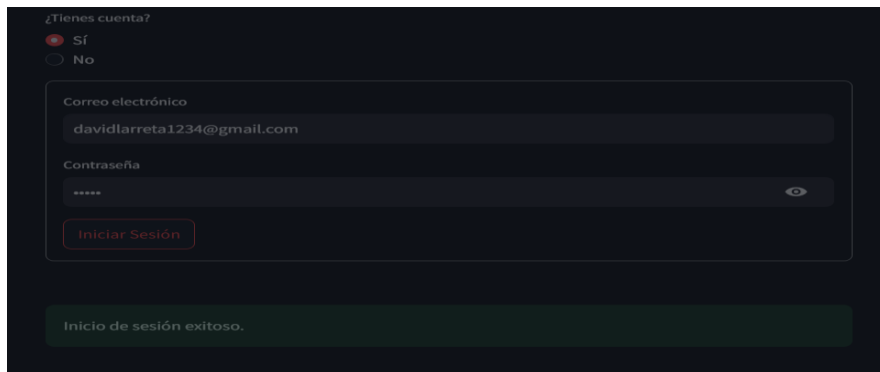


## Criterio C: Desarrollo

### Técnicas Utilizadas para el desarrollo de la solución informática (De más compleja a menos compleja):

Autenticación y gestión de credenciales con Google API:.....	1
Gestión de cookies encriptadas: .....	3
Envío de correos electrónicos usando SMTP: .....	4
Comprobación y manejo de conflictos en reservaciones:.....	5
Uso de marcos de tiempo y fechas en Python: .....	7
Manejo de excepciones:.....	8
Despliegue condicional de contenido con streamlit: .....	9

### Autenticación y gestión de credenciales con Google API:



¿Tienes cuenta?

☒ Sí

☐ No

Correo electrónico

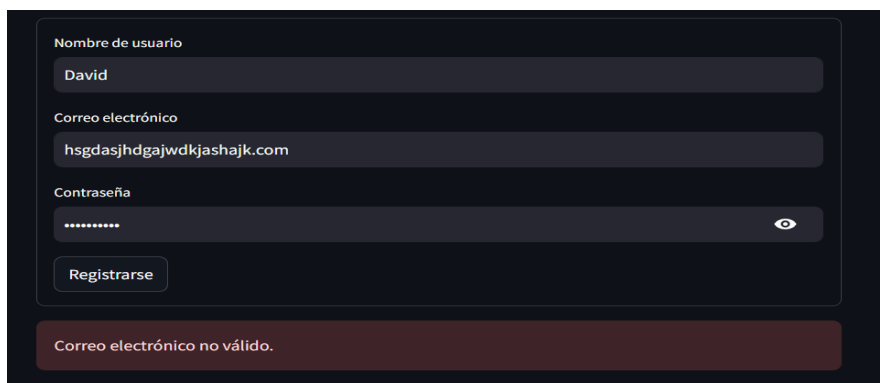
davidlarreta1234@gmail.com

Contraseña

\*\*\*\*\*

Iniciar Sesión

Inicio de sesión exitoso.



