

Descripción del Proyecto

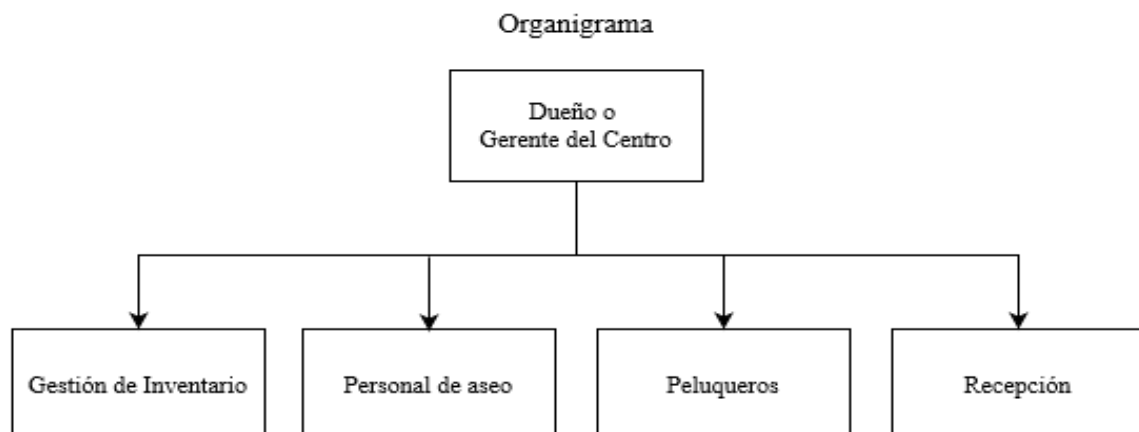
La empresa de peluquería **Arte en el Cabello** ha solicitado el desarrollo de un proyecto buscando la digitalización y fácil control de su información. El interés de la empresa es el poder proporcionar un sistema de reserva de citas en línea a través de una aplicación web para que los clientes puedan reservar o cancelar citas, ver la disponibilidad de los peluqueros y poder llevar un registro de los productos existentes.

También se busca que el mismo software permita tener un sistema de gestión de inventario, que ayudaría a controlar los suministros y productos de la peluquería y a generar un informe del inventario y las ventas.

Arte en el Cabello es un lugar donde se ofrecen servicios relacionados con el cuidado del cabello y la apariencia en general. Algunos de los servicios que se pueden encontrar son el corte del cabello, peinados, coloraciones. En general, se enfoca en ofrecer servicios que mejoren la apariencia y el bienestar de sus clientes.

La empresa cuenta actualmente con varias plantillas hechas en Excel para el manejo de la información, la gestión de las citas y los recursos. Esto hace que conforme aumenta la clientela se hace más complicado el poder consultar o modificar las plantillas. Esto hace que su uso sea limitado debido a que obliga al usuario a estar frente al ordenador para poder usarlo.

La empresa cuenta con una única sede debido al poco tiempo que lleva en el sector. También el personal con el que dispone es el mínimo para su correcto funcionamiento.



Los cargos y funciones se describen a continuación.

Dueño o Gerente General: Es la persona encargada de liderar el equipo de trabajo y asegurarse de que el negocio funcione de manera eficiente y rentable. Entre sus principales responsabilidades se encuentran la planificación y organización del trabajo de los empleados, la gestión de las finanzas y el presupuesto del negocio, la toma de decisiones estratégicas y la implementación de estrategias para mejorar la calidad de los servicios y aumentar la clientela.

Recepcionista: Es la persona encargada de atender a los clientes que entran al establecimiento, lo que incluye recibirlos, hacerles preguntas para determinar sus necesidades y brindarles información sobre los servicios que ofrece la peluquería. También es responsable de hacer citas para los clientes y organizar el horario de los peluqueros. En general, la recepcionista es la primera persona con la que se encuentran los clientes al entrar a la peluquería, por lo que su trabajo es esencial para brindar una buena atención al cliente y crear una buena impresión.

Personal de Aseo: es responsable de mantener el establecimiento limpio y en buenas condiciones. Sus funciones incluyen tareas como barrer y fregar el piso, desinfectar y limpiar las mesas, sillas y otros equipos, vaciar la basura y realizar tareas de mantenimiento en el establecimiento.

