

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

Obligatorio

Taller de Ingeniería de Software



Antonella Ferrín Montaña

Nro. Estudiante: 338267



Santiago Armand Pilon

Nro. Estudiante: 245347

Docente: Natalia Giordano

Tabla de contenido

I.	Introducción	3
II.	Requerimientos Funcionales	4
III.	Requerimientos no funcionales.....	5
IV.	Reglas de negocios	5
V.	Historias de usuario	6
VI.	Priorización de requerimientos	14
VII.	Mockups.....	16
VIII.	Trello	21
IX.	Pruebas Unitarias JEST	24

I. Introducción

El presente trabajo tiene como objetivo el desarrollo de una aplicación web para el cliente Juan Mendoza, titular de la barbería “Estilo Clásico”, ubicada en la ciudad de Montevideo. Con la intención de modernizar su negocio y brindar una experiencia más cómoda a sus clientes, Juan ha manifestado la necesidad de contar con una herramienta que permita gestionar las reservas de forma ágil y sencilla.

Actualmente, muchos de sus clientes le consultan por una alternativa digital que facilite la solicitud de turnos. Ante esta demanda, el cliente se ha contactado con nuestro equipo para acompañarlo en este proceso de transformación digital.

El objetivo principal es desarrollar una solución web simple, intuitiva y visualmente profesional, que no solo optimice la gestión de citas, sino que también resulte fácil de usar tanto para los usuarios como para el propio Juan Mendoza.

II. Requerimientos Funcionales

1. Secciones claras : Inicio, Reservas, Servicios, Barberos, Precios , Contacto y Login.
2. **Sección Inicio:** debe tener el lema de la empresa “tradición y estilo en cada corte” * logo.
3. **Sección Reservas:** Formulario para los clientes se puedan agendar con su nombre completo, teléfono, email, puedan elegir barbero (opcional), tipo de servicios a realizarse, fecha y horario (se dan turnos cada media hora de 9 a 18hs). Un mismo cliente no puede reservar el mismo día a la misma hora. Una vez hecha la reserva, ningún usuario puede reservar ese día a esa hora con ese barbero.
4. Al agendarse deberá surgir un cartel con la información de la reserva.
5. **Sección Servicios:** fotos con servicios realizados anteriormente por la barbería con pequeña descripción.
6. **Sección Barberos:** foto del barbero, especialidad y breve descripción de su trayectoria.
7. **Sección Precios:** Precio fijo de cada servicio. Solo se cobra en efectivo.
8. **Sección Contacto:** Mapa con la ubicación del local de la barbería, teléfono, email, horarios, redes sociales. Especificar qué días no se trabaja (los 5 feriados no laborales del año).
9. **Sección Login:** solo para el dueño de la Barberia (usuario: admin, password : Barberia). Solo podrá ver las reservas del día corriente.
10. **Login de usuarios**
11. **Reseñas de clientes** sobre los servicios prestados.

III. Requerimientos no funcionales

1. Servidor pequeño. Se esperan no más de 100 usuarios diarios
2. Seguridad, protocolo HTTPS de implementación
3. La interfaz debe estar disponible en idioma español
4. El sistema debe funcionar correctamente en los navegadores más utilizados: Chrome, Firefox y Safari, tanto en desktop como en dispositivos móviles.

IV. Reglas de negocios

1. Lema en la sección inicio : “tradición y estilo en cada corte”
2. Colores gris, azul, negro.

V. Historias de usuario

Al trabajar con la metodología de desarrollo ágil SCRUM, las historias de usuario son muy común y útil de expresar requerimientos funcionales como descripciones breves y simples de una funcionalidad o necesidad del sistema escritas desde la perspectiva del usuario final o del cliente.

HISTORIA DE USUARIO	
Número: 1	Usuario: Cliente
Nombre Historia: Ver página de inicio	
DoR: aceptación de la interfaz por el Cliente	DoD: pasar tetsing
Esfuerzo Fibonacci: 3	Priorización: 4
Descripción: al ingresar a la página, lo primero que ve el usuario es la página de inicio o “index” donde se puede visualizar: header de la página con todas las secciones, foto de la barbería y lema “tradición y estilo en cada corte”.	
Como cliente de la barbería, quiero ver una página de inicio atractiva donde pueda ver el logo y lema de la barbería para sentir confianza y motivación para continuar navegando y agendar un turno.	
CA #1 <u>Página de inicio con logo y lema.</u> Dado un cliente de la barbería cuando accede a la página de inicio, entonces puede visualizar una sección superior con el logo de la empresa y las opciones: Inicio, Reservar Turno, Precios, Barbero, Galería, Contacto y Login. Debajo de dicha sección el usuario puede visualizar una imagen de la Barberia y el lema “Tradición y estilo en cada corte”. La sección superior con las opciones se mantiene fija en todas las secciones.	

HISTORIA DE USUARIO	
Número: 2	Usuario: Cliente
Nombre Historia: Reserva de turnos.	
DoR: Diseño del formulario aprobado por el cliente. Campos definidos. Validaciones de datos acordadas.	DoD: Formulario funcional en la sección "Reservas", con validaciones. Al confirmar, se guarda la reserva en el sistema y se muestra el cartel resumen.
Esfuerzo Fibonacci: 8	Priorización: 1
<p>Descripción: al seleccionar en el header, sección “reserva”, el usuario podrá visualizar un formulario para realizar la reserva. Se le solicitarán los siguientes datos: nombre y apellido, teléfono, email, selección de barbero, tipo de servicio a realizarse, horario (estos últimos tres datos se seleccionarán a través de combo desplegable) y fecha (a través de un calendario). Seguido se mostrará un botón para confirmar la reserva.</p>	
<p>Como cliente, quiero poder agendar un turno en la barbería eligiendo fecha, horario y tipo de servicio, para asegurarme de tener mi cita reservada sin necesidad de llamar o presentarme físicamente.</p>	
<p><u>CA #2: Agenda de turnos.</u> Dado un cliente de la barbería. Cuando accede a la sección para agendar un turno. Entonces puede visualizar un formulario para completar con los campos: Nombre, teléfono, barbero, servicio, fecha, hora y email.</p>	
<p><u>CA #3: Nombre.</u> Dado un cliente cuando selecciona el input para completar su nombre, entonces puede escribir solo letras con un mínimo de 5 caracteres y un máximo de 30. Obligatorio.</p>	
<p><u>CA #4: Teléfono.</u> Dado un cliente cuando selecciona el input para completar su teléfono, entonces puede escribir solo números con una extensión de 9 caracteres. Obligatorio.</p>	

CA #5: Barbero Dado un cliente cuando selecciona el combo desplegable para seleccionar un barbero con quien realizarse el servicio, entonces puede seleccionar un barbero del combo desplegable o dejar la opción por defecto – Cualquiera-.