Nombre de usuario

David

Correo electrónico

hsgdasjhdgajwdkjashajk.com

Contraseña

\*\*\*\*\*

Registrarse

Correo electrónico no válido.

```
def autenticar_usuario(correo, contrasena, es_admin):
    worksheet = login_worksheet if es_admin else usuarios_worksheet
    usuarios = worksheet.get_all_records()
    for usuario in usuarios:
        if usuario['correo'] == correo and str(usuario['Contraseña']) == contrasena:
            return True
    return False
```

```
def pagina_inicio_sesion():
    st.header("Inicio de Sesión")

    if 'tipo_usuario' not in st.session_state:
        tipo_usuario = st.radio("¿Eres cliente o administrador?", ("Cliente", "Administrador"))
        submit_tipo = st.button("Enviar")

        if submit_tipo:
            st.session_state['tipo_usuario'] = tipo_usuario
            st.experimental_rerun()
    else:
        tipo_usuario = st.session_state['tipo_usuario']

        if tipo_usuario == "Administrador":
            with st.form(key='login_form'):
                correo = st.text_input("Correo electrónico")
                contrasena = st.text_input("Contraseña", type="password")
                submit_button = st.form_submit_button(label='Iniciar Sesión')

            if submit_button:
                if autenticar_usuario(correo, contrasena, es_admin=True):
                    st.session_state['logged_in'] = True
                    st.session_state['usuario'] = correo
                    cookies["logged_in"] = "True"
                    cookies["tipo_usuario"] = tipo_usuario
                    cookies["usuario"] = correo
                    cookies.save()
                    st.success("Inicio de sesión exitoso.")
                    st.experimental_rerun()
```

```
def pagina_inicio_sesion():
    else:
        tiene_cuenta = st.radio("¿Tienes cuenta?", ("Sí", "No"))

        if tiene_cuenta == "No":
            with st.form(key='registro_form'):
                usuario = st.text_input("Nombre de usuario")
                correo = st.text_input("Correo electrónico")
                contrasena = st.text_input("Contraseña", type="password")
                submit_button = st.form_submit_button(label='Registrarse')

            if submit_button:
                if len(contrasena) < 5:
                    st.error("La contraseña debe tener al menos 5 caracteres.")
                elif not "@" in correo:
                    st.error("Correo electrónico no válido.")
                else:
                    usuarios_worksheet.append_row([usuario, correo, contrasena])
                    st.success("Registro exitoso. Ahora puedes iniciar sesión.")
            else:
                with st.form(key='login_form'):
                    correo = st.text_input("Correo electrónico")
                    contrasena = st.text_input("Contraseña", type="password")
                    submit_button = st.form_submit_button(label='Iniciar Sesión')

                if submit_button:
                    if autenticar_usuario(correo, contrasena, es_admin=False):
                        st.session_state['logged_in'] = True
                        st.session_state['usuario'] = correo
                        cookies["logged_in"] = "True"
                        cookies["tipo_usuario"] = tipo_usuario
                        cookies["usuario"] = correo
```

Este método utiliza funciones relativamente sencillas como la conexión de APIs de Google Sheets las cuales hacen que sea como una base de datos para el programa, esta solución me pareció la más ingeniosa, no solo para mí sino también para mi cliente, puesto como el recién está empezando a adentrarse en el mundo de la informática y lo

digital, no sería lo más lógico ponerlo a gestionar una base de datos y él ya está relacionado con una herramienta más parecida como lo es una hoja de cálculo. Con esto se cumple el criterio de verificación de credenciales y creación de las nuevas credenciales.

## Gestión de cookies encriptadas:

```
def cerrar_sesion():
    st.session_state['logged_in'] = False
    st.session_state.pop('usuario', None)
    st.session_state.pop('tipo_usuario', None)
    cookies["logged_in"] = "False"
    cookies["tipo_usuario"] = ""
    cookies["usuario"] = ""
    cookies.save()
    st.experimental_rerun()

def cambiar_a_administrador():
    st.session_state['logged_in'] = False
    st.session_state.pop('usuario', None)
    st.session_state.pop('tipo_usuario', None)
    cookies["logged_in"] = "False"
    cookies["tipo_usuario"] = ""
    cookies["usuario"] = ""
    cookies.save()
    st.experimental_rerun()
```

