UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

MISIÓN TIC 2022

DESARROLLO DE SOFTWARE

PROYECTO SCRUM GRUPO G6

ANDRES FELIPE ARDILA LORA

MARIA FERNANDA MUÑOZ TORRES

ANDRIUN ESNAIDER SANTACRUZ NANDAR

DANIELA SANTAMARIA UNATES

JOHANN LEONARDO CASTIBLANCO BERNAL

ANDREA MARCELA FERNÁNDEZ VARELA

MARTHA YOLANDA DÍAZ SÁNCHEZ

BOGOTÁ, SEPTIEMBRE 2021

***Sprint Planning***

**Fecha:** 12/09/21

**Horario:** 11:00 am a 12:30 pm

**Recursos:** El link conexión es: meet.google.com/nag-xwiq-gts

**Participantes:**

| **Integrantes** | **Roll** |
| --- | --- |
| Andres Felipe Ardila | Development team |
| Daniela Santamaria | Development team |
| Andriun Santacruz | Development team |
| Johann Castiblanco | Development team |
| Maria Fernanda Muñoz | Development team |
| Andrea Fernández | Scrum Master |

***Descripción de proyecto.***

En la actualidad muchas personas cuentan con un tiempo limitado debido al alto volumen de trabajo, no se cuenta con el tiempo suficiente para poder realizar agendamientos para actividades de salud, ocio, recreación entre otros.

La revolución tecnológica ha sido un gran apoyo para todas las personas y han logrado facilitar la vida en todos los sentidos, por ello se han desarrollado una serie de herramientas web que permiten conectar e informar a los usuarios de cierta plataforma. Al ver la problemática debido al tema de la pandemia y la carencia en disponibilidad de las personas se requiere implementar un sistema de información que le permita realizar a cada cliente, agendamientos en el centro estético **The Beauty Concept**, conocer sus servicios sin necesidad de acercarse al lugar y de manera virtual obtener la facturación de los servicios solicitados antes de llegar al centro estético; este sistema permitirá renovación y reactivación en el sector. Cada cliente podrá verificar que servicio requiere, los horarios de atención y los precios de cada servicio. De igual manera el dueño de The Beauty Concept podrá administrar desde su usuario los servicios prestados, validar los agendamientos y las facturaciones por cliente, y a su vez listar sus clientes para realizar un seguimiento de cada uno.

De cada cliente se requiere la siguiente información:

Nombres completos, dirección, teléfono, e-mail, número de cuenta bancaria

De la empresa **The Beauty Concept** se requiere la siguiente información:

NIT, dirección, teléfono, e-mail, número de cuenta, servicios prestados

Se requiere que el cliente se registre con todos sus datos personales, anteriormente presentados. Se debe tener la opción de modificar estos datos personales si en algún momento el cliente requiere realizar algún ajuste. Adicional se debe presentar al cliente un listado con los servicios prestados por el centro estético y los precios de cada servicio.

En cuanto a la sección del empresario, el gerente de The Beauty Concept debe diligenciar el formulario con los datos mencionados anteriormente, se debe poder tener la opción de modificar dirección, teléfono, e-mail, número de cuenta y servicios prestados. No será posible cambiar el nombre de la empresa ni el NIT.

A través de la plataforma la empresa podrá revisar los clientes que realizaron su agendamiento respectivo y el tipo de servicio que solicitaron, adicional pueden validar las facturaciones por cliente para realizar su respectivo seguimiento de cartera.

Para entrar a la plataforma se tendrá un login para usuarios y para la empresa.

***Sprint 1***

Para el desarrollo del proyecto se realizará en 5 ***Sprint***. El primer ***Sprint*** que tiene una duración de 1 semana se desarrollará de la siguiente manera:

