Obligatorio

Taller de Ingeniería de Software

Un hombre con lentes y camiseta gris

El contenido generado por IA puede ser incorrecto.A person with curly hair and beards

AI-generated content may be incorrect.

Matías Pietrafesa

N.º 318802

Matías Oreiro

N.º 239479

Francisco Arenas

N.º 332800

Docente: Sebastián Pombo

Carrera: Analista en Tecnologías de la información

Grupo: M3.5A

Contents

[Entrevista 2](#_Toc828011812)

[Requerimientos 7](#_Toc887577956)

[HU-01: Reserva de Turno 7](#_Toc572444129)

[HU-02: Login de Administradores 10](#_Toc1261596454)

[HU-03: Listado de Reservas (Solo Lectura Administrador) 12](#_Toc381394062)

[HU-04: Landing Page 14](#_Toc1818320778)

[HU-05: Visualización de Servicios 15](#_Toc1696149061)

[HU-06: Sección Equipo de Barberos 17](#_Toc1119151106)

[HU-07: Sección Contacto 19](#_Toc381772328)

[HU-08: Galería de Fotos de Trabajos Realizados 20](#_Toc108081000)

[Trello 23](#_Toc337012575)

## Entrevista

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | PREGUNTA | RESPUESTA | ENTREVISTADO | ENTREVISTADOR | FECHA |
| 0 | ¿Cuáles son los horarios y días que abren? | Los 7 días de la semana. Horario habitual 9:00 a 21:00 |  |  | 7/15/2025 |
| 1 | ¿Cuántos barberos trabajan? | 6 (seis). Quiere una foto de cada uno para identificación.  Todo el equipo está capacitado para brindar todos los servicios. |  |  | 7/15/2025 |
| 2 | ¿Qué servicios brinda la barbería al público? | Corte de pelo, barba,  Colorimetría, lavado y acondicionamiento del pelo, servicio premium nutrición del pelo de vitaminas y minerales que favorecen la salud del cabello. |  |  | 7/15/2025 |
| 3 | ¿Qué requisitos debe cumplir una reserva para ser valida? | El teléfono y el correo electrónico deben estar correctamente cargados. La fecha debe ser desde el día en adelante. Los servicios duran media hora, debe seleccionar el horario y el barbero. Debe tener nombre.  Guardar en LocalStorage(json) con la información necesaria para el listado. |  |  | 7/15/2025 |
| 4 | El sistema debe gestionar login? | No, solo un formulario, con datos. |  |  | 7/15/2025 |
| 5 | ¿Debe contar con un sistema de roles? ¿O es exclusiva para el uso de clientes? | El administrador debe tener login y debe poder ver las reservas. (cliente, día, hora, servicio y barbero).  Los barberos no tienen acceso al sistema y los clientes no deben contar con login. |  |  | 7/15/2025 |
| 6 | ¿El diseño debe ser adaptable a distintos dispositivos? | Quiere que sea una aplicación responsive, que funcione tanto en la computadora como en el celular |  |  | 7/15/2025 |
| 7 | ¿Como quieres que sea el estilo visual de la página? | Un estilo hípster, enfocado a un público de ese estilo |  |  | 7/15/2025 |
| 8 | Tienes una paleta de color o un logo distintivo? | Colores opacos oscuros, nada estridente. La barbería cuenta con un logo. |  |  | 7/15/2025 |
| 9 | Luego de confirmada una reserva ¿qué posibilidad tendrá el cliente de cancelarla o cambiar el horario? | Los clientes no tienen posibilidad de cancelar una reserva desde el sistema. |  |  | 7/15/2025 |
| 10 | En caso de haber roles diferenciados. ¿Qué funcionalidades deben estar disponibles para cada uno? | Solo admin (pregunta 5) |  |  |  |
| 11 | ¿Cuál es la funcionalidad que no puede fallar? | La reserva (cliente), el listado de reservas (admin), que sea responsive y navegación fluida. |  |  |  |
| 12 | ¿Como quieres que luzca la página principal de la web? | Quiere un botón claro que dirija a la reserva.  Quiere tener al inicio una sección "sobre nosotros" que haga énfasis en sus 15 años de experiencia.  Además, quiere tener un lugar donde se vean las redes, una sección de contacto con la ubicación y los datos (tel, dirección, mapa, como llegar).  También una galería de trabajos hechos, los servicios, los barberos. |  |  |  |
| 13 | Tienes algún sitio web similar del que quieras dar alguna referencia? | Pide que revisemos nosotros. Quiere que sea atractiva. |  |  |  |
| 14 | ¿Como contactan actualmente sus clientes para reservar? | Los clientes llaman por teléfono, no hay ninguna solución tecnológica implementada. |  |  |  |
| 15 | ¿Hay alguna restricción tecnológica? | Js, html y css  Todo vanilla. |  |  |  |
| 16 | ¿Querés tener una sección interna donde puedas ver y administrar todas las reservas? | (pregunta 5) |  |  |  |
| 17 | ¿Cuál es el plazo aproximado que tardara el desarrollo de la web? |  |  |  |  |
| 18 | ¿Hay algo que *no* querés que tenga la aplicación? | Sin login en el lado del cliente, sin pagos, sin productos a la venta. |  |  |  |
| 19 | ¿Qué datos personales recogerán de los clientes? | Nombre, apellido, celular y mail. |  |  |  |
| 20 | ¿El sistema debe gestionar los pagos? | No, el pago se gestiona en la barbería |  |  |  |
| 21 | En caso de poder cancelar o posponer una reserva ¿qué requisitos debe cumplir esta modificación? | No hay cancelación |  |  |  |
| 22 |  |  |  |  |  |
| 23 |  |  |  |  |  |