CA #5: Servicio Dado un cliente cuando selecciona el combo desplegable para seleccionar un servicio, entonces puede seleccionar uno del combo desplegable. Obligatorio.

CA #6: Fecha Dado un cliente cuando selecciona la opción de "fecha", entonces se despliega un calendario donde el cliente puede seleccionar día en que desea realizarse un servicio.

CA #7: Hora Dado un cliente cuando selecciona el combo desplegable para seleccionar una hora, entonces puede seleccionar un turno entre las 9 y 18hs en turnos de media hora. Obligatorio.

CA #8: Email. Dado un cliente cuando selecciona el input para completar su correo electrónico. Entonces puede ingresar letras, números o caracteres especiales, con una longitud mínima de 5 y máxima de 40 caracteres, incluyendo obligatoriamente el carácter "@". Campo obligatorio.

HISTORIA DE USUARIO	
Número: 3	Usuario: Cliente
Nombre Historia: Confirmación de reserva.	
DoR: El formulario de reserva ya está desarrollado y completado por el usuario. Se definió el diseño del cartel de confirmación.	DoD: Al enviar una reserva válida, se muestra un cartel con el día, hora, barbero (si lo hay) y servicio seleccionado.
Esfuerzo Fibonacci: 3	Priorización: 3
Descripción: al seleccionar en el botón “confirmar” seguido del formulario de reserva, aparece un cartel confirmando la reserva con día y hora.	
Como cliente, quiero ver un cartel que confirme mi reserva después de agendar, para asegurarme de que el turno fue registrado correctamente.	
<p><u>CA #9: Reserva/validación correcta</u> Dado un cliente que completa correctamente todos los campos del formulario de reserva. Cuando confirma la solicitud haciendo click en el botón “reservar”. Entonces se muestra un cartel con el mensaje “Reserva agendada correctamente”.</p> <p><u>CA #9: Reserva / validación incorrecta.</u> Dado un cliente que no completa correctamente uno o más campos obligatorios del formulario de reserva. Cuando intenta confirmar la solicitud. Entonces se muestra un cartel con el mensaje del error correspondiente, esto es:</p> <p>Error en el nombre = “Error en el nombre”</p> <p>Error en el teléfono = “Número inválido”</p> <p>Error en el email = “Email inválido”</p> <p>Servicio no seleccionado = “Debe seleccionar un servicio”</p> <p>Hora no seleccionada = “Debe seleccionar una hora”.</p>	

HISTORIA DE USUARIO	
Número: 4	Usuario: Cliente
Nombre Historia: Servicios realizados.	
DoR: Se cuenta con imágenes de trabajos realizados y descripciones proporcionadas por el cliente o el equipo de diseño.	DoD: La sección muestra correctamente una galería con imágenes y una breve descripción de cada servicio. Se visualiza bien en escritorio y móvil.
Esfuerzo: Fibonacci: 2	Priorización: 6
Descripción: En la sección “servicios realizados” se podrán ver imágenes con trabajos realizados por la barbería con una pequeña descripción.	
Como cliente, quiero ver imágenes de servicios anteriores realizados por la barbería, para tener una idea de los resultados que puedo esperar y sentir mayor confianza al reservar.	
CA #10: Galería de servicios anteriores. Dado un cliente de la barbería. Cuando accede a la sección de “Galería”. Entonces puede ver imágenes de trabajos anteriores realizados por la barbería, dispuestas una debajo de la otra, mostrando tres imágenes por pantalla.	

HISTORIA DE USUARIO	
Número: 5	Usuario: Cliente
Nombre Historia: Barberos	
DoR: Se cuenta con imágenes de cada barbero, sus nombres y textos breves con su trayectoria. El diseño de la sección fue aprobado.	DoD: La sección muestra correctamente una tarjeta por barbero con su imagen, nombre completo y una breve descripción profesional.
Esfuerzo: Fibonacci: 2	Priorización: 5
Descripción: En la sección “Barberos” se podrán ver imágenes de los barberos que trabajan en el local, su nombre completo y una pequeña descripción de su trayectoria profesional.	

Como cliente, **quiero** conocer a los barberos disponibles en la barbería, **para** poder elegir el que más se ajuste a mis preferencias y sentirme más seguro con mi reserva.

CA #11: Sección Barberos. Dado un cliente de la barbería. Cuando accede a la sección “Barberos”. Entonces puede visualizar imágenes de los barberos que trabajan en el local, junto con su nombre completo y una breve descripción de su trayectoria profesional.

HISTORIA DE USUARIO	
Número: 6	Usuario: Cliente
Nombre Historia: Precios	
DoR: Se definieron los servicios y sus precios. Se cuenta con el diseño aprobado de la tabla y se acordó el mensaje de forma de pago.	DoD: La sección muestra correctamente una tabla con nombre del servicio, descripción y precio. Debajo figura el texto “Solo se acepta efectivo”. Pasar el testing.
Esfuerzo Fibonacci: 2	Priorización: 7
Descripción: En la sección “Precios” se podrán ver todos los precios de los servicios prestados por el local, especificando el tipo de servicio y su respectivo precio.	
Como cliente, quiero consultar una tabla con los precios y servicios ofrecidos por la barbería, para saber cuánto cuesta cada servicio antes de reservar, y conocer el método de pago.	
CA #12: Tabla de precios y servicios. Dado un cliente de la barbería. Cuando accede a la sección “Precios”. Entonces puede consultar una tabla con los servicios ofrecidos, el precio de cada uno.	

HISTORIA DE USUARIO	
Número: 7	Usuario: Cliente
Nombre Historia: Contacto	
DoR: Se cuenta con los datos reales de contacto y se definió el mapa a utilizar (Google Maps con acceso a la API). Diseño de la sección aprobado.	DoD: Se visualizan correctamente los datos de contacto, incluyendo nota de días feriados. Mapa interactivo cargado y visible debajo de los datos. Pasar el testing.
Esfuerzo Fibonacci: 4	Priorización: 8
Descripción: En la sección “Contacto” se puede visualizar los datos de contacto: teléfono, horario (especificando que no se abre el local los días feriados no laborales), email, dirección y redes sociales, Debajo de estos datos un mapa interactivo con la ubicación de la barbería y seguido.	
Como cliente, quiero ver los datos de contacto y ubicación del local, para poder comunicarme o llegar fácilmente a la barbería.	
<p>CA #13: Información de contacto. Dado un cliente de la barbería. Cuando accede a la sección “Contacto”. Entonces puede visualizar los datos de contacto, incluyendo teléfono, email, dirección, horario de atención (especificando que no se abre los días feriados no laborables) mapa y redes sociales.</p> <p>CA #14: Mapa. Dado un cliente de la barbería. Cuando accede a la sección “Contacto”. Entonces puede ver un mapa de Google Maps con la ubicación de la barbería, el cual es interactivo y permite al usuario desplazarse dentro del mismo.</p>	

