



# CONSULTORIO ODONTOLOGICO

integrantes

- Álvarez Ivan
- Herrera Hernan
- Molina Cristina
- Romaya Santino
- Larralde Dolores



INGENIERIA DE SOFTWARE I



## **1. SISTEMA WEB PARA CONSULTORIO ODONTOLÓGICO "ODONTURNO"**

### **Introducción**

Los cambios que se han generado en la sociedad actual se ven afectados por los retos del nuevo siglo. Y es el manejo de la información por medio de la informática, donde la expectativa de apertura hacia la tecnología tiene un lugar.

El consultorio necesita enfrentar la reducción de problemas, no solo en lo estructural sino también en como proyectarse hacia un futuro, donde la creatividad y la innovación sean el empujón de su día a día.

De este modo, el presente proyecto nace con el objetivo de implementar un sistema web capaz de satisfacer las necesidades de un consultorio odontológico, que permita el control de los procesos y brinde tanto a los odontólogos las herramientas adecuadas para poder agilizar la atención a cada paciente. Logrando ahorrar espacio físico en el almacenamiento del historial clínico, optimizando los tiempos para la búsqueda de información, alcanzando un mayor control en el flujo de citas y finalmente ofrecer un servicio de mejor calidad con mayor seguridad y estabilidad de los datos.

Resumen:

### **Tema del Software:**

El software "ODONTURNO" es un sistema web diseñado para abordar los desafíos en la gestión de citas en el campo de la odontología. Su objetivo principal es facilitar a los odontólogos la programación y seguimiento de las citas de los pacientes, mejorando la eficiencia y la experiencia tanto para los profesionales de la salud como para los pacientes.

### **Problema Presentado:**

La gestión manual de citas en consultas odontológicas suele ser un proceso laborioso y propenso a errores. Los odontólogos a menudo enfrentan desafíos al programar citas, realizar recordatorios a los pacientes y gestionar cambios en el horario. Esto puede resultar en la pérdida de tiempo y oportunidades, así como en la insatisfacción de los pacientes debido a posibles errores en las citas o retrasos.

### **Solución Ideada:**

La solución propuesta es el software "ODONTURNO", que automatiza la gestión de citas para odontólogos. Permite a los profesionales de la salud programar citas de manera eficiente, enviar recordatorios automáticos a los pacientes a través de mensajes de texto o correo electrónico, y gestionar su calendario de manera más efectiva. Además, ofrece la posibilidad de realizar un seguimiento de las citas pasadas y futuras, lo que facilita la planificación de la jornada de trabajo.

### **Mejora Esperada con la Implementación del Software Propuesto:**

La implementación de "ODONTURNO" promete una serie de mejoras significativas. En primer lugar, se espera una mayor eficiencia en la gestión de citas, lo que reduce la carga de trabajo administrativo para los odontólogos y sus equipos. Además, al enviar recordatorios automáticos a los pacientes, se reduce el riesgo de cancelaciones de última hora o la falta de asistencia a las citas. Esto no solo optimiza el flujo de trabajo, sino que también mejora la satisfacción del paciente al brindar un servicio más organizado y puntual. En última instancia,

"ODONTURNO" tiene el potencial de aumentar la productividad de la clínica odontológica y mejorar la experiencia tanto para los profesionales como para los pacientes.