Preguntas hechas por otros equipos

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Pregunta | Respuesta | Equipo |
|  | ¿Todos los barberos ofrecen los mismos servicios? | Si |  |
|  | ¿Hay un rango de edad entre los clientes? | No |  |
|  | ¿Cuánto dura un servicio? | 30 minutos. |  |
|  | ¿Debemos tener un sistema de calificaciones por parte de los clientes? | No |  |
|  | ¿Los precios son todos iguales? | No, son distintos y deben estar visibles a la hora de hacer una reserva |  |
|  | ¿Los servicios y los barberos deben tener una sección aparte? | Si, los servicios con descripción y precio y los barberos con imagen. |  |
|  | ¿Como se deben guardar los servicios? | Desde un array no deben ser estáticos. |  |
|  |  | Los barberos deben estar precargados en un array también |  |
|  | Sección de contactos y redes sociales. | Teléfono, mapa de cómo llegar, redes |  |
|  | ¿Recordatorio? | Solo un mansaje en pantalla de que la reserva quedo hecha. |  |
|  | ¿Shop de productos? | Tiene, pero no quiere incluirlo en la implementación por ahora. |  |
|  |  | Guardar la data en JSON local storage |  |
|  |  | Agregar un párrafo de introducción del local y sus 15 años de exp. |  |
|  | Pantalla de inicio | Debe haber un botón que redirija a la reserva, y una muestra de su trabajo, sus barberos y sus redes sociales y teléfonos |  |
|  |  | Fotos de calidad 1920x1080px.  Los trabajos hechos de 800x600.  Los barberos de 600x600. |  |
|  |  | Efecto hover, sombra y elevación.  En los trabajos con zoom ligero. (light box para ampliar el grid responsive) |  |

## Requerimientos

* Reservas de turno (nombre, apellido, celular, correo, barbero, servicio, fecha)
  + 30 min
  + Ver disponibilidad
  + Mensaje de reserva realizada
  + Almacenado en local storage
* Landing
* Contacto (email, redes)
* Mapa
* Sección equipo de barberos
* Sección de servicios
* Admin gestionar reservas
* Login Admin
* Galería de fotos de trabajos realizados
* Responsive
* Profesional, funcional, intuitiva, navegación fluida
* Tamaño imagenes
  + barberos 600x600 con efecto sombra y elevación
  + trabajos 800x600 con zoom ligero
  + las demás 1920x1080

### HU-01: Reserva de Turno

**User Story**  
 Como cliente de la barbería,  
 quiero poder reservar un turno indicando mis datos, el barbero, el servicio y la fecha,  
 para asegurarme un horario de atención sin complicaciones.

**Descripción**  
 Esta funcionalidad permite a los usuarios reservar un turno en línea desde la aplicación web. El formulario solicita nombre, apellido, celular, correo electrónico, barbero, servicio y fecha. El horario debe seleccionarse en el formulario que muestra los turnos de 30 minutos que están disponibles para ese barbero en esa fecha. Los turnos comienzan todos los días a las 9:00 y terminan a las 21:00. Al confirmar, se muestra un mensaje de reserva exitosa y los datos se almacenan en el local storage para mantener la información en el navegador del cliente.  
 Además, el sistema debe validar la disponibilidad del barbero en el día y horario elegido para evitar superposiciones.