|  |
| --- |
|  | ***No. Historias de Usuario*** | ***Nombre de la Historia de Usuario*** | ***Como <tipo de usuario>*** | ***Quiero <realizar alguna tarea>*** | ***Para <completar mis tareas a tiempo>*** | ***Criterios de Aceptación*** |  |  |  |  |  |
| Sprint 1 | SIA-01 | ***Diagrama de casos de uso*** | Ingeniero de soporte | Desarrollar las relaciones entre los componentes del sistema | Debe tener representada la lógica interna del sistema | \*Documentación de los componentes \*Se debe especificar las relaciones entre componentes \*Un componente puede tener más de una relación |
| SIA-02 | ***Modelo de datos*** | Ingeniero de soporte | Modelar la estructura de la base de datos del sistema | Debe tener organizada la información, haciendo optima la asignación y la petición de datos | \*modelo Entidad relación o  \* modelo Relacional  y documentación  \*Cada tabla debe tener su respectivo identificador, el cual debe ser único. |
| SIA-03 | ***Implementación de Base de Datos*** | Ingeniero de soporte | Realizar la sincronización del modelo de la base de datos con el sistema | Debe complementar el correcto funcionamiento del sistema | \*Documentación de la BD \*La base de datos tiene que estar montada en algún gestor de bases de datos |
| SIA-04 | ***Diagrama Navegacional*** | Ingeniero de soporte | Diseñar la estructura de navegación entre interfaces del sistema | Debe tener representada la lógica navegacional del sistema | \*Se debe especificar qué acciones se pueden ejecutar en cada interfaz \*Se mostrarán atributos de las interfaces (cual será el home, y cual necesita autentificación) \*Se debe presentar de manera jerárquica. |

***Daily Meeting***

**Fecha:** 13/09/21

**Horario:** 08:00 pm a 08:15 pm

**Recursos:** meet.google.com/ose-zixp-ndc

**Participantes:**

| **Integrantes** | **Roll** |
| --- | --- |
| Andres Felipe Ardila | Development team |
| Daniela Santamaria | Development team |
| Andriun Santacruz | Development team |
| Johann Castiblanco | Development team |
| Maria Fernanda Muñoz | Development team |
| Andrea Fernández | Scrum Master |

***Descripción.***

Se lleva a cabo la reunión a través de la plataforma meet en el enlace relacionado, donde se distribuyen cada una de las actividades a trabajar quedando de la siguiente manera:

1. Descripción del proyecto: Johann Castiblanco

2. SIA-01: Andrea Fernández

3. SIA-02: Andriun Santacruz

4. SIA-03: Andriun Santacruz

5. SIA-04: Daniela Santamaria

Al evidenciar que se cuenta con todas las actividades correspondientes al sprint cubiertas con un encargado se sugiere ir avanzando en el sprint #2 para contar con avances al momento de su inicio.

**Fecha:** 15/09/21

**Horario:** 08:00 pm a 08:15 pm

**Recursos:** meet.google.com/ose-zixp-ndc

**Participantes:**

| **Integrantes** | **Roll** |
| --- | --- |
| Andres Felipe Ardila | Development team |
| Daniela Santamaria | Development team |
| Andriun Santacruz | Development team |
| Johann Castiblanco | Development team |
| Maria Fernanda Muñoz | Development team |
| Andrea Fernández | Scrum Master |

***Descripción.***

En este día se revisan los avances obtenidos en el desarrollo de las actividades asignadas a cada integrante. En el inicio de la sesión se hace la aclaración que la descripción del proyecto debe ajustarse a un solo salón de belleza y no dejarlo abierto para evitar malinterpretaciones del objetivo en desarrollo; seguido se revisa el mapa navegacional donde se entra en consenso si se llevara a cabo el desarrollo del Login de Usuario para que cada cliente cuente con su propia cuenta y le sea posible acceder al historial de sesiones en caso de requerirse, llegando a la conclusión que si es necesario incluirla por temas de seguridad en la información, con base en el login dependerá la interfaz a mostrar si es de cliente y/o de propietario del negocio.

Finalmente se verifica el diagrama de entidad relación y se da el VoBo del grupo sugiriendo ajustes mínimos alineado al punto anteriormente mencionado, adicional se evalúa la posibilidad de tener también el diagrama relacional teniendo en cuenta la disponibilidad de los compañeros con actividades aún pendientes por asignar, Andres Felipe Ardila se postula como persona al cargo.