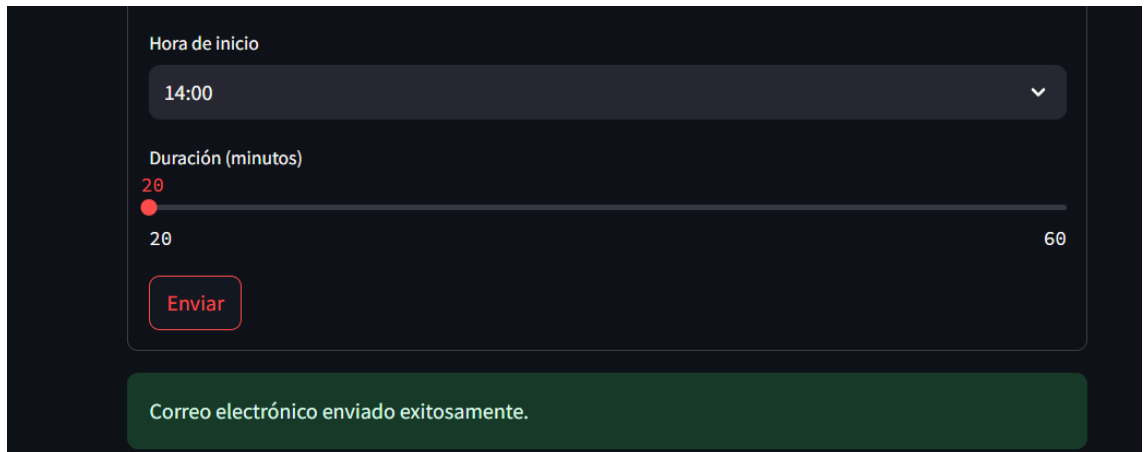
```
# Configurar el gestor de cookies
cookies = EncryptedCookieManager(prefix="streamlit_app", password="your_secret_password")

# Asegúrate de llamar a la función de cookies en el inicio de tu aplicación
if not cookies.ready():
    st.stop()
```

```
if submit_button:
    if autenticar_usuario(correo, contrasena, es_admin=False):
        st.session_state['logged_in'] = True
        st.session_state['usuario'] = correo
        cookies["logged_in"] = "True"
        cookies["tipo_usuario"] = tipo_usuario
        cookies["usuario"] = correo
        cookies.save()
        st.success("Inicio de sesión exitoso.")
        st.experimental_rerun()
    else:
        st.error("Correo o contraseña incorrectos.")
```

La gestión de cookies me ayudó en que se mantenga el inicio de sesión activado, incluso si se cierra la pestaña o si se reinicia la pestaña, además estas cookies tienen la labor de verificar si esta iniciada una sesión o no, es la técnica que yo considere la técnica más difícil de entender y aplicar, además podría decir que fue la técnica a la que mas tiempo le dedique.

Envío de correos electrónicos usando SMTP:



```
def enviar_correo(sender_email, mensaje, fecha, hora_inicio, hora_fin, nombre):
    receiver_email = "fabricio14@hotmail.com"
    gmail_user = "davidlarreta1234@gmail.com"
    gmail_password = "dvpe fumi zrfr ixjy"

    # Crear el mensaje
    msg = MIMEText(mensaje)
    msg['From'] = gmail_user
    msg['To'] = receiver_email
    msg['Subject'] = "Nueva Reserva"

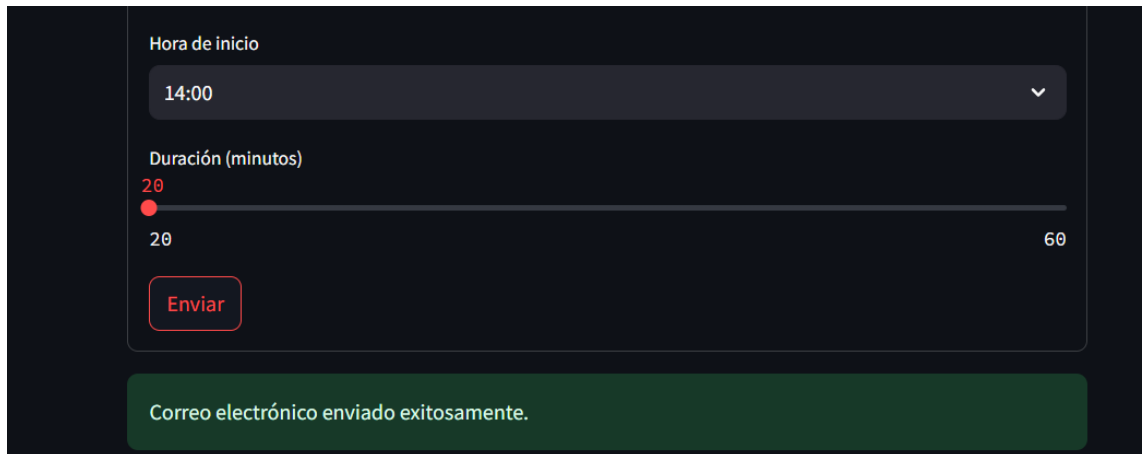
    cuerpo_mensaje = f"Nombre: {nombre}\nCorreo electrónico: {sender_email}\nMensaje: {mensaje}\nFecha: {fecha}"
    msg.attach(MIMEText(cuerpo_mensaje, 'plain'))

    # Enviar el correo
    try:
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(gmail_user, gmail_password)
        text = msg.as_string()
        server.sendmail(gmail_user, receiver_email, text)
        server.quit()
        st.success("Correo electrónico enviado exitosamente.")
    except Exception as e:
        st.error(f"Error al enviar el correo electrónico: {e}")
```