**Criterios de Aceptación**

* **CA1**  
   Dado que el usuario accede a la sección de reserva,  
   cuando completa el formulario con todos los datos requeridos y elige un horario disponible,  
   entonces se muestra un mensaje de reserva exitosa y los datos se almacenan en local storage.
* **CA2**  
   Dado que el usuario elige un barbero,  
   cuando el barbero no tiene horarios disponibles para esa fecha,  
   entonces se debe mostrar un mensaje de que no hay disponibilidad.
* **CA3**  
   Dado que el usuario no completa uno o más campos obligatorios,  
   cuando intenta enviar el formulario,  
   entonces se muestran mensajes de validación claros por cada campo incompleto.
* **CA4**  
   Dado que el usuario ya hizo una reserva,  
   cuando recarga la página,  
   entonces sus datos deben seguir disponibles en el local storage.

**Scope / Out of Scope**

* **In Scope:**
  + Formulario de reserva con validaciones.
  + Verificación de disponibilidad por barbero, fecha y horario.
  + Mensaje de éxito al confirmar reserva.
  + Almacenamiento de datos en local storage del navegador.
* **Out of Scope:**
  + Envío de correos electrónicos o SMS de confirmación.
  + Historial de reservas por usuario.

**Definition of Ready (DoR)**

* El formulario y sus campos requeridos están definidos.
* Se cuenta con estructura de datos de barberos, servicios y horarios disponibles (mock o real).
* El diseño del formulario está validado por el equipo de diseño.
* Se confirma el uso de local storage para persistencia básica de datos.
* Se definieron las reglas de validación y disponibilidad.

**Definition of Done (DoD)**

* El formulario fue implementado según el diseño aprobado.
* La lógica de verificación de disponibilidad está funcional.
* Los datos de la reserva se almacenan correctamente en local storage.
* Se muestra un mensaje de éxito al confirmar la reserva.
* Todos los criterios de aceptación fueron validados (manual o automáticamente).
* La funcionalidad fue revisada por QA y validada por el Product Owner.
* El código fue documentado e integrado en la rama principal del proyecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

### HU-02: Login de Administradores

**User Story**

**Como** administrador del sitio, **quiero** un formulario de login seguro, con validaciones claras y persistencia de sesión básica,  
**para** acceder al listado de reservas sin exponer la información a usuarios no autorizados.

**Descripción**

Esta funcionalidad permite a los administradores acceder a una sección privada del sistema mediante autenticación. El formulario solicita “usuario” y “contraseña”, mostrando mensajes claros de error en caso de datos inválidos o campos vacíos.

Al iniciar sesión correctamente, el sistema redirige al listado de reservas y guarda la sesión en localStorage para mantener el acceso mientras se navega (hasta que el usuario cierre sesión o la pestaña).

**Criterios de Aceptación**

1. Dado que un administrador accede a la URL de login,  
   cuando se carga la página,  
   entonces se muestra un formulario con los campos "Usuario" y "Contraseña" y un botón "Ingresar".
2. Dado que el administrador deja uno o ambos campos vacíos,  
   cuando intenta hacer clic en "Ingresar",  
   entonces se muestran mensajes de validación para los campos incompletos.
3. Dado que el administrador introduce credenciales válidas,  
   cuando presiona "Ingresar",  
   entonces es redirigido al listado de reservas y se guarda su sesión en localStorage.
4. Dado que el administrador introduce credenciales incorrectas,  
   cuando presiona "Ingresar",  
   entonces se muestra un mensaje de error ("Usuario o contraseña incorrectos") y el campo contraseña se limpia.
5. Dado que el administrador ya inició sesión previamente,  
   cuando vuelve a cargar o navegar entre páginas,  
   entonces el sistema lo mantiene logueado mientras no cierre sesión o borre el localStorage.

**Scope / Out of Scope**

* **In Scope:**
  + Formulario de login con validaciones.
  + Verificación de credenciales contra un conjunto mock predefinido.
  + Mensajes de error claros.
  + Persistencia de sesión en localStorage.
* **Out of Scope:**
  + Autenticación en servidor real.
  + Recuperación de contraseña vía correo electrónico.
  + Autenticación multifactor o por roles avanzados.