## 2. Requisitos del Sistema para un Sistema Web Odontológico:

### Requisitos Funcionales:

- ✓ **Inicio de Sesión Seguro:** El sistema web debe ofrecer un mecanismo de inicio de sesión seguro, que requiera un nombre de usuario y una contraseña válida para garantizar el acceso solo a usuarios autorizados.
- ✓ **Registro y Gestión de Pacientes:** Debe permitir el registro y gestión de información de pacientes, incluyendo datos personales como nombre, DNI, dirección correo electrónico y número de teléfono. Debe ser posible actualizar la información y realizar búsquedas rápidas.
- ✓ **Historias Clínicas:** El sistema debe permitir la creación y acceso a las historias clínicas electrónicas de los pacientes. Esto incluye la posibilidad de registrar información médica relevante como antecedentes médicos, medicamentos recetados, avance de tratamiento, etc. Además, debe permitir la visualización y actualización segura de esta información.
- ✓ **Control de Cantidad de Pacientes Atendidos y Tiempo de Atención:** El sistema debe llevar un registro de la cantidad de pacientes atendidos diariamente y el tiempo dedicado a cada uno. Debe generar informes y estadísticas para evaluar la productividad y el tiempo promedio de atención.
- ✓ **Acceso Seguro y Controlado a la Información:** El sistema debe garantizar la seguridad y confidencialidad de la información mediante mecanismos de autenticación y autorización. Además, debe registrar las acciones realizadas por los usuarios para fines de auditoría.
- ✓ **Recordatorios de Citas:** El sistema debe enviar recordatorios automáticos a los pacientes sobre sus citas y seguimientos médicos.
- ✓ **Interfaz Intuitiva y Fácil de Usar:** El sistema debe contar con una interfaz intuitiva y fácil de usar para médicos y pacientes, permitiendo la realización eficiente de tareas.

### Requisitos No Funcionales:

- ✓ **Seguridad de la Información:** El sistema debe garantizar la seguridad y confidencialidad de la información de los pacientes, cumpliendo con regulaciones de privacidad y protección de datos.

- ✓ **Disponibilidad del Sistema:** El sistema debe estar disponible en todo momento, evitando interrupciones en el servicio mediante una infraestructura confiable y redundante.
- ✓ **Rendimiento y Velocidad de Respuesta:** El sistema debe ser rápido y eficiente, capaz de manejar grandes volúmenes de datos sin retrasos.
- ✓ **Escalabilidad:** El sistema debe ser escalable para adaptarse al crecimiento del consultorio sin perder rendimiento ni funcionalidades.
- ✓ **Usabilidad:** La interfaz del sistema debe ser fácil de usar y comprender, permitiendo a los usuarios realizar tareas eficientemente.
- ✓ **Mantenibilidad:** El sistema debe ser fácil de mantener y actualizar.
- ✓ **Cumplimiento de Regulaciones y Normativas:** El sistema debe cumplir con regulaciones de privacidad y protección de datos, garantizando la confidencialidad de la información.
- ✓ **Soporte Técnico:** Debe ofrecer un soporte técnico eficiente y oportuno en caso de problemas técnicos, actualizaciones o consultas relacionadas con el sistema.

#### **Stakeholders y Actores del Sistema:**

- Odontólogos: Usuarios principales del sistema, responsables de la atención odontológica y el registro de información y de programación de citas.

Pacientes: Los pacientes utilizan el sistema para acceder a sus historias clínicas y recibir recordatorios de citas.

#### ***Técnicas Utilizadas para Determinar los Requerimientos:***

**Entrevistas con odontóloga:** realizamos entrevistas con la profesional de salud bucal para entender sus necesidades.

**Observación de Tareas:** Se observaron las tareas realizadas en el consultorio odontológico para identificar procesos y flujos de trabajo existentes.

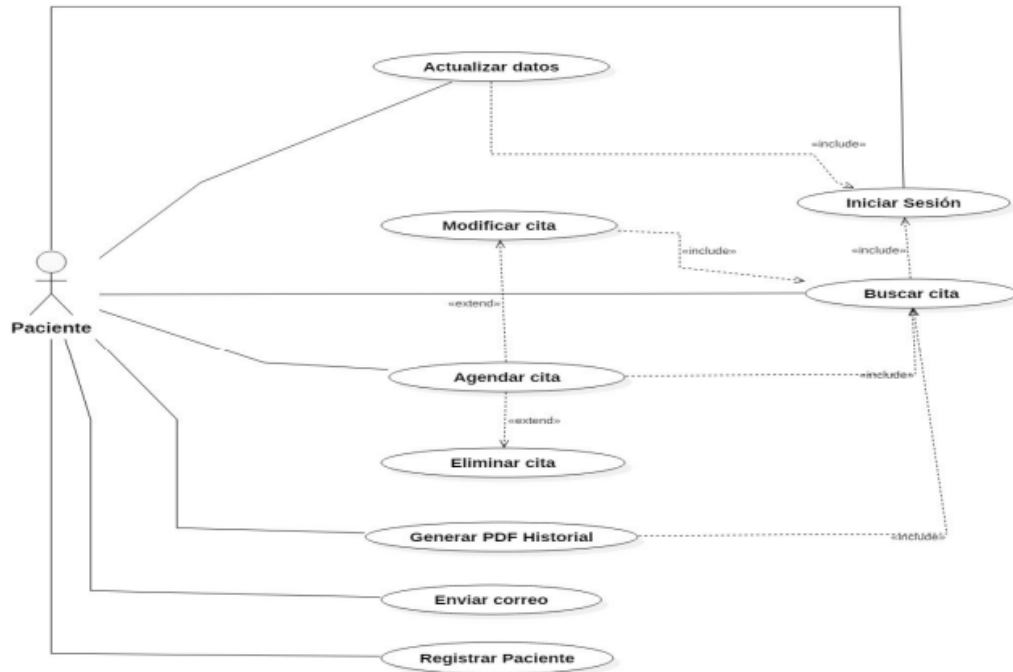
**Reuniones:** Se llevaron a cabo reuniones de trabajo con la profesional para discutir y analizar los requerimientos del sistema.

