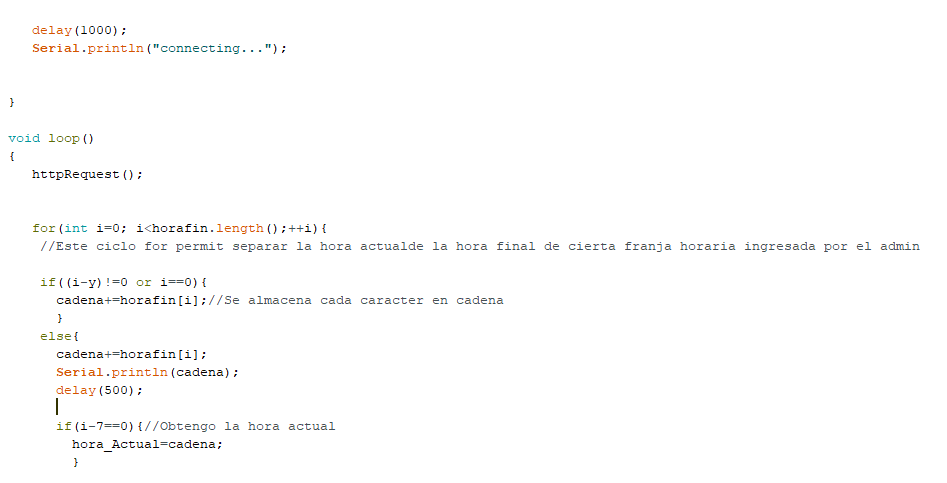






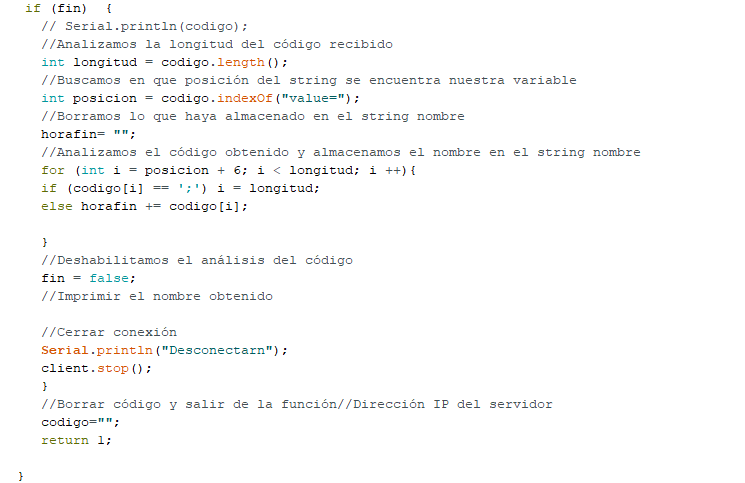
**Código Arduino:** Obtiene a hora del archivo conect.php y la hora 30 minutos antes de que finalice la hora límite de la franja horario ingresada por el administrador mediante la función httpRequest().

****

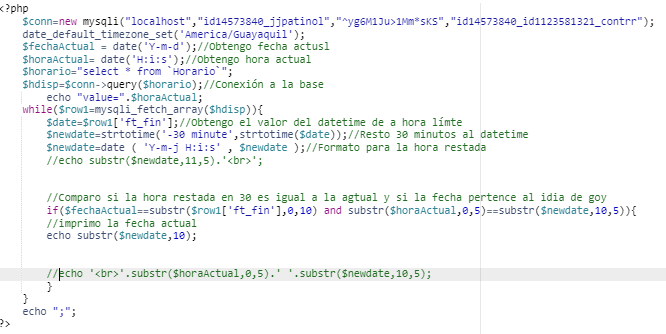
****

****

****

****

**Archivo conect.php:** Obtiene la hora actual y realiza la consulta a la base de datos sobre los horarios límite que ha ingresado el administrador, les resta 30 minutos. Este archivo sirv de intermediario entre la base de datos y la placa Arduino.



# Análisis de presupuesto

Debido a que para el manejo de información se usó una base de datos a través de una página web gratuita 000webhost nuestro presupuesto se limita al dispositivo de sonido que usa el administrador.

|  |  |
| --- | --- |
| Componente | Precio ($) |
| Arduino Ethernet Shield | 17.50 |
| Cable de red | 2.00 |
| Buzzer | 0.50 |
| Microcontrolador Arduino UNO | 12.00 |
| Total | 32.00 |

Incluso, se podría realizar optimizar el sistema si en lugar de usar el Arduino Ethernet Shield y el Microcontrolador Arduino UNO, se usa simplemente la placa Arduino Ethernet. El costo sería aproximadamente de $22.5, tomando en cuenta esta consideración.

# Conclusiones

* Este proyecto cumple con el propósito del control de la clientela del establecimiento el cual ayuda al administrador de local cumplir las condiciones actuales del uso del local.
* El proyecto es rentable ya que su único requerimiento de presupuesto es del dispositivo de sonido empleado.
* Diseño de la aplicación web permite el manejo de información de manera dinámica por medio de una base de datos.
* El sistema es una alternativa para el control del distanciamiento social, a través de la restricción de un máximo de personas en un sitio.

# Referencias bibliográficas

* Cobo, Á. (2005). PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. Ediciones Díaz de Santos.
* WIRE, B. (2020). Andromeda Launches Queue Saver, a Social Distancing Solution App. Retrieved 24 July 2020, from https://www.businesswire.com/news/home/20200409005226/en/Andromeda-Launches-Queue-Saver-Social-Distancing-Solution
* Coronavirus: Crowdless app offers shoppers supermarket crowd levels. (2020, April 27). Retrieved July 27, 2020, from https://www.bbc.com/news/uk-england-oxfordshire-52446658
* Louis, L. (2016). working principle of Arduino and u sing it. International Journal of Control, Automation, Communication and Systems (IJCACS), 1(2), 21-29.
* <https://startbootstrap.com/previews/sb-admin-2/>

**Link del video:**

* <https://espolec-my.sharepoint.com/personal/jechaiz_espol_edu_ec/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fjechaiz%5Fespol%5Fedu%5Fec%2FDocuments%2Fpf2%2Emp4&parent=%2Fpersonal%2Fjechaiz%5Fespol%5Fedu%5Fec%2FDocuments&originalPath=aHR0cHM6Ly9lc3BvbGVjLW15LnNoYXJlcG9pbnQuY29tLzp2Oi9nL3BlcnNvbmFsL2plY2hhaXpfZXNwb2xfZWR1X2VjL0VSVmhXbnVUN0E5QW42dGM0dUU4Rk80Qjd4a19PbVU2SVFRMmlCMmREcHNNTlE_cnRpbWU9RjhvN3gtWlEyRWc>
* https://www.youtube.com/watch?v=TRr2D5oW1no