**Definition of Ready (DoR)**

* La historia fue priorizada en el backlog.
* Se definieron los mensajes esperados para cada error.
* Se establecieron las credenciales mock válidas para login (ej: usuario: admin, contraseña: 1234).
* Se validó que se usará localStorage para simular sesión activa.
* El diseño del formulario fue aprobado por el equipo de diseño.

**Definition of Done (DoD)**

* El formulario fue implementado según el diseño.
* Se validan correctamente campos vacíos y credenciales inválidas.
* Al hacer login exitoso, se redirige y se guarda sesión en localStorage.
* El mensaje de error aparece cuando corresponde, limpiando solo la contraseña.
* El sistema reconoce si el administrador ya está logueado y mantiene la sesión.
* Todo fue probado manualmente y aprobado por QA y PO.
* El código fue documentado e integrado en la rama principal.

### HU-03: Listado de Reservas (Solo Lectura Administrador)

**User Story**  
Como administrador  
quiero acceder a un listado completo de reservas con fecha, hora, cliente, contacto, barbero y servicio para visualizar toda la información de cada cita sin posibilidad de modificarla.

**Descripción**  
 Esta funcionalidad ofrece al administrador una tabla en modo solo lectura que muestra todas las reservas realizadas. Cada fila incluye los campos:

* **Fecha** (YYYY‑MM‑DD)
* **Hora** (HH:MM)
* **Cliente** (nombre completo)
* **Contacto** (teléfono o email)
* **Barbero** asignado
* **Servicio** reservado

No hay botones ni enlaces para editar, cancelar o posponer reservas: es una vista puramente informativa.

**Criterios de Aceptación**

* Dado que el administrador accede a la sección “Reservas”,  
   cuando carga la página,  
   entonces se muestran todas las reservas ordenadas por fecha y hora con las columnas Fecha, Hora, Cliente, Contacto, Barbero y Servicio.
* Dado que la tabla esté desplegada,  
   cuando el administrador visualiza las filas,  
   entonces no aparecen controles de edición, cancelación o reprogramación en ninguna fila.
* Dado que el administrador navega desde un dispositivo móvil,  
   cuando accede al listado,  
   entonces la tabla se adapta manteniendo la legibilidad y proporciones adecuadas.

**Scope / Out of Scope**  
 *In Scope:*

* Vista en tabla estática de reservas en modo solo lectura.
* Columnas Fecha, Hora, Cliente, Contacto, Barbero y Servicio.
* Paginación y diseño responsive.

*Out of Scope:*

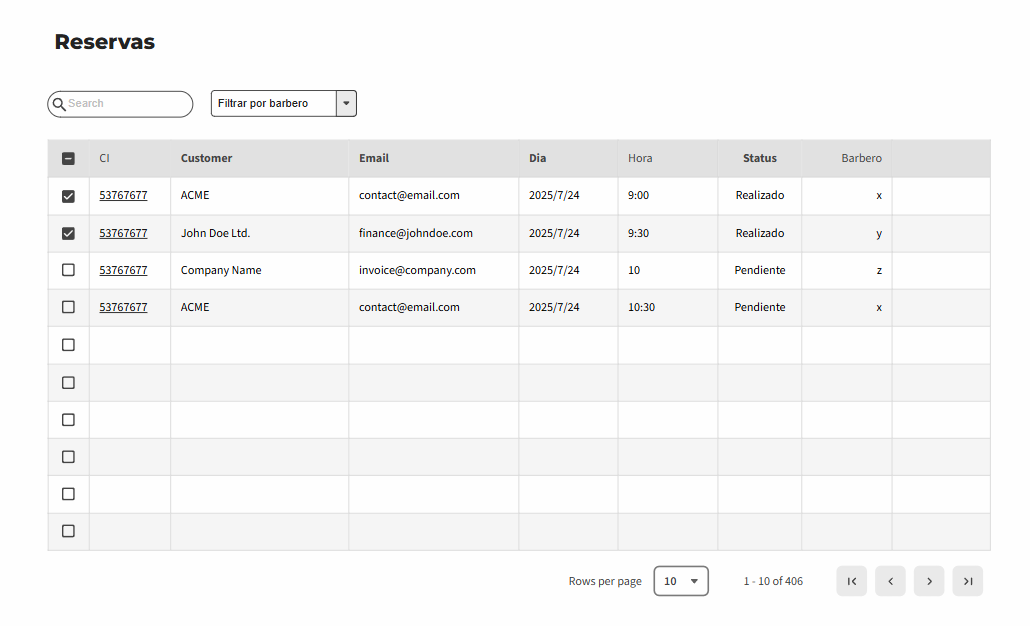
* Edición, cancelación o reprogramación de reservas.
* Exportación o descarga del listado.