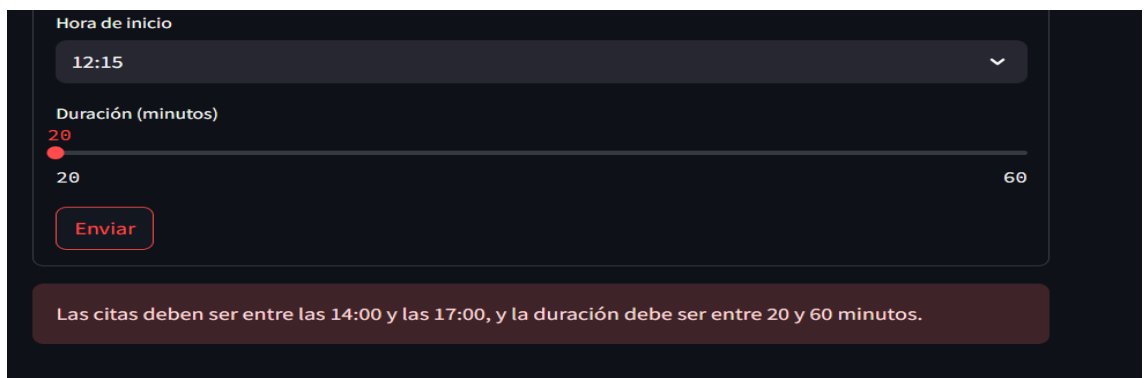
Se usa un correo auxiliar con una clave de aplicación para poder usar este correo como intermediario entre mi cliente y los usuarios, esto me pareció la solución más ingeniosa, puesto que además de que el correo le llega perfectamente y con todos los

datos necesarios para responder a la reservación. Además, dentro de esta función aparte de tomar todos los valores ingresados por el usuario, ya de por si crea el cuerpo del mensaje, poniendo de título, nueva reservación y poniendo todos los datos del formulario, incluyendo el correo del inicio sesión, puesto no se envía desde ese correo sino desde el correo auxiliar. Por lo que mi cliente sabrá correctamente el correo al que escribirle, ayudando al criterio de gestión de credenciales y el de envío de correo electrónico al mismo tiempo.

### Comprobación y manejo de conflictos en reservaciones:



This screenshot shows a reservation form with a dark theme. The 'Hora de inicio' (Start time) is set to 14:00. The 'Duración (minutos)' (Duration in minutes) is set to 20, with a range from 20 to 60 minutes indicated by a slider. A red 'Enviar' (Send) button is visible. Below the form, a green notification bar states: 'Correo electrónico enviado exitosamente.' (Email sent successfully).



This screenshot shows the same reservation form, but with a validation error. The 'Hora de inicio' is set to 12:15, which is outside the allowed range of 14:00 to 17:00. The 'Duración' is still 20 minutes. The 'Enviar' button is present. A red notification bar at the bottom states: 'Las citas deben ser entre las 14:00 y las 17:00, y la duración debe ser entre 20 y 60 minutos.' (Appointments must be between 14:00 and 17:00, and the duration must be between 20 and 60 minutes).

```
if hora_inicio < datetime.time(14, 0) or hora_fin > datetime.time(17, 0):  
    st.error("Las citas deben ser entre las 14:00 y las 17:00, y la duración debe ser entre 20 y 60 min")
```

```
if conflicto:  
    st.error("Esta franja horaria ya está reservada. Por favor, elige otra hora.")
```

```
if submit_button:  
    if email != st.session_state['usuario']:  
        st.error("El correo electrónico debe coincidir con el usado para iniciar sesión.")  
        return
```

Esto es una técnica de comprobación dentro de un archivo/base de datos, pero es una parte esencial el comprobar dentro del Google Sheets la información que se quiere escribir dentro del formulario, puesto muchas cosas tienen que coincidir para que funcione. Esta parte/función del programa me pareció lo más correcto y práctica, además que al verificar que el correo sea el mismo que el del inicio de sesión, con esto protejo también la seguridad de los usuarios, al hacer que nadie se pueda hacer pasar por ellos. Esto no solo mejora el orden y seguridad, sino también la organización de mi cliente, por lo que ayuda al criterio de gestión de credenciales.