**Fecha:** 17/09/21

**Horario:** 08:00 pm a 08:15 pm

**Recursos:** meet.google.com/ose-zixp-ndc

**Participantes:**

| **Integrantes** | **Roll** |
| --- | --- |
| Andres Felipe Ardila | Development team |
| Daniela Santamaria | Development team |
| Andriun Santacruz | Development team |
| Johann Castiblanco | Development team |
| Maria Fernanda Muñoz | Development team |
| Andrea Fernández | Scrum Master |

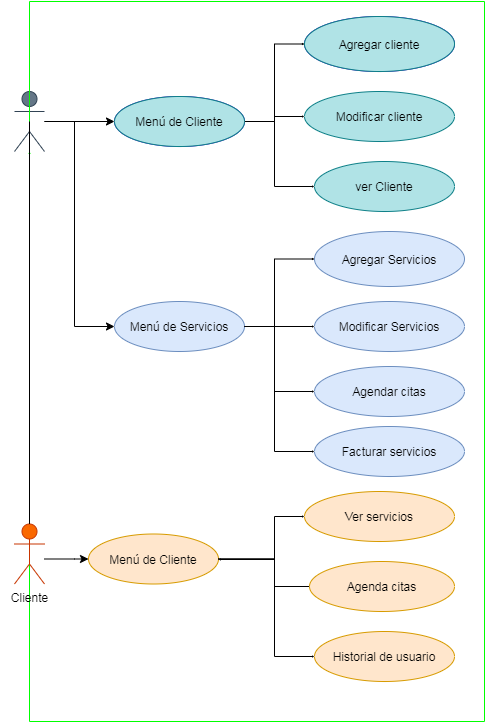
***Descripción.***

Se pide colocar mínimo 3 párrafos donde evidencien el avance por cada número de historia de usuario. Si hay novedades positivas o negativas y cómo se dio solución.

***Product Increment***

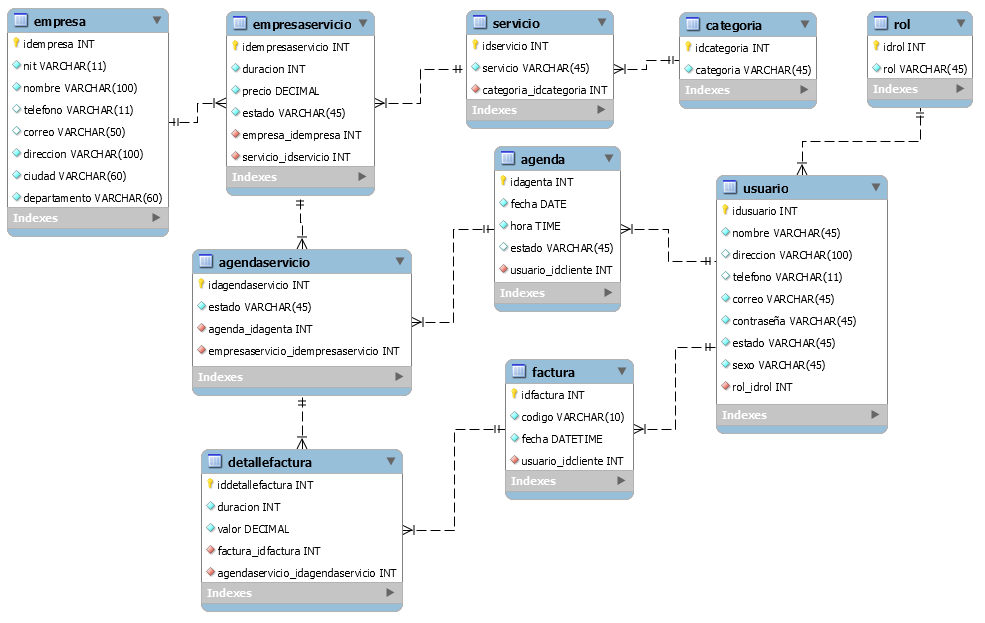
**SIA-1**

El diagrama de casos de uso fue desarrollado en el aplicativo draw.io con la versión en línea sincronizado directamente a Google Drive.