**Definition of Ready (DoR)**

* Historia priorizada en el backlog.
* Roles y permisos de administrador definidos.
* Mockups o especificaciones de la tabla disponibles.

**Definition of Done (DoD)**

* Sección “Reservas” implementada y accesible solo para administradores.
* La tabla muestra correctamente todas las columnas y datos.
* No existen controles para modificar ninguna reserva.
* Paginación funciona y el diseño es responsive.
* Pruebas visuales y de usabilidad aprobadas en desktop y móvil.
* Código revisado, documentado e integrado en la rama principal.



### HU-04: Landing Page

**User Story**  
 Como visitante del sitio,  
 quiero ver una landing profesional, funcional e intuitiva,  
 para entender rápidamente los servicios y la identidad de la barbería.

**Descripción**  
 La landing page será la primera impresión del sitio, debe ser atractiva y fácil de navegar, adaptándose a dispositivos móviles (responsive). Incluirá elementos visuales acordes al branding y navegación fluida hacia otras secciones importantes. Esta debe contar con un botón de fácil acceso para que los clientes se dirijan a la reserva.

**Criterios de Aceptación**

* Dado que un usuario accede al sitio,  
   cuando visualiza la landing page,  
   entonces la página debe cargarse rápidamente y ser completamente responsive.
* Dado que el usuario navega por la landing,  
   cuando hace clic en los links del menú,  
   entonces la navegación es fluida y sin errores.
* Dado que la landing se muestra en dispositivos móviles y escritorio,  
   entonces el diseño y contenido se adaptan correctamente a cada pantalla.

**Scope / Out of Scope**

* In Scope: Diseño responsive, estructura básica, navegación fluida.
* Out of Scope: Animaciones avanzadas o contenido dinámico complejo.

**Definition of Ready (DoR)**

* Wireframes y diseño visual aprobados.
* Contenidos textuales e imágenes definidas.
* Reglas básicas de navegación acordadas.
* 

**Definition of Done (DoD)**

* Landing implementada y testeada en múltiples dispositivos y navegadores.
* Navegación entre secciones funciona correctamente.
* Diseño y contenido según especificación.
* Revisada y aprobada por QA y PO.

### HU-05: Visualización de Servicios

**User Story**

**Como** cliente de la barbería,  
**quiero** poder ver todos los servicios disponibles con su descripción,  
**para** entender qué opciones ofrece la barbería antes de reservar un turno.

**Descripción**

Esta funcionalidad permitirá al usuario acceder a una sección informativa donde se listan todos los servicios que ofrece la barbería "Estilo Clásico". Cada servicio debe mostrarse con su nombre y una breve descripción clara. El objetivo es que los clientes conozcan las opciones y puedan decidir con mayor confianza cuál reservar.

**Criterios de Aceptación**

1. **Dado** que el usuario accede a la sección “Servicios”,  
   **cuando** la visualiza,  
   **entonces** se muestran todos los servicios disponibles, cada uno con su nombre y descripción.
2. **Dado** que los servicios son fijos,  
   **cuando** se renderiza la página,  
   **entonces** se muestran los siguientes servicios:
   * **Corte de pelo**
   * **Barba**
   * **Colorimetría**
   * **Lavado y acondicionamiento del pelo**
   * **Servicio premium**: Nutrición del pelo con vitaminas y minerales que favorecen la salud del cabello.
3. **Dado** que el usuario está navegando desde un celular,  
   **cuando** entra a la sección,  
   **entonces** el diseño se adapta correctamente (responsive).

**Scope / Out of Scope**

* **In Scope**:
  + Página estática o dinámica con la lista de servicios.
  + Nombre y descripción de cada servicio.
  + Diseño responsive para desktop y mobile.
* **Out of Scope**:
  + Selección directa del servicio desde esta sección (eso ocurre en el formulario de reserva).
  + Cálculo de precios o tiempo por servicio.
  + Filtros, ordenamiento o edición de servicios.

**Definition of Ready (DoR)**

* La historia está priorizada en el backlog.
* Se definió el contenido textual de cada servicio.
* El equipo cuenta con mockups o guía visual aprobada.
* El diseño será responsive.
* Los criterios de aceptación están validados por el docente.