Gestion del Inventario: Se encarga de asegurar que se cuenten con los productos y herramientas necesarios para brindar los servicios de calidad que ofrece la peluquería. Esto incluye llevar un registro de los productos que se utilizan en la peluquería, controlar el nivel de existencias y hacer pedidos de reposición cuando sea necesario.

Peluqueros: Su función principal es la de brindar servicios de corte, peinado, coloración y otros tratamientos capilares a los clientes de la peluquería. Los peluqueros suelen contar con una formación especializada y están altamente capacitados en el uso de técnicas y herramientas especializadas para el cuidado del cabello.

Los recursos de hardware y software que tiene la empresa para mejorar la eficiencia y la productividad en el negocio se detallan a continuación:

- **Computadora:** Son uno de los recursos informáticos más importantes en la peluquería, ya que se utiliza para llevar a cabo una gran cantidad de tareas, desde el registro de clientes y la gestión de citas hasta la facturación y el control de finanzas.
- **Impresora:** Son otro recurso informático esencial en la peluquería, ya que permiten imprimir documentos importantes como facturas, recibos y otros documentos necesarios para llevar a cabo las operaciones diarias de la empresa.
- **Software:** En este caso se utilizan varias plantillas hechas en Excel. Esto permite llevar a cabo tareas específicas de gestión de la empresa. Esto puede incluir la gestión de citas, registro de clientes, facturación, finanzas y control del inventario.
- **Herramientas de comunicación y colaboración:** En una peluquería, es importante mantenerse en contacto con los clientes y el personal, y para ello se utiliza el correo electrónico y las aplicaciones de mensajería instantánea. Esto permite a los empleados mantenerse en contacto y coordinarse entre sí para llevar a cabo las operaciones de la empresa de manera eficiente.
- **La caja registradora:** Se utiliza para llevar un registro de las ventas y hacer cálculos de cambio.
- **Un sistema de seguridad y vigilancia:** Incluye cámaras de vigilancia y un sistema de alarma para proteger contra robos y otras amenazas.
- **Una conexión a Internet:** Para permitir el uso de aplicaciones en línea y el acceso a información relevante para el negocio.

El equipo de software y hardware en la peluquería también puede ayudar a mejorar la imagen del negocio y aumentar la satisfacción de los clientes. Un sistema de gestión de citas y una aplicación de registro de clientes pueden proporcionar una experiencia más profesional y organizada, mientras que el uso de herramientas y dispositivos de alta calidad puede demostrar que la peluquería se preocupa por ofrecer un servicio de alta calidad. Además, también pueden mejorar la experiencia del cliente y hacer que se sienta cómodo y satisfecho durante su visita a la peluquería.

Requisitos funcionales y no funcionales

Los requisitos funcionales de un sistema para una peluquería pueden incluir:

Registro de clientes: una base de datos que almacene información sobre los clientes, como nombre, dirección, número de teléfono y preferencias de corte de cabello.

Reservas: la capacidad de hacer reservas para servicios de corte de cabello y otros tratamientos, así como la posibilidad de cancelar o modificar reservas.

Programación de citas: un calendario que muestre los horarios disponibles y permita a los clientes elegir una hora para su cita.

Cobro: la capacidad de cobrar por los servicios prestados, incluyendo el manejo de pagos en efectivo, tarjetas de crédito y débito y otros medios de pago.

Registro de ventas: un registro de las ventas realizadas, incluyendo el tipo de servicio prestado y el monto cobrado.

Gestión de inventario: la capacidad de llevar un registro de los productos utilizados y necesarios para los servicios de la peluquería, y la posibilidad de realizar pedidos de reposición de inventario cuando sea necesario.

Reportes: la generación de reportes sobre el rendimiento de la peluquería, incluyendo informes sobre ventas, clientes y productos más populares.

Seguridad: medidas de seguridad para proteger la información de los clientes y garantizar la privacidad de sus datos.

Y los requisitos no funcionales pueden ser:

Disponibilidad: asegurar que el sistema esté disponible para los clientes y el personal de la peluquería en todo momento.

Escalabilidad: la capacidad del sistema para adaptarse y crecer junto con la empresa, sin interrumpir el funcionamiento normal.

Usabilidad: diseño intuitivo y fácil de usar para que el personal de la peluquería pueda utilizar el sistema de manera eficiente.

Rendimiento: un rendimiento rápido y eficiente del sistema para minimizar tiempos de espera y mejorar la experiencia del cliente.

Integración: la capacidad de integrar el sistema con otros sistemas utilizados por la peluquería, como sistemas de pago o de gestión de inventario.