HISTORIA DE USUARIO	
Número: 8	Usuario: dueño Barberia / Socio
Nombre Historia: Login	
DoR: Credenciales usuario: admin, contraseña: Barberia.	Restricción: solo los administradores tienen acceso.
Esfuerzo Fibonacci: 5	Priorización: 2
<p>Descripción: Solo los administradores de la barbería pueden ingresar con usuario y contraseña. Estas credenciales serán: usuario: Admin, contraseña: Barbería.</p> <p>Al ingresar podrán ver únicamente las citas agendadas para el día corriente.</p>	
<p>Como dueño de la barbería, quiero ingresar al sistema con mis credenciales, para poder ver las citas agendadas del día y así organizar la jornada laboral.</p>	
<p>CA #15: Inicio de sesión exitoso. Dado el dueño/socio de la barbería. Cuando ingresa sus credenciales correctamente en el sistema (usuario “admin” y contraseña “Barberia”). Entonces puede acceder al panel donde visualiza las citas agendadas del día corriente.</p> <p>CA #16: Inicio de sesión fallido. Dado el dueño de la barbería. Cuando ingresa credenciales incorrectas en el sistema. Entonces se muestra un mensaje de error que indica que los datos ingresados son inválidos y se solicita que los verifique para volver a intentar.</p>	

VI. Priorización de requerimientos

Must Have



Should Have



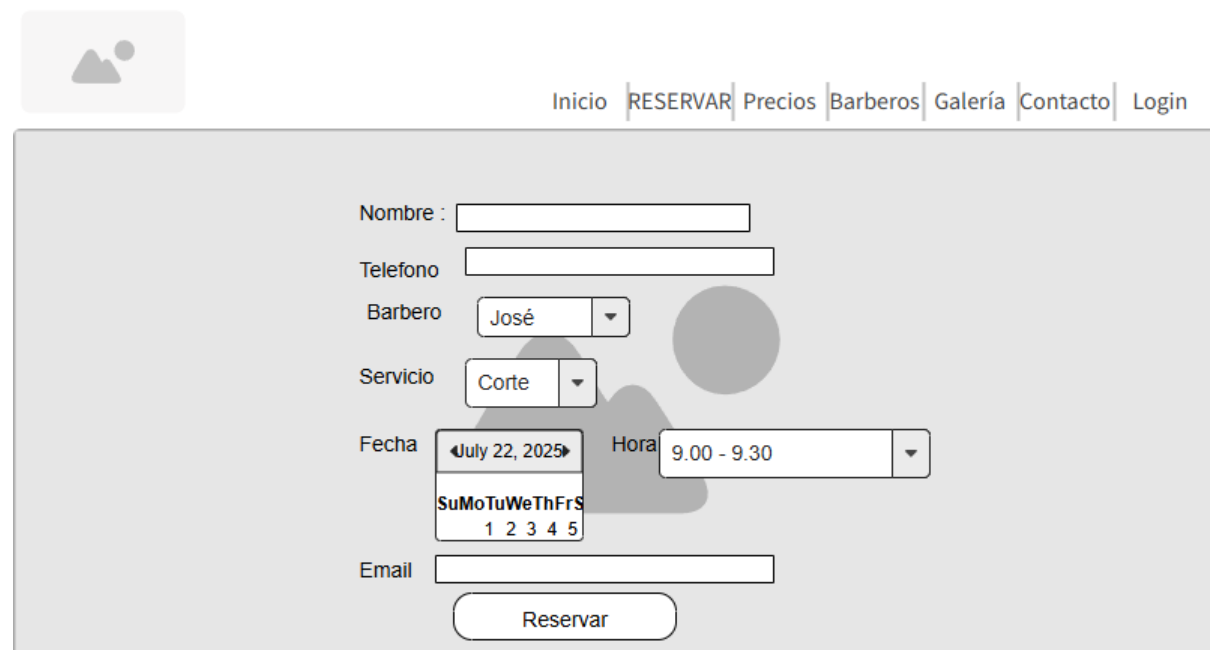
Could Have



Won't Have



VII. Mockups

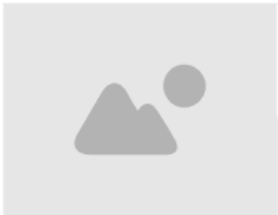




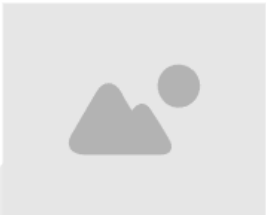
Servicio	Precio
Corte común	\$550
Color	\$800



Barbero 1



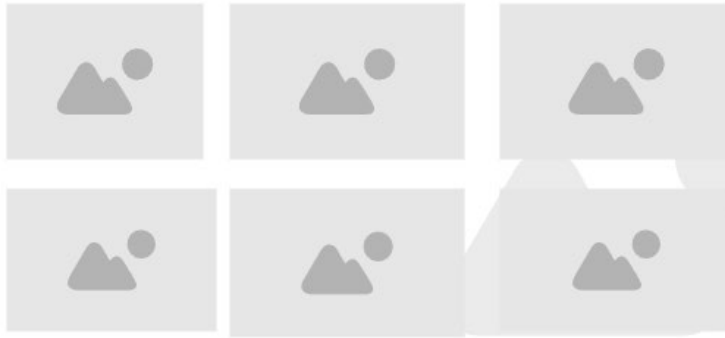
Barbero 2



Barbero 3

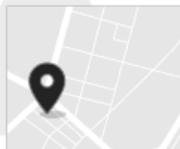


[Inicio](#) | [Reservar](#) | [Precios](#) | [Barberos](#) | [GALERIA](#) | [Contacto](#) | [Login](#)



[Inicio](#) | [Reservar](#) | [Precios](#) | [Barberos](#) | [Galería](#) | [CONTACTO](#) | [Login](#)

Dirección
Teléfono
Horarios
Días Cerrados





[Inicio](#) | [Reservar](#) | [Precios](#) | [Barberos](#) | [Galería](#) | [Contacto](#) | [LOGIN](#)