## Uso de marcos de tiempo y fechas en Python:

```
try:
    formulario_worksheet.append_row([nombre, email, mensaje, str(fecha), str(hora_inicio), str(h
    # Crear evento en Google Calendar
    event = {
        'summary': 'Cita de {}'.format(nombre),
        'description': mensaje,
        'start': {
            'dateTime': '{}T{}'.format(fecha, hora_inicio),
            'timeZone': 'UTC',
        },
        'end': {
            'dateTime': '{}T{}'.format(fecha, hora_fin),
            'timeZone': 'UTC',
        }
    }

    calendar_service.events().insert(calendarId='davidlarreta1234@gmail.com', body=event).execut

    enviar_correo(email, mensaje, fecha, hora_inicio, hora_fin, nombre)

    st.success("Correo enviado exitosamente")
except Exception as e:
    st.error(f"Error al enviar los datos a Google Sheets, Google Calendar o correo electrónico:
```

```
# Incrustar el calendario de Google
st.markdown(
    """
    <iframe src="https://calendar.google.com/calendar/embed?src=davidlarreta1234%40gmail.com&ctz=UTC" style="t
    """,
    unsafe_allow_html=True,
)
```

Esto se usa para meter un evento dentro del calendario y dentro del correo electrónico que se envía a mi cliente, agregar este evento al calendario de Google no solo ayuda a que la aplicación web se vea interactiva, simple pero llamativa, también ayuda para que el usuario no tengo que estar adivinando a qué hora puede hacer una reserva. Además de que es un método bastante sencillo, ya que el propio Google Calendar te da la opción de incrustar su calendario en el código como un iframe. Por lo que para mi cliente sería como manejar su Google Calendar sin necesidad de intervenir en el código porque es una opción propia de este sitio.

## Manejo de excepciones:

```
if submit_button:
    if email != st.session_state['usuario']:
        st.error("El correo electrónico debe coincidir con el usado para iniciar sesión.")
        return

    hora_fin = (datetime.datetime.combine(datetime.date.today(), hora_inicio) + datetime.timedelta(minutes=60))

    if hora_inicio < datetime.time(14, 0) or hora_fin > datetime.time(17, 0):
        st.error("Las citas deben ser entre las 14:00 y las 17:00, y la duración debe ser entre 20 y 60 minutos.")
    else:
        reservas = formulario_worksheet.get_all_records()
        nueva_reserva_inicio = f"{fecha} {hora_inicio}"
        nueva_reserva_fin = f"{fecha} {hora_fin}"
        conflicto = False

        for reserva in reservas:
            if 'Fecha' in reserva and 'Hora de inicio' in reserva and 'Hora de fin' in reserva:
                reserva_inicio = f"{reserva['Fecha']} {reserva['Hora de inicio']}"
                reserva_fin = f"{reserva['Fecha']} {reserva['Hora de fin']}"
                if (nueva_reserva_inicio >= reserva_inicio and nueva_reserva_inicio < reserva_fin) or (nueva_reserva_fin >= reserva_inicio and nueva_reserva_fin < reserva_fin):
                    conflicto = True
                    break
```