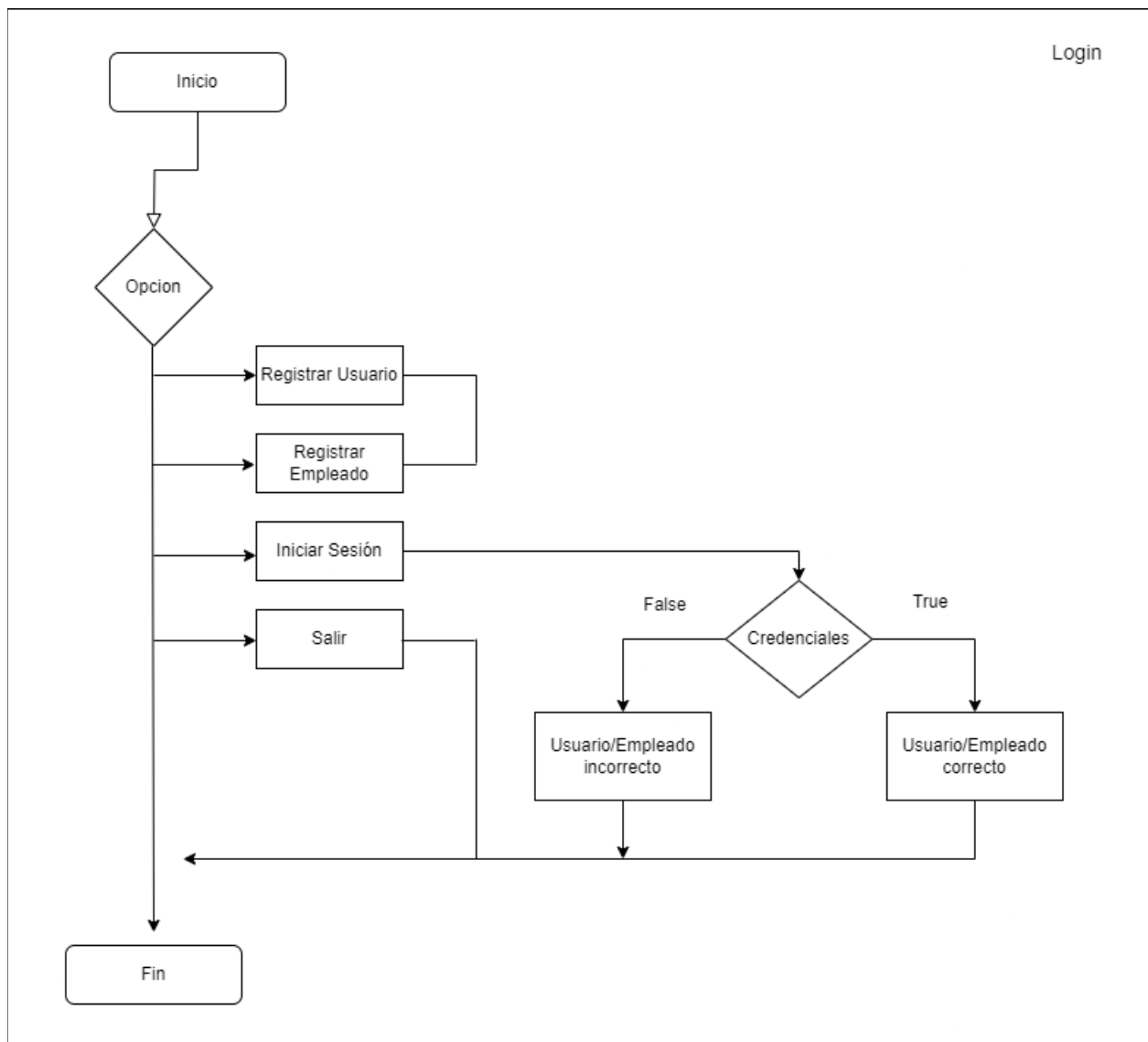
Mantenibilidad: la facilidad de mantenimiento y actualización del sistema para garantizar su correcto funcionamiento a largo plazo.

Seguridad: medidas de seguridad para proteger el sistema de posibles vulnerabilidades y garantizar la confidencialidad de la información.

Cumplimiento: cumplimiento de los estándares y regulaciones aplicables, como la protección de datos personales.