**Revisión de Documentación Existente:** Se revisaron historias clínicas de pacientes y registros existentes para identificar datos y procesos relevantes que deben incluirse en el sistema.

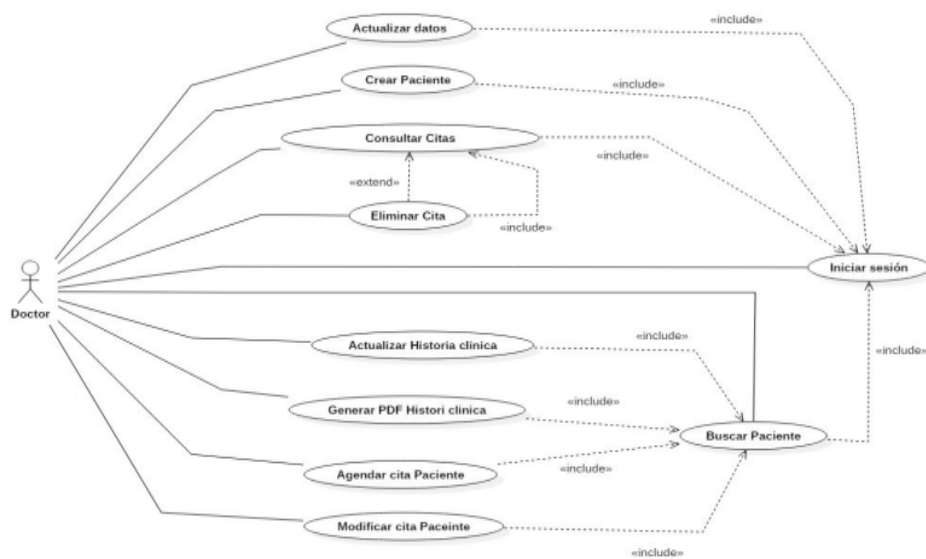
### 3. Análisis y Diseño

- Diagrama general de casos de uso

#### Caso de uso Paciente



#### Caso de uso Doctor



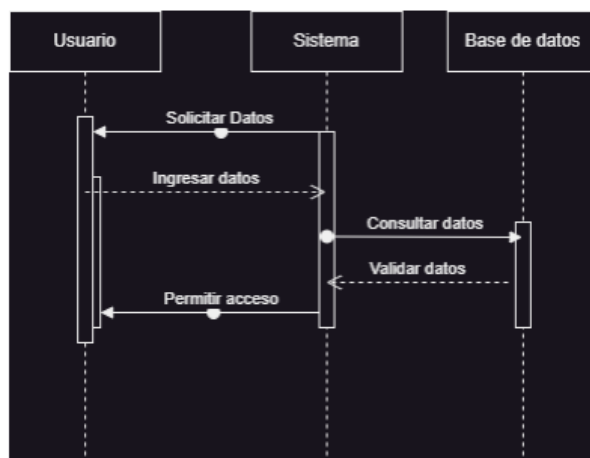
A continuación, se presentan los diagramas de casos de uso y las especificaciones en formato de texto para tres requisitos funcionales específicos:

Nombre:	Inicio de Sesión
Actores:	Usuario
Función:	Permite el ingreso mediante una autenticación.
Descripción:	<ol style="list-style-type: none"> <li>1- El sistema solicita el nombre de usuario y su contraseña.</li> <li>2- El usuario ingresa sus datos.</li> <li>3- El sistema analiza si los datos son correctos.</li> <li>4- Si los datos son válidos, el sistema habilita el ingreso. Caso contrario, lo deniega y vuelve a pedir los datos.</li> </ol>
Referencias:	De requerimientos: R1
Precondición:	1- El usuario debe haber sido registrado con anterioridad
Diagrama del caso de uso:	<pre> graph LR     subgraph Sistema         UC1([Inicio de Sesión])     end     Usuario[Usuario] --- UC1         </pre> <p>The diagram shows a stick figure actor labeled 'Usuario' connected by a line to an oval use case labeled 'Inicio de Sesión'. The use case is contained within a rectangular box labeled 'Sistema'.</p>

Nombre:	Registro y gestión de pacientes
Actores:	Usuario, base de datos
Función:	Permite el registro de datos de un paciente
Descripción:	<ol style="list-style-type: none"> <li>1- El sistema ofrece al usuario la opción de CREAR y BORRAR un paciente.</li> <li>2- En caso de que se elija crear: <ul style="list-style-type: none"> <li>• El sistema pide ingresar los datos pertinentes del paciente</li> <li>• El usuario ingresa los datos requeridos</li> <li>• El sistema registra los datos.</li> <li>• El sistema envía los datos a la base de datos.</li> </ul> </li> <li>3- En caso de que se elija eliminar: <ul style="list-style-type: none"> <li>• El sistema muestra los pacientes registrados</li> <li>• El sistema pide al usuario que elija qué paciente desea eliminar.</li> <li>• El sistema manda la consulta a la base de datos para eliminar los datos pertinentes.</li> </ul> </li> <li>4- Se guardan los cambios.</li> </ol>
Referencias:	De requerimientos: R2
Precondición:	1- El usuario debe haber sido admitido en el sistema
Diagrama del caso de uso:	<pre> graph LR     subgraph Sistema         UC1([Crear Paciente])         UC2([Eliminar Paciente])     end     Usuario[Usuario] --- UC1     Usuario --- UC2     UC1 --- BD[Base de datos]     UC2 --- BD         </pre> <p>The diagram shows two stick figure actors: 'Usuario' on the left and 'Base de datos' on the right. A central rectangular box labeled 'Sistem' contains two oval use cases: 'Crear Paciente' and 'Eliminar Paciente'. 'Usuario' is connected to both use cases. Both use cases are connected to 'Base de datos'.</p>

Nombre:	Consultar historial clínico
Actores:	Usuario, base de datos
Función:	Permite la consulta de un historial clínico existente.
Descripción:	<ol style="list-style-type: none"> <li>1- El sistema requiere al usuario ingresar un nombre para encontrar el registro del paciente deseado.</li> <li>2- El sistema realiza una consulta a la base de datos.</li> <li>3- En caso de no existir, se muestra un mensaje de error. En caso de existir, se traen todos <u>los</u> datos del paciente solicitado.</li> </ol>
Referencias:	De requerimientos: R3
Precondición:	1- El usuario debe haber sido admitido en el sistema. La historia clínica debe estar relacionada con un paciente registrado.
Diagrama del caso de uso:	<pre> graph LR     Usuario((Usuario)) --- Sistem[Sistem]     subgraph Sistem         ConsultarHistorial((Consultar historial))     end     Sistem --- BaseDatos[Base de datos] </pre>

#### Diagrama de Secuencia (Caso de uso: Inicio de sesión)



## **Parte 1.**

Presentación en plataforma de un documento formato PDF con el siguiente contenido:

### ***Inicio***

- Primera página tipo portada con imagen y título del proyecto de [software](#); y nombre de los integrantes del equipo de realización (quienes están interviniendo ya que este trabajo se defiende de manera presencial).