Usuario

Contraseña

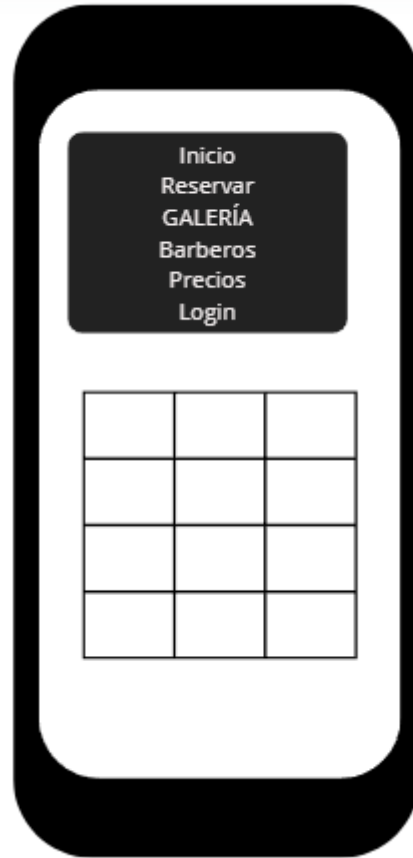
Ingresar



[Inicio](#) | [Reservar](#) | [Precios](#) | [Barberos](#) | [Galería](#) | [Contacto](#) | [LOGIN](#)

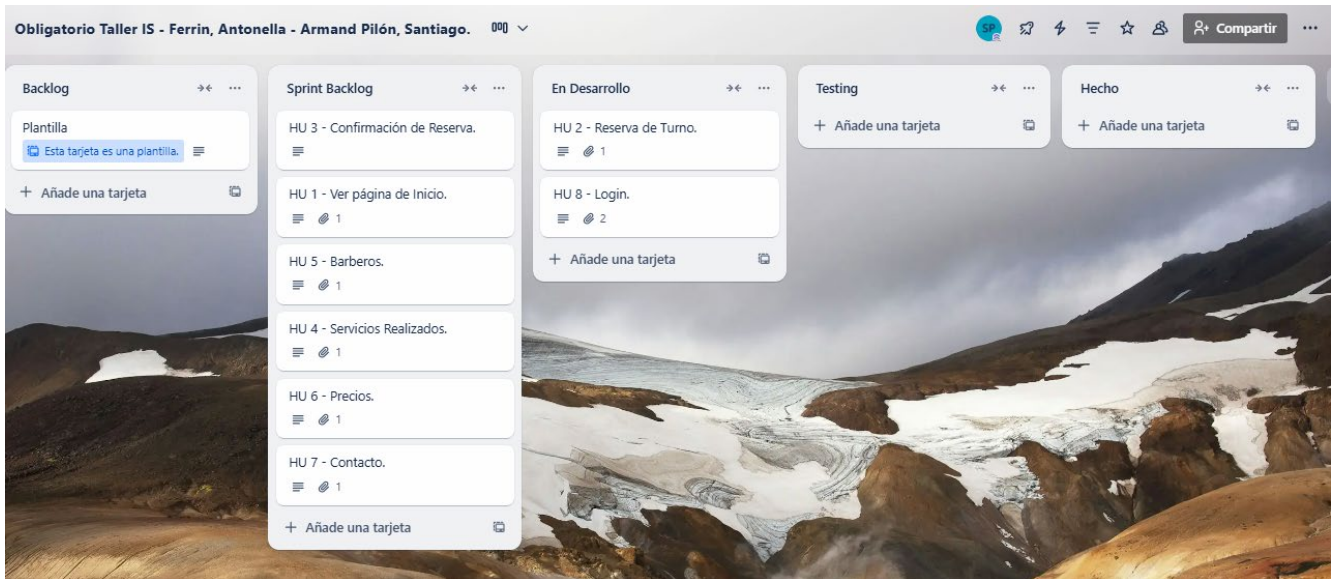
AGENDA

CLIENTE : SERVICIO : FECHA: BARBERO: HORA:

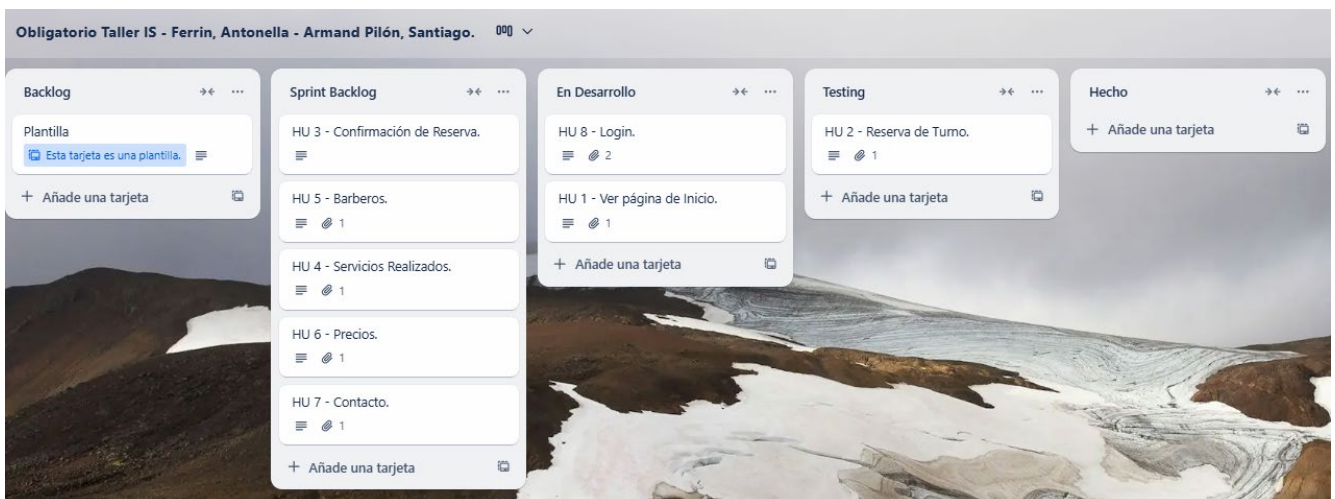


VIII. Trello

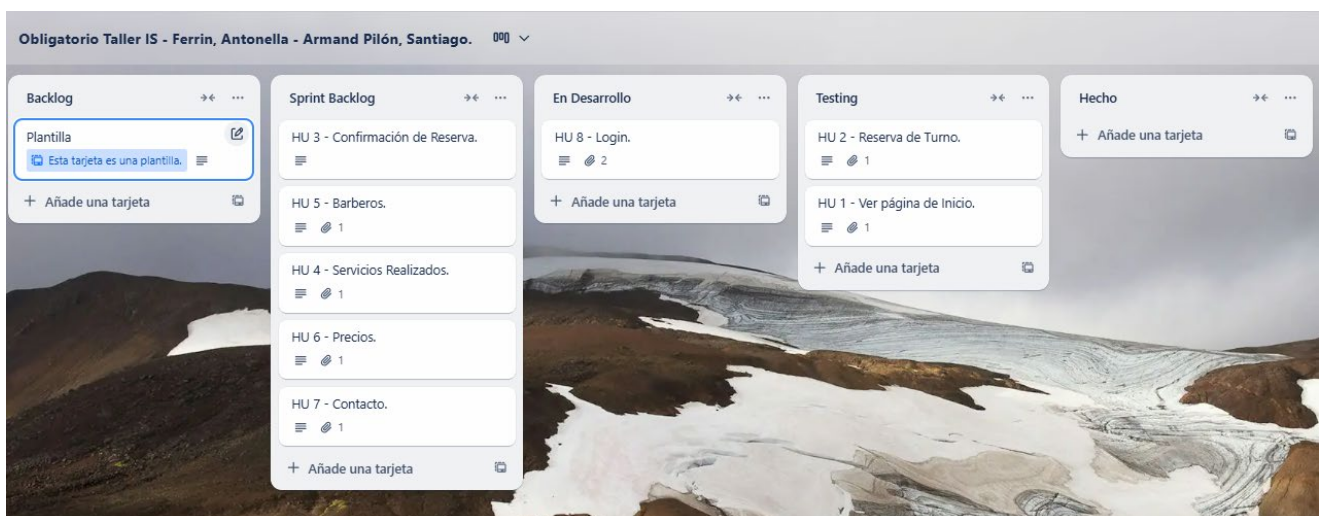
Luego de la Daily Meeting - 24-07-2025.



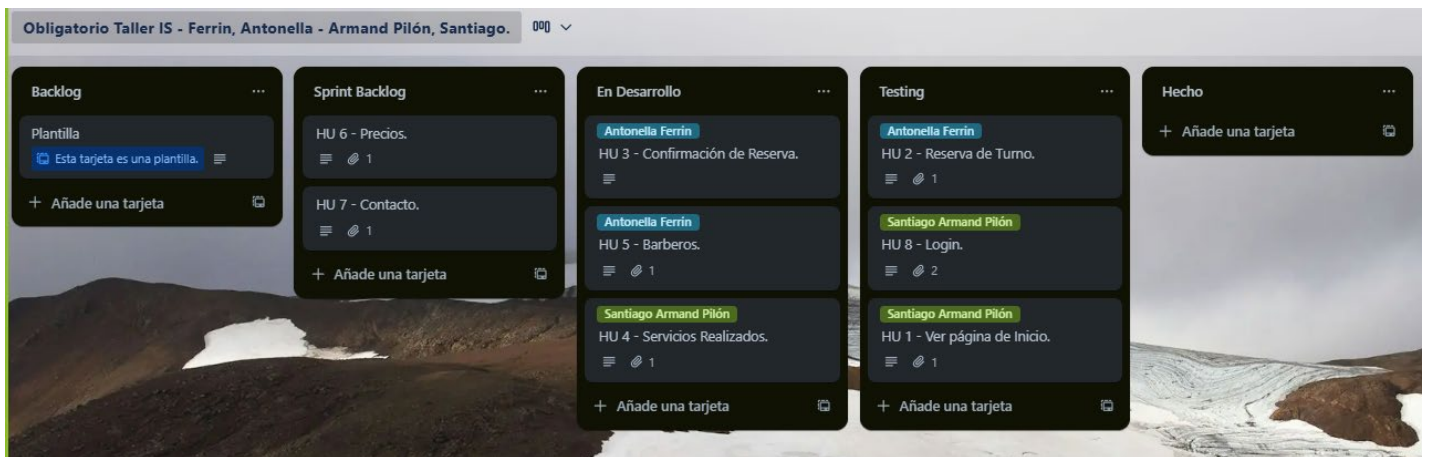
Durante la clase - 24-07-2025.



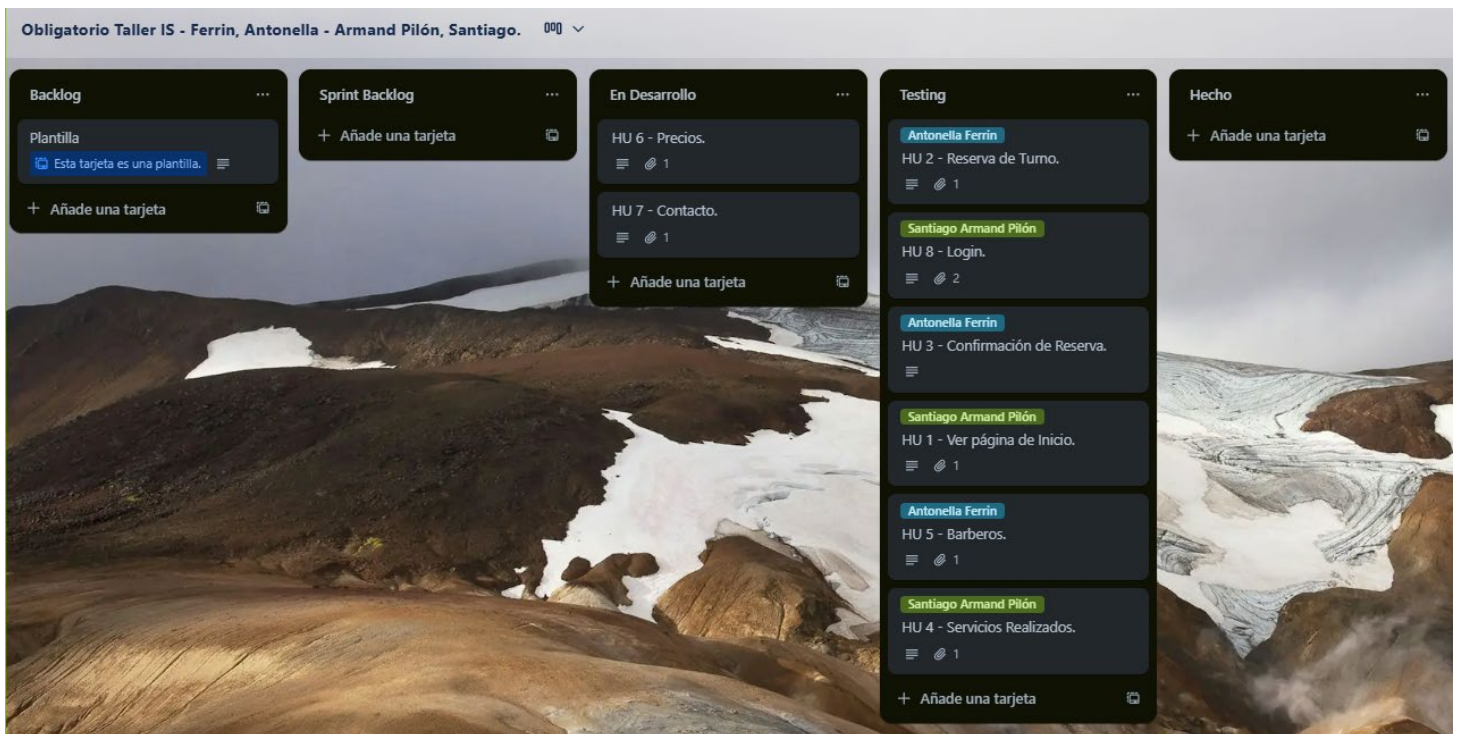
Al finalizar la clase - 24-07-2025.