Test

Caja blanca



Caminos básicos

- Camino 1: Inicio -> Opción -> Registro Usuario -> Fin
- Camino 2: Inicio -> Opción -> Registro Empleado -> Fin
- Camino 3: Inicio -> Opción -> Inicio de Sesión -> Credenciales Correctas -> Fin
- Camino 4: Inicio -> Opción -> Inicio de Sesión -> Credenciales Incorrectas -> Fin
- Camino 5: Inicio -> Opción -> Salir -> Fin

Casos de prueba

- Caso de prueba 1: Seleccionar Opción 1, ingresar los datos de registro de un usuario y finalizar el programa.
- Caso de prueba 2: Seleccionar Opción 2, ingresar los datos de registro de un empleado y finalizar el programa.
- Caso de prueba 3: Seleccionar Opción 3, seleccionar Inicio de Sesión, ingresar credenciales correctas y finalizar el programa.
- Caso de prueba 4: Seleccionar Opción 3, seleccionar Inicio de Sesión, ingresar credenciales incorrectas y finalizar el programa.
- Caso de prueba 5: Seleccionar Opción 4, seleccionar Salir y finalizar el programa.

Caja negra

```
1 import org.junit.After;
2 import org.junit.AfterClass;
3 import org.junit.Before;
4 import org.junit.BeforeClass;
5 import org.junit.runner.RunWith;
6 import org.junit.runners.Suite;
7
8 @RunWith(Suite.class)
9 @Suite.SuiteClasses({
10     com.mycompany.login.UsuarioIT.class, com.mycompany.login.EmpleadoIT.class, com.mycompany.login.LoginAppIT.class, com.mycompany.login.ArchivoRegistroIT.class
11 })
12 public class NewTestSuite {
13
14     @BeforeClass
15     public static void setUpClass() throws Exception {
16     }
17
18     @AfterClass
19     public static void tearDownClass() throws Exception {
20     }
21
22     @Before
23     public void setUp() throws Exception {
24     }
25
26     @After
27     public void tearDown() throws Exception {
28     }
29 }
30
31
```



```
1  import org.junit.jupiter.api.AfterEach;
2  import org.junit.jupiter.api.AfterAll;
3  import org.junit.jupiter.api.BeforeEach;
4  import org.junit.jupiter.api.BeforeAll;
5  import org.junit.jupiter.api.Test;
6  import static org.junit.jupiter.api.Assertions.*;
7
8  public class LoginAppIT {
9
10     public LoginAppIT() {
11     }
12
13     @BeforeAll
14     public static void setUpClass() {
15     }
16
17     @AfterAll
18     public static void tearDownClass() {
19     }
20
21     @BeforeEach
22     public void setUp() {
23     }
24
25     @AfterEach
26     public void tearDown() {
27     }
28
29     @Test
30     public void testMain() {
31         System.out.println("main");
32         String[] args = null;
33         LoginApp.main(args);
34     }
35 }
36
37
```

```
1  import org.junit.jupiter.api.AfterEach;
2  import org.junit.jupiter.api.AfterAll;
3  import org.junit.jupiter.api.BeforeEach;
4  import org.junit.jupiter.api.BeforeAll;
5  import org.junit.jupiter.api.Test;
6  import static org.junit.jupiter.api.Assertions.*;
7
8  public class EmpleadoIT {
9
10     public EmpleadoIT() {
11     }
12
13     @BeforeAll
14     public static void setUpClass() {
15     }
16
17     @AfterAll
18     public static void tearDownClass() {
19     }
20
21     @BeforeEach
22     public void setUp() {
23     }
24
25     @AfterEach
26     public void tearDown() {
27     }
28
29     @Test
30     public void testGetNombreEmpleado() {
31         System.out.println("getNombreEmpleado");
32         Empleado instance = null;
33         String expectedResult = "";
34         String result = instance.getNombreEmpleado();
35         assertEquals(expectedResult, result);
36     }
37
38     @Test
39     public void testGetCargoEmpleado() {
40         System.out.println("getCargoEmpleado");
41         Empleado instance = null;
42         String expectedResult = "";
43         String result = instance.getCargoEmpleado();
44         assertEquals(expectedResult, result);
45     }
46 }
47
```

```
1  import org.junit.jupiter.api.AfterEach;
2  import org.junit.jupiter.api.AfterAll;
3  import org.junit.jupiter.api.BeforeEach;
4  import org.junit.jupiter.api.BeforeAll;
5  import org.junit.jupiter.api.Test;
6  import static org.junit.jupiter.api.Assertions.*;
7
8
9  public class UsuarioIT {
10
11      public UsuarioIT() {
12      }
13
14      @BeforeAll
15      public static void setUpClass() {
16      }
17
18      @AfterAll
19      public static void tearDownClass() {
20      }
21
22      @BeforeEach
23      public void setUp() {
24      }
25
26      @AfterEach
27      public void tearDown() {
28      }
29
30      @Test
31      public void testGetNombreUsuario() {
32          System.out.println("getNombreUsuario");
33          Usuario instance = null;
34          String expectedResult = "";
35          String result = instance.getNombreUsuario();
36          assertEquals(expectedResult, result);
37      }
38
39
40      @Test
41      public void testGetContraseña() {
42          System.out.println("getContraseña");
43          Usuario instance = null;
44          String expectedResult = "";
45          String result = instance.getContraseña();
46          assertEquals(expectedResult, result);
47      }
48  }
49 }
```



```
1  import org.junit.jupiter.api.AfterEach;
2  import org.junit.jupiter.api.AfterAll;
3  import org.junit.jupiter.api.BeforeEach;
4  import org.junit.jupiter.api.BeforeAll;
5  import org.junit.jupiter.api.Test;
6  import static org.junit.jupiter.api.Assertions.*;
7
8  public class ArchivoRegistroIT {
9
10     public ArchivoRegistroIT() {
11     }
12
13     @BeforeAll
14     public static void setUpClass() {
15     }
16
17     @AfterAll
18     public static void tearDownClass() {
19     }
20
21     @BeforeEach
22     public void setUp() {
23     }
24
25     @AfterEach
26     public void tearDown() {
27     }
28
29     @Test
30     public void testGuardarRegistro() {
31         System.out.println("guardarRegistro");
32         Object objeto = null;
33         ArchivoRegistro instance = new ArchivoRegistro();
34         instance.guardarRegistro(objeto);
35     }
36
37     @Test
38     public void testVerificarCredenciales() {
39         System.out.println("verificarCredenciales");
40         String nombreUsuario = "";
41         String contraseña = "";
42         ArchivoRegistro instance = new ArchivoRegistro();
43         boolean expResult = false;
44         boolean result = instance.verificarCredenciales(nombreUsuario, contraseña);
45         assertEquals(expResult, result);
46     }
47 }
48
```