### ***Parte Central***

- Páginas siguientes de Introducción con títulos y breve resumen del tema del [software](#), el problema presentado, la solución ideada y la mejora que se espera con la implementación del [software](#) propuesto.
- **Requisitos con la especificación de requisitos de sistema funcionales y no funcionales. Descripción de los Stakeholders y los actores de sistema.**
- **Análisis y Diseño** con diagrama general de casos de uso, diagrama de clases completo y el diagrama de secuencia de un caso de uso.
- Implementación. Descripción de alguna rutina del [software](#) con un diagrama de flujo y lista de comandos que se usarían en lenguaje elegido.



- [V&V](#). Lista de 10 casos de prueba con las entradas que se van a probar y el resultado que se espera, escritos para condiciones de entrada válida como para condiciones inválidas.
- [Mantenimiento](#). Descripción de las características específicas que tendrá el [software](#) para cumplir con el requisito de calidad de Mantenibilidad.

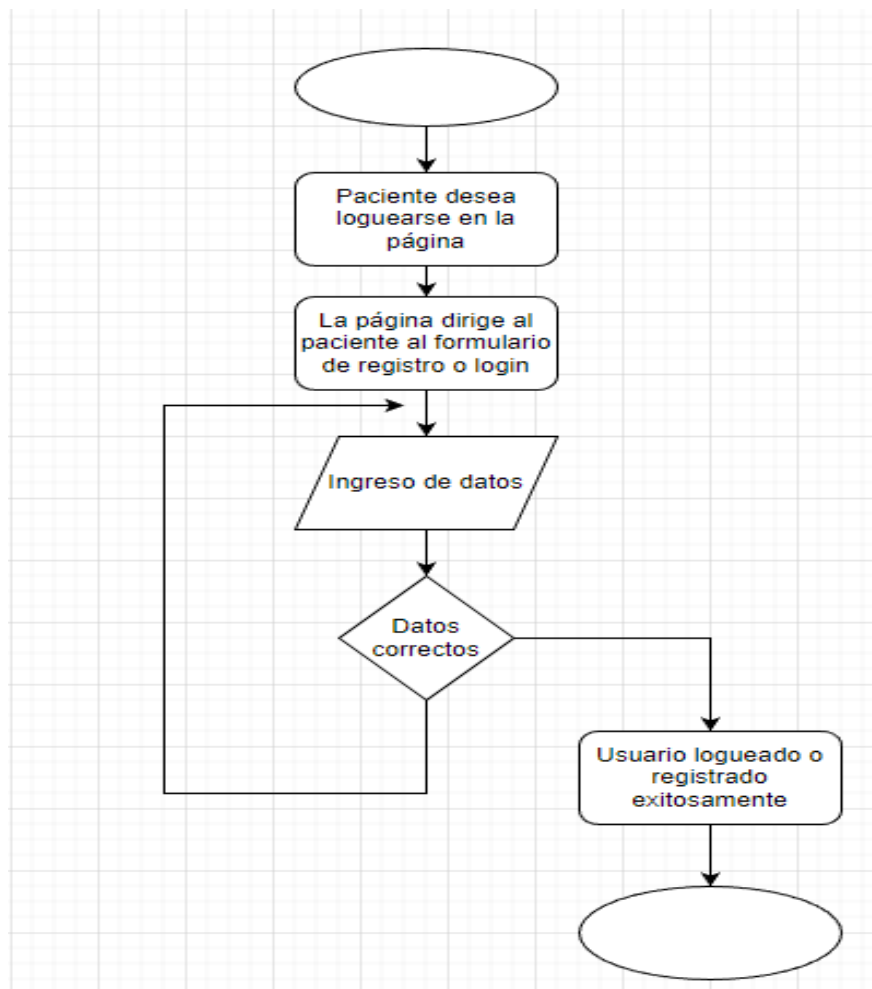
#### **Final**

- Problemas presentados en el presente cuatrimestre para las actividades del [software](#) y si se han resuelto, de qué manera
- Problemas relacionados al equipo en si mismo, soluciones ideadas.
- Conocimiento o capacidades que deberá adquirir el equipo para llegar a buen término con el desarrollo del [software](#).
- Habilidades y /o capacidades que ya tienen los integrantes y que aportan al desarrollo del [software](#) propuesto.