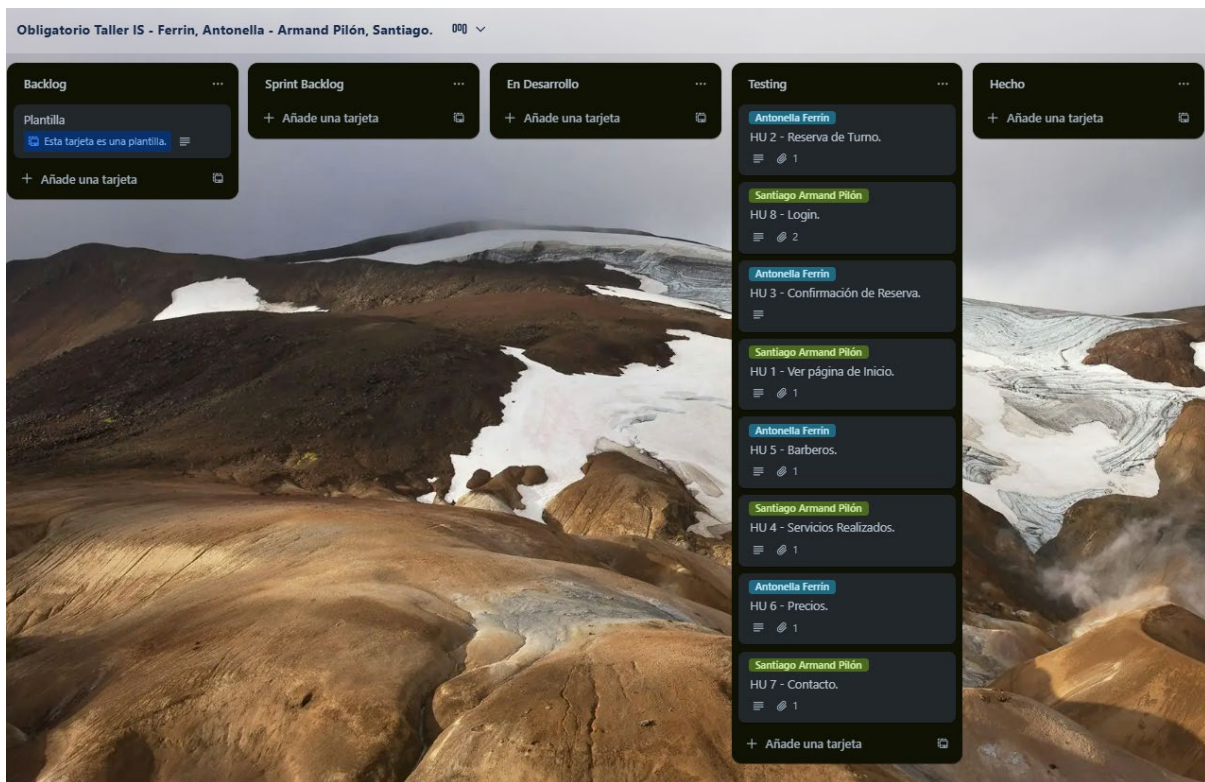
Luego Daily Meeting – 29-07-2025.



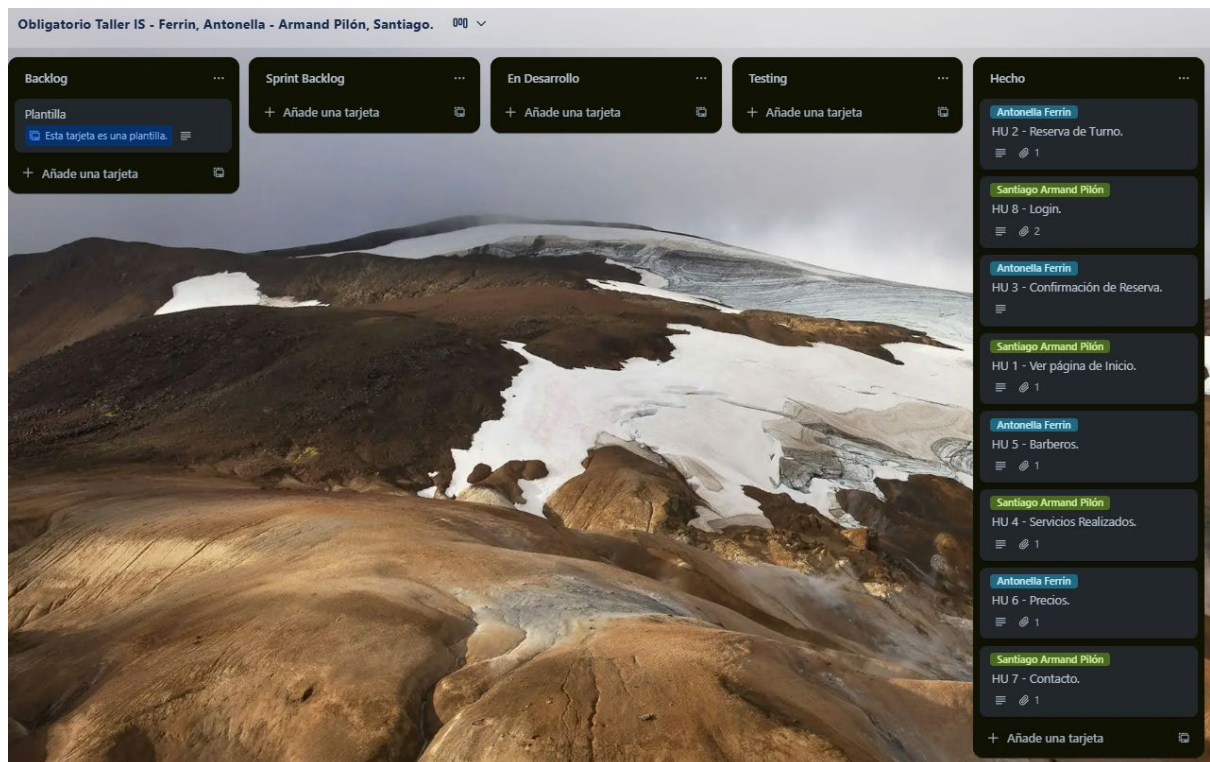
Al finalizar la clase - 29-07-2025.



Luego de Daily Meeting - 31-07-2025.