**SIA-2**

El diagrama Entidad - Relación, se desarrolló con el gestor de base de datos MySQL Workbeach, versión 8.0, se anexa el fichero SIA-2 Diagrama entidad-relación.mwb



**SIA-3**

El código fue creado al exportar la BD utilizando MySQL Workbench, versión 8.0, se anexa el fichero SIA-3 ProyectoC3.sql

-- MySQL Script generated by MySQL Workbench

-- Thu Sep 16 23:51:13 2021

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema proyectoC3

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `proyectoC3` DEFAULT CHARACTER SET utf8 ;

USE `proyectoC3` ;

-- -----------------------------------------------------

-- Table `proyectoC3`.`empresa`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`empresa` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`empresa` (

`idempresa` INT NOT NULL AUTO\_INCREMENT,

`nit` VARCHAR(11) NOT NULL,

`nombre` VARCHAR(100) NOT NULL,

`telefono` VARCHAR(11) NULL,

`correo` VARCHAR(50) NULL,

`direccion` VARCHAR(100) NOT NULL,

`ciudad` VARCHAR(60) NOT NULL,

`departamento` VARCHAR(60) NOT NULL,

PRIMARY KEY (`idempresa`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`rol`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`rol` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`rol` (

`idrol` INT NOT NULL,

`rol` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idrol`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`usuario`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`usuario` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`usuario` (

`idusuario` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(45) NOT NULL,

`direccion` VARCHAR(100) NULL,

`telefono` VARCHAR(11) NULL,

`correo` VARCHAR(45) NOT NULL,

`contraseña` VARCHAR(45) NOT NULL,

`estado` VARCHAR(45) NOT NULL,

`sexo` VARCHAR(45) NOT NULL,

`rol\_idrol` INT NOT NULL,

PRIMARY KEY (`idusuario`),

INDEX `fk\_usuario\_rol1\_idx` (`rol\_idrol` ASC) VISIBLE,

CONSTRAINT `fk\_usuario\_rol1`

FOREIGN KEY (`rol\_idrol`)

REFERENCES `proyectoC3`.`rol` (`idrol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`categoria`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`categoria` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`categoria` (

`idcategoria` INT NOT NULL AUTO\_INCREMENT,

`categoria` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idcategoria`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`servicio`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`servicio` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`servicio` (

`idservicio` INT NOT NULL AUTO\_INCREMENT,

`servicio` VARCHAR(45) NOT NULL,

`categoria\_idcategoria` INT NOT NULL,

PRIMARY KEY (`idservicio`),

INDEX `fk\_servicio\_categoria1\_idx` (`categoria\_idcategoria` ASC) VISIBLE,

CONSTRAINT `fk\_servicio\_categoria1`

FOREIGN KEY (`categoria\_idcategoria`)

REFERENCES `proyectoC3`.`categoria` (`idcategoria`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`empresaservicio`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`empresaservicio` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`empresaservicio` (

`idempresaservicio` INT NOT NULL AUTO\_INCREMENT,

`duracion` INT NOT NULL DEFAULT 0,

`precio` DECIMAL NOT NULL DEFAULT 0,

`estado` VARCHAR(45) NOT NULL,

`empresa\_idempresa` INT NOT NULL,

`servicio\_idservicio` INT NOT NULL,

PRIMARY KEY (`idempresaservicio`),

INDEX `fk\_empresaservicios\_empresa1\_idx` (`empresa\_idempresa` ASC) VISIBLE,

INDEX `fk\_empresaservicios\_servicio1\_idx` (`servicio\_idservicio` ASC) VISIBLE,

CONSTRAINT `fk\_empresaservicios\_empresa1`

FOREIGN KEY (`empresa\_idempresa`)

REFERENCES `proyectoC3`.`empresa` (`idempresa`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_empresaservicios\_servicio1`

FOREIGN KEY (`servicio\_idservicio`)

REFERENCES `proyectoC3`.`servicio` (`idservicio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`agenda`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`agenda` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`agenda` (

`idagenta` INT NOT NULL AUTO\_INCREMENT,

`fecha` DATE NOT NULL,

`hora` TIME NOT NULL,

`estado` VARCHAR(45) NULL,

`usuario\_idcliente` INT NOT NULL,

PRIMARY KEY (`idagenta`),

INDEX `fk\_agenta\_usuario1\_idx` (`usuario\_idcliente` ASC) VISIBLE,

CONSTRAINT `fk\_agenta\_usuario1`

FOREIGN KEY (`usuario\_idcliente`)

REFERENCES `proyectoC3`.`usuario` (`idusuario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`agendaservicio`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`agendaservicio` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`agendaservicio` (