#### **Parte 2.**

- Defensa presencial del trabajo. Cada integrante deberá defender una parte distinta del trabajo presentado y tiene hasta 2 minutos (y no mas de 2 minutos cada integrante).

#### **4. Implementación: Descripción de alguna rutina del software con un diagrama de flujo y lista de comandos que se usarían en lenguaje elegido**



```

<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Formulario Login - Consultorio Odontológico</title>
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css"
      integrity="sha512-
iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcentmpDlypASozpmT/E0iPtmFIB46Zmdt
Ac9eNBvH0H/ZpiBw=="
      crossorigin="anonymous"
      referrerpolicy="no-referrer"
    />
    <style>
      * {
        margin: 0;
        padding: 0;
        font-family: Arial, Helvetica, sans-serif;
        box-sizing: border-box;
        text-decoration: none;
      }

      .container {
        width: 100%;
        height: 100vh;
        background-color: #d0d0d0;
        display: flex;
        align-items: center;
        justify-content: center;
      }

      .form-content {
        width: 100%;
        max-width: 450px;
        background-color: #2c2f35;
        padding: 50px 60px 70px;
        text-align: center;
        border-radius: 25px;
      }

      .form-content h1 {
        font-size: 30px;
        margin-bottom: 60px;
        color: #ffffff;
        position: relative;
      }

      .form-content h1::after {
        content: "";
        width: 30px;
        height: 4px;
        border-radius: 3px;
        background-color: #04bd7d;
        position: absolute;
        bottom: -12px;