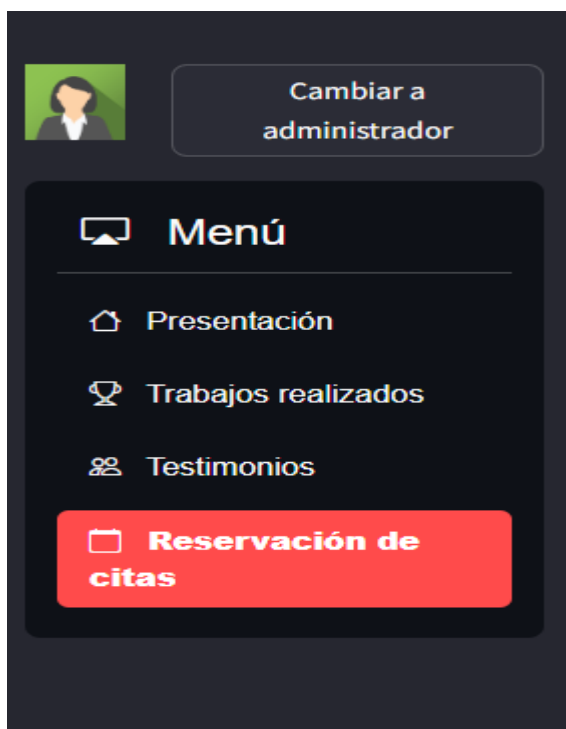
```
except Exception as e:
    st.error(f"Error al enviar los datos a Google Sheets, Google Calendar o correo electrónico: {e}")
```

En esta técnica, podemos ver como maneja algún error que tenga el programa, puesto normalmente si un programa no funciona va a salir error en la línea 23 y un pantallazo rojo por decir algo, eso es lo que queremos evitar, de que en el caso de que haya un error en alguna línea o en algún dato no se muestre un mensaje en rojo que haga el programa deje de funcionar, simplemente se le mostrara un mensaje al usuario diciendo que hubo un error en el envío de los datos, esto ayuda a que el programa tenga una forma más “amigable” de mostrar un error. Además, con esta función también se verifican los datos que se encuentran dentro del formulario y se avisa por cualquier problema que tenga el programa, por ejemplo, si los datos en la base de datos tienen la misma hora y el mismo día, no mostrara un error, sino que se le pedirá al usuario que vuelva a ingresar otro día u hora.



## Despliegue condicional de contenido con streamlit:

Esta para mí fue la estrategia más importante a la hora de la creación del programa, pues esta librería me permitió crear este programa en el lenguaje de programación que ya manejo el cual es Python, además el nombre de sus funciones ayuda bastante, pues me da opciones bastante interactivas y sencillas para crear una aplicación web funcional.



### **Bibliografía:**

- Backyard, D. D. (2023, noviembre 14). *Email Form using Python and Streamlit — Sending emails with attachment using best practices*. Medium.  
<https://blog.devgenius.io/email-form-using-python-and-streamlit-sending-emails-with-attachment-using-best-practices-122a525753a>
- Chaithanya Vamshi Sai. (2023, marzo 2). *Building Contact Form with Email Integration using Python & Streamlit*. <https://www.youtube.com/watch?v=Ep24FCpO2Ak>
- Coding Is Fun. (2022, mayo 26). *How to Add a User Authentication Service (Login Form) in Streamlit*. <https://www.youtube.com/watch?v=JoFGrSRj4X4>
- Coding Is Fun. (2023a, agosto 27). *Streamlit & Google Sheets: The Easiest “Database”*.  
<https://www.youtube.com/watch?v=HwxrXnYVIIU>
- Coding Is Fun. (2023b, octubre 29). *Create a Google Sheets Data Entry Form with Python & Streamlit / Quick & Easy Tutorial*.  
[https://www.youtube.com/watch?v=\\_G5f7og\\_Dpo](https://www.youtube.com/watch?v=_G5f7og_Dpo)
- Diego G.S. (2020, abril 20). *11.c. HTML. Iframes. Insertar un calendario de Google en nuestra página*. <https://www.youtube.com/watch?v=sdNpZnF5xaM>
- NeuralNine. (2023, mayo 26). *Google Calendar Automation in Python*.  
<https://www.youtube.com/watch?v=B2E82UPUnOY>
- Toni Dev. (2023a, mayo 14).  *Crea una web con Python y Streamlit en pocos minutos*. [https://www.youtube.com/watch?v=mAhJ\\_mQEoO4](https://www.youtube.com/watch?v=mAhJ_mQEoO4)
- Toni Dev. (2023b, junio 10).  *Cómo usar Google Sheet como base de datos en python*.  
<https://www.youtube.com/watch?v=b07HiZCwNx8>
- Toni Dev. (2023c, agosto 14).  *Cómo usar la API de Google Calendar con python*.  
<https://www.youtube.com/watch?v=eqZyPYlp3nw>