**Definition of Done (DoD)**

* Se implementó la interfaz con los servicios definidos.
* La información está visible correctamente en desktop y mobile.
* El contenido fue revisado ortográfica y semánticamente.
* El código fue probado y validado.
* La historia fue revisada por el equipo y aprobada por el PO.
* Fue integrada en la rama principal y documentada.

### HU-06: Sección Equipo de Barberos

**User Story**  
 Como cliente,  
 quiero ver una sección con el equipo de barberos,  
 para conocer al personal y elegir barbero para mi turno.

**Descripción**  
 Sección con fotos profesionales de cada barbero (tamaño 600x600 px), con efecto sombra y elevación para destacar, además de su nombre y rol. Debe ser responsive y mantener una estética coherente con el sitio.

**Criterios de Aceptación**

* Dado que el usuario accede a la sección,  
   cuando visualiza las fotos de los barberos,  
   entonces las imágenes se muestran en tamaño 600x600 con sombra y elevación.
* Dado que el usuario navega en móvil o escritorio,  
   entonces la sección mantiene buen diseño y legibilidad.
* Dado que el usuario ve la sección,  
   cuando pasa el mouse o toca,  
   entonces se mantiene la estética sin errores visuales.

**Scope / Out of Scope**

* In Scope: Fotos con efecto, nombres y roles, diseño responsive.
* Out of Scope: Videos o perfiles detallados.

**Definition of Ready (DoR)**

* Fotos y datos del equipo disponibles y aprobados.
* Diseño y estilo visual definidos.
* 

**Definition of Done (DoD)**

* Sección implementada según diseño.
* Efectos visuales funcionales y sin errores.
* Probada en distintos dispositivos.
* Aprobada por QA y PO.

### HU-07: Sección Contacto

**User Story**  
 Como cliente,  
 quiero acceder a una sección de contacto con email y redes sociales,  
 para comunicarme fácilmente con la barbería.

**Descripción**  
La sección incluirá datos de contacto como email, links a redes sociales y un mapa embebido para facilitar la ubicación. Debe ser responsiva y visualmente coherente.

**Criterios de Aceptación**

* Dado que el usuario visita la sección,  
   cuando ve el email y las redes sociales,  
   entonces los links funcionan correctamente.
* Dado que el usuario visualiza el mapa,  
   entonces se muestra correctamente embebido y es responsivo.
* La sección se adapta bien a todos los dispositivos.

**Scope / Out of Scope**

* In Scope: Email, redes sociales, mapa embebido.
* Out of Scope: Formularios de contacto avanzados o chat en vivo.

**Definition of Ready (DoR)**

* Datos de contacto confirmados.
* Diseño aprobado.

**Definition of Done (DoD)**

* Sección implementada, funcional y testeada.
* Aprobación QA y PO.

### HU-08: Galería de Fotos de Trabajos Realizados

**User Story**

**Como** cliente de la barbería,  
**quiero** ver una galería de imágenes de trabajos realizados,  
**para** conocer la calidad del servicio y decidir con mayor confianza.

**Descripción**

Esta funcionalidad permite al usuario visualizar una galería organizada de imágenes relacionadas con la barbería. La galería incluye:

* Fotos de trabajos realizados (800x600 px), con efecto de **zoom ligero** al pasar el cursor.
* Imágenes complementarias (ambiente, productos, etc.) en tamaño grande (1920x1080 px), a modo de presentación destacada.

La sección busca transmitir profesionalismo, calidad y confianza mediante una experiencia visual cuidada.

**Criterios de Aceptación**

1. **Dado** que el usuario accede a la sección “Galería”,  
   **cuando** visualiza las imágenes,  
   **entonces** se muestran agrupadas por tipo (barberos, trabajos, otras), cada una con su tamaño y efecto correspondiente.
2. **Dado** que se muestran imágenes de barberos,  
   **cuando** se cargan,  
   **entonces** tienen tamaño 600x600 px con sombra y efecto de elevación.
3. **Dado** que se muestran imágenes de trabajos,  
   **cuando** el usuario pasa el cursor,  
   **entonces** se aplica un efecto de **zoom ligero** y tienen tamaño 800x600 px.
4. **Dado** que se muestran imágenes generales (como del local),  
   **cuando** se renderiza la sección,  
   **entonces** se muestran en resolución grande de 1920x1080 px.
5. **Dado** que el usuario navega desde un dispositivo móvil,  
   **cuando** accede a la galería,  
   **entonces** las imágenes se adaptan manteniendo proporciones adecuadas.