```

```
    left: 50%;
    transform: translate(-50%);
}

.input-field {
    background-color: #515966;
    margin: 15px 0;
    border-radius: 3px;
    display: flex;
    align-items: center;
    transition: max-height 0.1s;
    overflow: hidden;
}

input {
    width: 100%;
    background-color: transparent;
    border: 0;
    outline: none;
    padding: 18px 15px;
    color: #ffffff;
}

input::placeholder {
    color: #ffffff4e;
}

.input-field i {
    margin-left: 15px;
    color: #04bd7d;
}

form p {
    color: #ffffff;
    text-align: left;
    font-size: 13px;
}

form p a {
    color: #04bd7d;
    margin-left: 5px;
}

.btn-field {
    width: 100%;
    display: flex;
    justify-content: space-between;
}

.btn-field button {
    flex-basis: 48%;
    font-size: 16px;
    background-color: #04bd7d;
    color: #ffffff;
    height: 40px;
    border-radius: 10px;
    border: 0;
    outline: 0;
    cursor: pointer;
    transition: background 1s;
}
```

```

.btn-field button:hover {
  font-size: 18px;
}

.input-group {
  height: 250px;
}

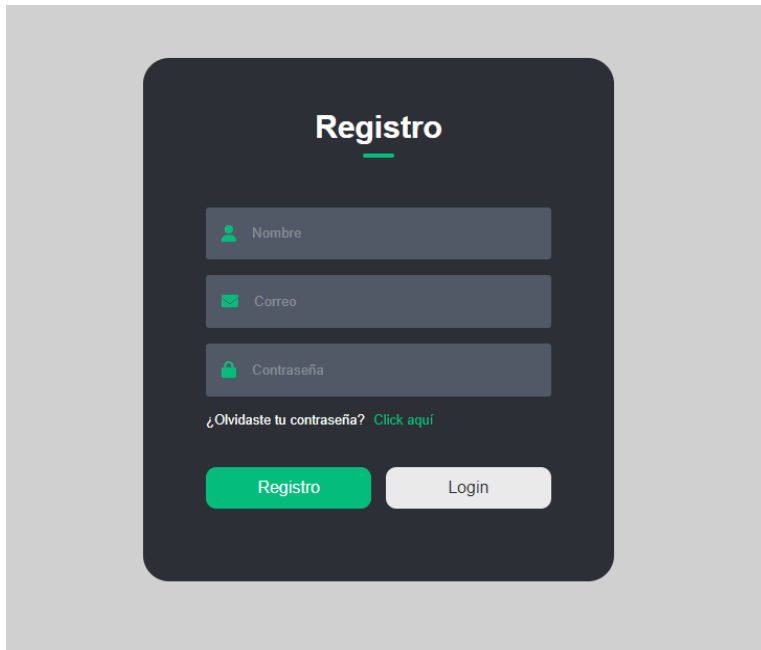
.btn-field button.disable {
  background-color: #eaeaea;
  color: #2c2f35;
}
</style>
</head>
<body>
  <div class="container">
    <div class="form-content">
      <h1 id="title">Registro</h1>
      <form>
        <div class="input-group">
          <div class="input-field" id="nameInput">
            <i class="fa-solid fa-user"></i>
            <input type="text" placeholder="Nombre" />
          </div>
          <div class="input-field">
            <i class="fa-solid fa-envelope"></i>
            <input type="text" placeholder="Correo" />
          </div>
          <div class="input-field">
            <i class="fa-solid fa-lock"></i>
            <input type="password" placeholder="Contraseña" />
          </div>
          <p>¿Olvidaste tu contraseña? <a
href="https://www.itse.gob.ar/" target="_blank">Click aquí</a></p>
        </div>
        <div class="btn-field">
          <button type="button" id="signUp">Registro</button>
          <button type="button" id="signIn"
class="disable">Login</button>
        </div>
      </form>
    </div>
  </div>
  <script>
    const signUp = document.getElementById("signUp");
    const signIn = document.getElementById("signIn");
    const nameInput = document.getElementById("nameInput");
    const title = document.getElementById("title");

    signIn.onclick = function () {
      nameInput.style.maxHeight = "0";
      title.innerHTML = "Login";
      signUp.classList.add("disable");
      signIn.classList.remove("disable");
    };

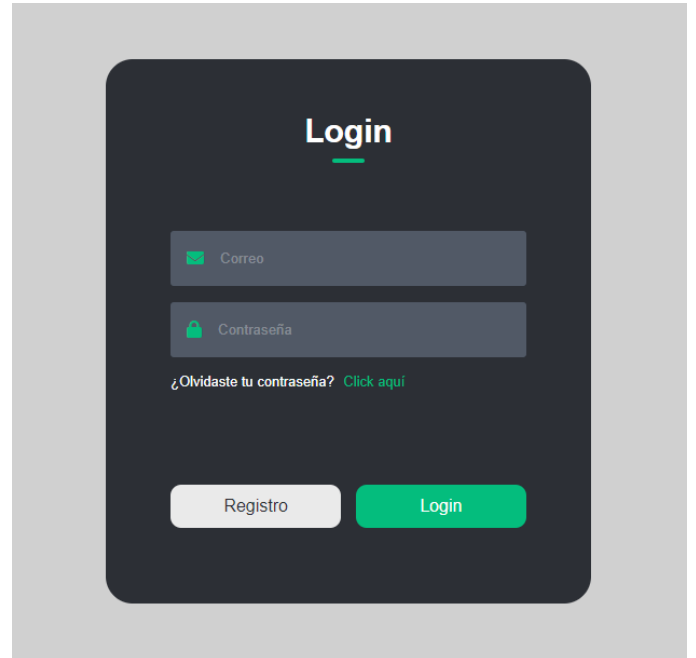
    signUp.onclick = function () {
      nameInput.style.maxHeight = "50px";
      title.innerHTML = "Registro";
      signUp.classList.remove("disable");
    };
  </script>

```

```
        signIn.classList.add("disable");
    };
</script>
</body>
</html>
```



The image shows a registration form titled "Registro" on a dark background. It features three input fields: "Nombre" with a person icon, "Correo" with an envelope icon, and "Contraseña" with a lock icon. Below the fields is a link "¿Olvidaste tu contraseña? Click aquí". At the bottom are two buttons: "Registro" (green) and "Login" (white).



The image shows a login form titled "Login" on a dark background. It features two input fields: "Correo" with an envelope icon and "Contraseña" with a lock icon. Below the fields is a link "¿Olvidaste tu contraseña? Click aquí". At the bottom are two buttons: "Registro" (white) and "Login" (green).

Copiar y pegar código fuente en documento HTML: <https://tohtml.com/>