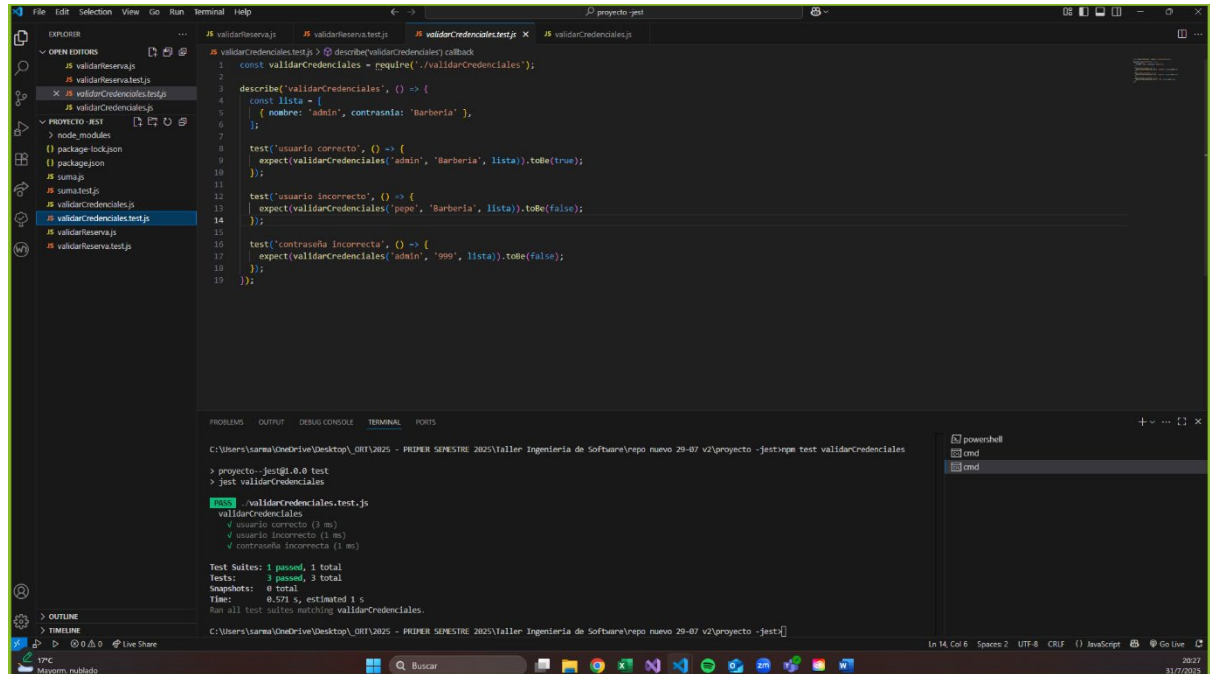


Al finalizar clase - 31-07-2025.



IX. Pruebas Unitarias JEST

Test Login

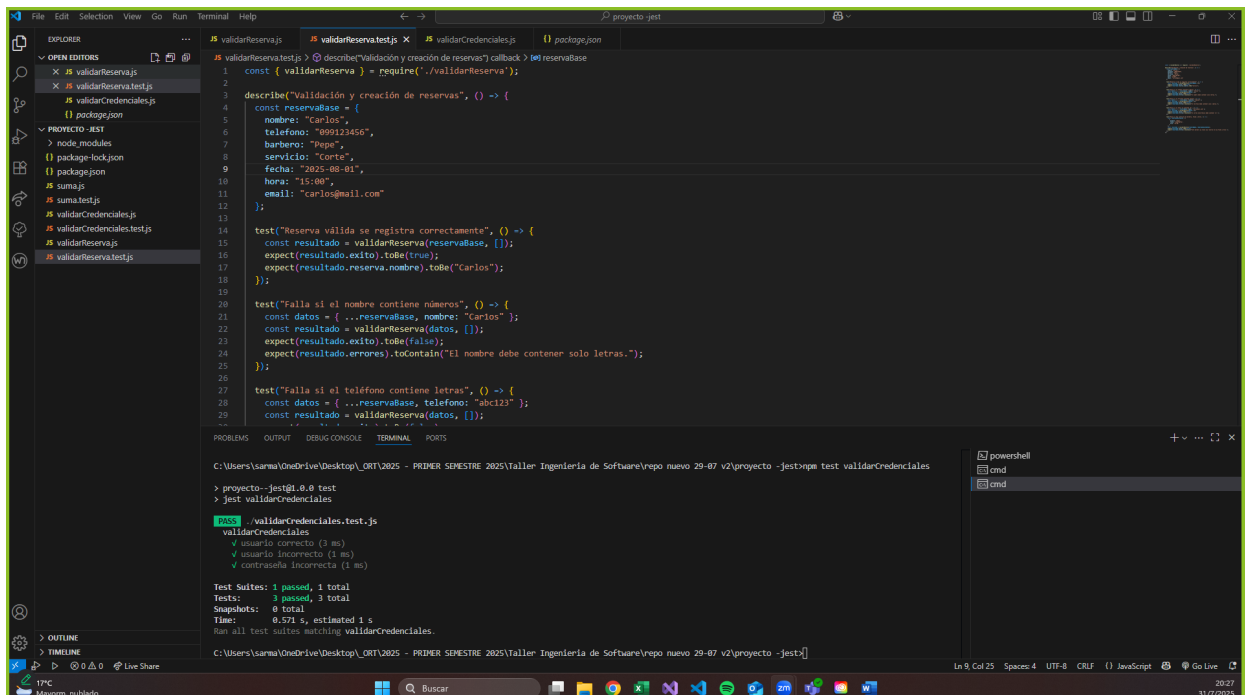


The screenshot shows a VS Code editor with a project named 'proyecto-jest'. The file explorer on the left shows the project structure, including 'validarCredenciales.test.js'. The main editor displays the test file content, which includes a 'describe' block for 'validarCredenciales' and three 'test' blocks for 'usuario correcto', 'usuario incorrecto', and 'contraseña incorrecta'. The terminal at the bottom shows the command 'projecto-jest --run test validarCredenciales' and the output of the Jest test runner, which indicates that all tests passed.

```
1 const validarCredenciales = require('./validarCredenciales');
2
3 describe('validarCredenciales', () => {
4   const lista = [
5     { nombre: 'admin', contraseña: 'Barberia' },
6   ];
7
8   test('usuario correcto', () => {
9     expect(validarCredenciales('admin', 'Barberia', lista)).toBe(true);
10   });
11
12   test('usuario incorrecto', () => {
13     expect(validarCredenciales('pepe', 'Barberia', lista)).toBe(false);
14   });
15
16   test('contraseña incorrecta', () => {
17     expect(validarCredenciales('admin', '999', lista)).toBe(false);
18   });
19 });
```

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 0.571 s, estimated 1 s
Run all test suites matching validarCredenciales.