**Scope / Out of Scope**

* **In Scope**:
  + Sección visual con carga de imágenes estáticas agrupadas.
  + Aplicación de efectos CSS específicos (zoom, sombra, elevación).
  + Diseño responsive.
* **Out of Scope**:
  + Carga dinámica de imágenes desde un servidor.
  + Galería con efecto slideshow automático o botones interactivos.
  + Subida de imágenes por parte de usuarios o administrador.

**Definition of Ready (DoR)**

* La historia está priorizada en el backlog.
* Se definieron las dimensiones y efectos CSS requeridos por tipo de imagen.
* Las imágenes de prueba (mock) están disponibles o referenciadas.
* El equipo entiende cómo aplicar los efectos visuales.
* Los criterios fueron validados con el docente.

**Definition of Done (DoD)**

* La galería fue implementada según diseño aprobado.
* Las imágenes tienen sus dimensiones y efectos correctamente aplicados.
* El diseño es responsive y funcional.
* La funcionalidad fue probada visualmente en distintos dispositivos.
* El código está documentado y revisado.
* Fue integrado a la rama principal del proyecto.

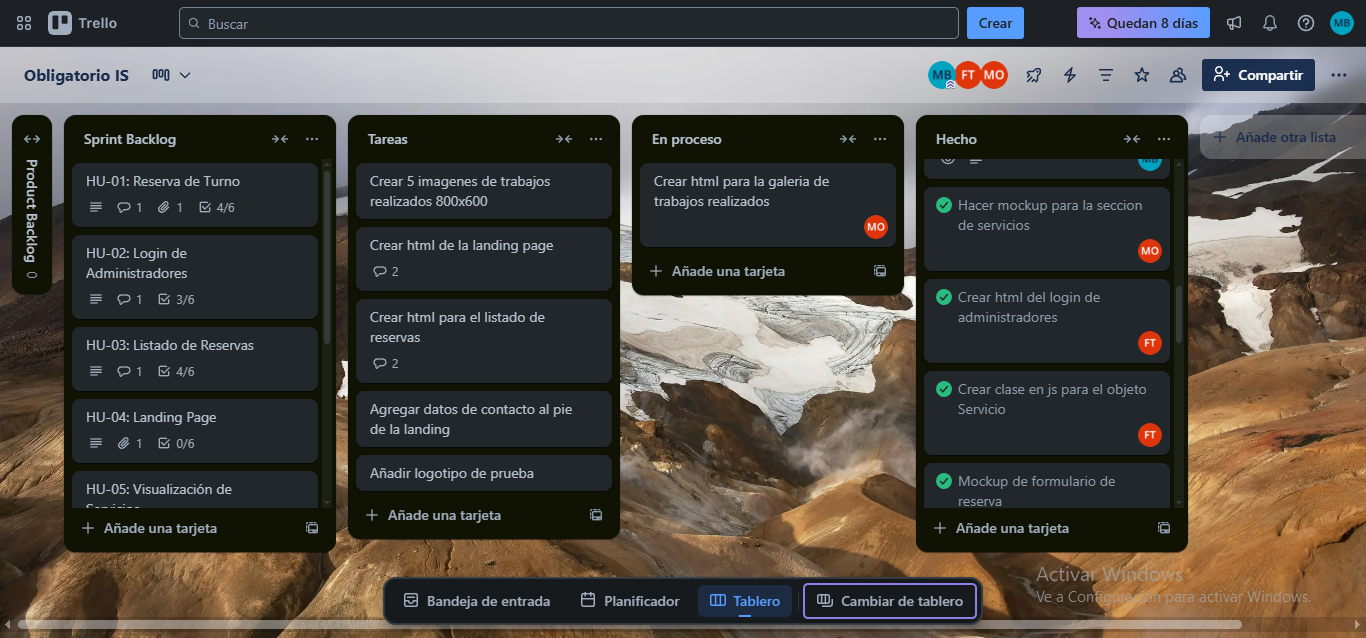
Una captura de pantalla de un celular

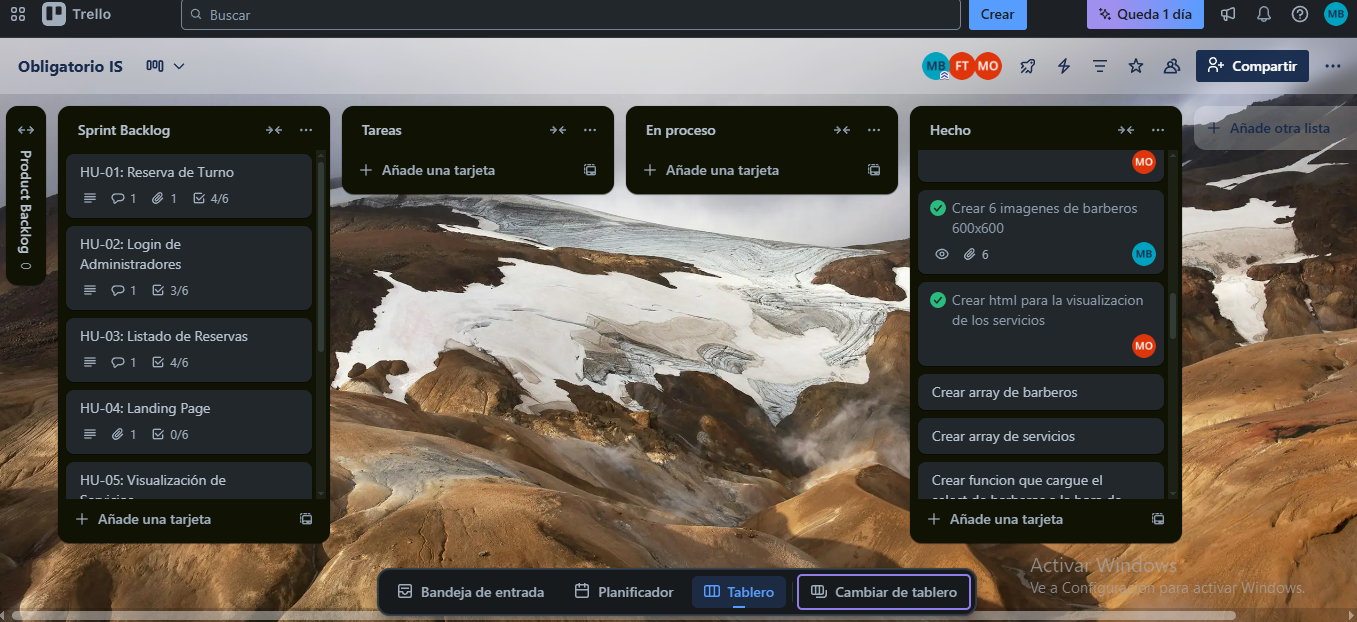
El contenido generado por IA puede ser incorrecto.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Correcto | Completo | Consistente | No ambiguo | Verificable | Factible | Observaciones |
| HU-1 |  |  |  |  |  |  | No se limitaron los horarios para la reserva… No aclara si se pueden reservar turnos consecutivos ni valida email o telefono |