MANUAL DE USUARIO

Arte en el Cabello

Prototipo Web

Carlos López y Erick Mejía

Manual de Usuario para la Página Web de Peluquería

Bienvenido a nuestra página web, en la que podrás acceder a todos los servicios que ofrecemos en nuestra peluquería. Encontrarás las pestañas Nosotros y Contacto para conocer nuestra trayectoria y poder contactar directamente con nosotros. Para utilizar nuestra plataforma, primero tendrás que registrarte o si ya estás registrado, introducir tu nombre de usuario y contraseña.

1. Haz clic en el botón “Registrarse” en la página principal de la web.
2. Rellena el formulario con tus datos personales y crea un nombre de usuario y contraseña seguros.
3. Haz clic en el botón “Registrarse” para completar el proceso.

Inicio de sesión

1. Haz clic en el botón “Iniciar sesión” en la página principal de la web.
2. Introduce tu nombre de usuario y contraseña.
3. Haz clic en el botón “Iniciar sesión” para acceder a tu cuenta.

Reservar Cita

1. Haz clic en el apartado “Reservar Cita”.
2. Selecciona la fecha y hora que prefieras para tu cita.
3. Haz clic en el botón “Reservar Cita” para confirmar tu reserva.

Personal de la peluquería

1. Haz clic en el apartado “Peluqueros”.
2. Conoce a nuestro equipo de profesionales y sus especializaciones.
3. Haz clic en el nombre de un miembro del personal para obtener más información sobre su perfil y servicios.

Servicios

1. Haz clic en el apartado “Servicios”.
2. Descubre todos nuestro menú de servicios: corte, afeitados, técnica, tratamientos...

Cita

1. Haz clic en el apartado “Cita”.
2. En este apartado encontrarás tu nombre, fecha y hora de la cita, servicio contratado y estilista elegido.

Cerrar sesión

1. Haz clic en apartado “Cerrar sesión”.
2. Haz clic en el botón “Cerrar sesión” para cerrar tu sesión y proteger la privacidad de tu información.