Test Realizar Reserva

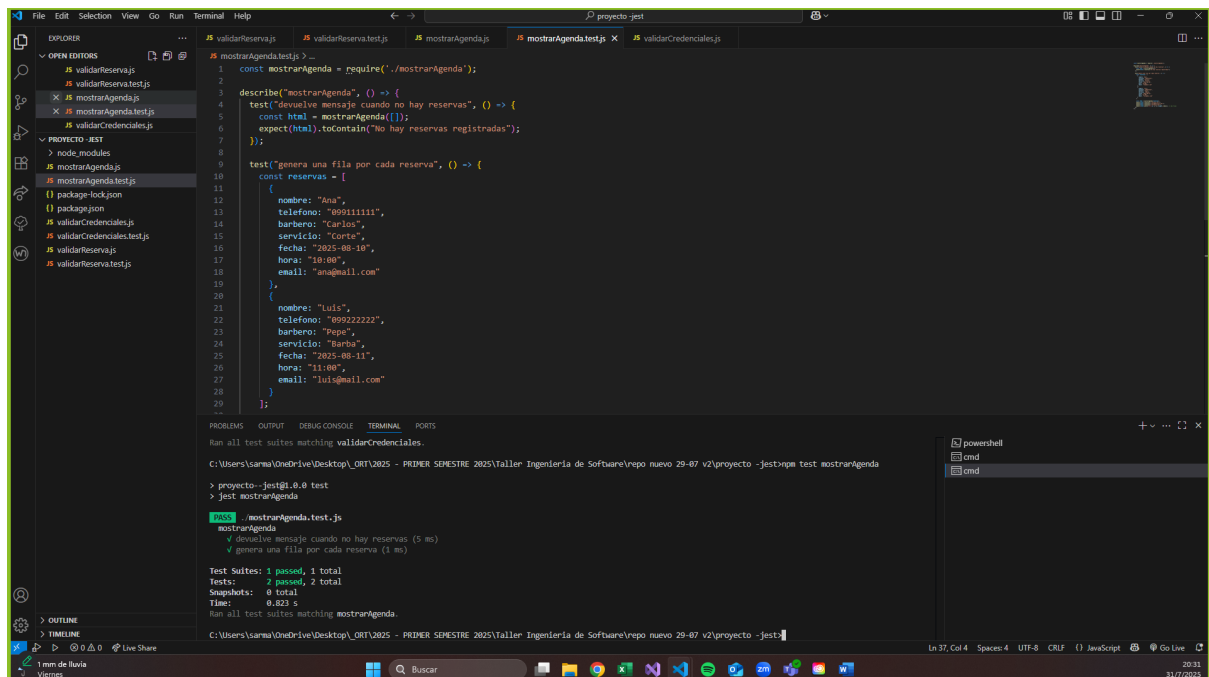


The screenshot shows a VS Code editor with a project named 'proyecto-jest'. The file explorer on the left shows the project structure, including 'validarReserva.test.js'. The main editor displays the test file content, which includes a 'describe' block for 'validación y creación de reservas' and three 'test' blocks for 'Reserva válida se registra correctamente', 'Falla si el nombre contiene números', and 'Falla si el teléfono contiene letras'. The terminal at the bottom shows the command 'projecto-jest --run test validarCredenciales' and the output of the Jest test runner, which indicates that all tests passed.

```
1 const { validarReserva } = require('./validarReserva');
2
3 describe('Validación y creación de reservas', () => {
4   const reservaBase = {
5     nombre: 'Carlos',
6     telefono: '899123456',
7     barbero: 'Pepe',
8     servicio: 'Corte',
9     fecha: '2025-08-01',
10    hora: '15:00',
11    email: 'carlos@mail.com'
12  };
13
14  test('Reserva válida se registra correctamente', () => {
15    const resultado = validarReserva(reservaBase, []);
16    expect(resultado.exito).toBe(true);
17    expect(resultado.reserva.nombre).toBe('Carlos');
18  });
19
20  test('Falla si el nombre contiene números', () => {
21    const datos = { ...reservaBase, nombre: 'car1os' };
22    const resultado = validarReserva(datos, []);
23    expect(resultado.exito).toBe(false);
24    expect(resultado.errores).toContain('El nombre debe contener solo letras.');
```

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 0.571 s, estimated 1 s
Run all test suites matching validarCredenciales.

Test Mostrar Reservas del Día



The screenshot shows a Visual Studio Code editor with a project named 'proyecto-jest'. The Explorer sidebar on the left shows the file structure with several JavaScript files, including 'mostrarAgenda.test.js'. The main editor displays the content of 'mostrarAgenda.test.js', which contains Jest tests for a function named 'mostrarAgenda'. The tests include a 'describe' block for the function and a 'test' block for a specific scenario. The test file is as follows:

```
1 const mostrarAgenda = require('./mostrarAgenda');
2
3 describe('mostrarAgenda', () => {
4   test('devuelve mensaje cuando no hay reservas', () => {
5     const html = mostrarAgenda();
6     expect(html).toContain("No hay reservas registradas");
7   });
8
9   test('genera una fila por cada reserva', () => {
10    const reservas = [
11      {
12        nombre: "Ana",
13        telefono: "0991111111",
14        barbero: "Carlos",
15        servicio: "Corte",
16        fecha: "2025-08-10",
17        hora: "10:00",
18        email: "ana@mail.com"
19      },
20      {
21        nombre: "Luis",
22        telefono: "0992222222",
23        barbero: "Pepe",
24        servicio: "Tumba",
25        fecha: "2025-08-11",
26        hora: "11:00",
27        email: "luis@mail.com"
28      }
29    ];
30  });
31 });
```

Below the editor, the 'TERMINAL' panel shows the command 'projecto-jest@0.0.0 test' and the output of the Jest test runner. The output indicates that the tests passed successfully. The terminal output is as follows:

```
Run all test suites matching validarCredenciales.
C:\Users\sama\OneDrive\Desktop\ORT\2025 - PRIMER SEMESTRE 2025\Taller Ingenieria de Software\repo nuevo 29-07 v2\proyecto -jest>npm test mostrarAgenda
> proyecto-jest@0.0.0 test
> jest mostrarAgenda

PASS ./mostrarAgenda.test.js
  mostrarAgenda
    ✓ devuelve mensaje cuando no hay reservas (5 ms)
    ✓ genera una fila por cada reserva (1 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 0.023 s
Run all test suites matching mostrarAgenda.
C:\Users\sama\OneDrive\Desktop\ORT\2025 - PRIMER SEMESTRE 2025\Taller Ingenieria de Software\repo nuevo 29-07 v2\proyecto -jest>
```

The status bar at the bottom of the editor shows the file encoding as UTF-8, the line and column position as Ln 37, Col 4, and the current date and time as 20:31 31/07/2025.