`idagendaservicio` INT NOT NULL AUTO\_INCREMENT,

`estado` VARCHAR(45) NOT NULL,

`agenda\_idagenta` INT NOT NULL,

`empresaservicio\_idempresaservicio` INT NOT NULL,

PRIMARY KEY (`idagendaservicio`),

INDEX `fk\_agendaservicio\_agenta1\_idx` (`agenda\_idagenta` ASC) VISIBLE,

INDEX `fk\_agendaservicio\_empresaservicio1\_idx` (`empresaservicio\_idempresaservicio` ASC) VISIBLE,

CONSTRAINT `fk\_agendaservicio\_agenta1`

FOREIGN KEY (`agenda\_idagenta`)

REFERENCES `proyectoC3`.`agenda` (`idagenta`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_agendaservicio\_empresaservicio1`

FOREIGN KEY (`empresaservicio\_idempresaservicio`)

REFERENCES `proyectoC3`.`empresaservicio` (`idempresaservicio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`factura`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`factura` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`factura` (

`idfactura` INT NOT NULL,

`codigo` VARCHAR(10) NOT NULL,

`fecha` DATETIME NOT NULL,

`usuario\_idcliente` INT NOT NULL,

PRIMARY KEY (`idfactura`),

INDEX `fk\_factura\_usuario1\_idx` (`usuario\_idcliente` ASC) VISIBLE,

CONSTRAINT `fk\_factura\_usuario1`

FOREIGN KEY (`usuario\_idcliente`)

REFERENCES `proyectoC3`.`usuario` (`idusuario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `proyectoC3`.`detallefactura`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `proyectoC3`.`detallefactura` ;

CREATE TABLE IF NOT EXISTS `proyectoC3`.`detallefactura` (

`iddetallefactura` INT NOT NULL AUTO\_INCREMENT,

`duracion` INT NOT NULL,

`valor` DECIMAL NOT NULL,

`factura\_idfactura` INT NOT NULL,

`agendaservicio\_idagendaservicio` INT NOT NULL,

PRIMARY KEY (`iddetallefactura`),

INDEX `fk\_detallefactura\_factura1\_idx` (`factura\_idfactura` ASC) VISIBLE,

INDEX `fk\_detallefactura\_agendaservicio1\_idx` (`agendaservicio\_idagendaservicio` ASC) VISIBLE,

CONSTRAINT `fk\_detallefactura\_factura1`

FOREIGN KEY (`factura\_idfactura`)

REFERENCES `proyectoC3`.`factura` (`idfactura`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_detallefactura\_agendaservicio1`

FOREIGN KEY (`agendaservicio\_idagendaservicio`)

REFERENCES `proyectoC3`.`agendaservicio` (`idagendaservicio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

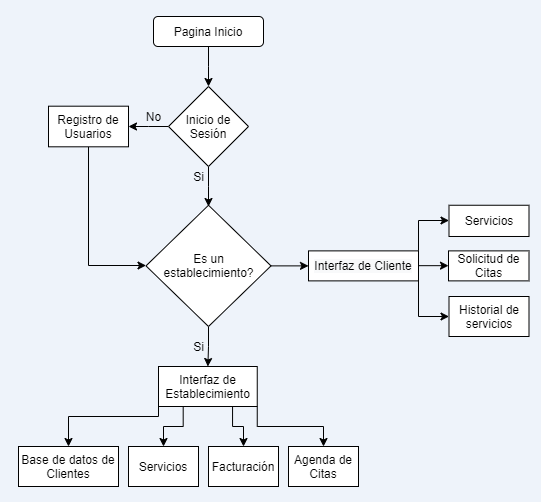
ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**SIA-4**



Para el desarrollo del mapa navegacional se utilizó el aplicativo draw.io

**Descripción:**

En la página de inicio encontraremos la sección del login donde el cliente podrá iniciar sesión o en su defecto podrá registrarse. Durante el registro los usuarios nuevos definen el rol de los mismos como establecimiento o cliente.

Si el usuario logra iniciar sesión será llevado a una interfaz que dependerá de su rol de usuario. Si es un establecimiento encontrará un menú de navegación con las siguiente opciones:

* Base de Datos de Clientes
* Base de Servicios (Detalles y precios)
* Sistema de Facturación
* Agenda

Si el usuario es un cliente será enviado a una interfaz donde encontrará la información general y promocional del establecimiento al igual que un menú con la siguientes opciones:

* Servicios
* Agenda de citas
* Historial de servicios