| HU-2 |  |  |  |  |  |  | Faltaba informacion en el DoR. No se especificaba cuáles eran los casos de error, cuales eran los formatos de las credenciales y su persistencia en el localStorage. |
| HU-3 |  |  |  |  |  |  | No hay CA para paginación (nº de filas/página) ni para estado “sin reservas”. Falta definir orden predeterminado y comportamiento en caso de tabla vacía. |
| HU-4 |  |  |  |  |  |  | Falta definir métricas cuantitativas de carga (p.ej. < 2 s) y navegadores/dispositivos soportados. |
| HU-5 |  |  |  |  |  |  |  |
| HU-6 |  |  |  |  |  |  |  |
| HU-7 |  |  |  |  |  |  |  |
| HU-8 |  |  |  |  |  |  |  |

### Trello

<https://trello.com/b/qAzeUELE/obligatorio-is>





### Tests

import { renderBookings, loadSelects, Barbero, Servicio, Reserva, reserva } from './app.js';

test('loadSelects carga correctamente barberos y servicios', () => {

  document.body.innerHTML = `

    <select id="servicioSelect"></select>

    <select id="barberoSelect"></select>

  `;

  loadSelects();

  const servicioSelect = document.getElementById('servicioSelect');

  const barberoSelect = document.getElementById('barberoSelect');

  expect(servicioSelect.children.length).toBe(6);

  expect(servicioSelect.children[1].textContent).toBe('Corte ($500)');

  expect(barberoSelect.children.length).toBe(7);

  expect(barberoSelect.children[1].textContent).toBe('Joaquín Rojas - Especialista en fades');

});

test('loadSelects coloca placeholders correctamente', () => {

  document.body.innerHTML = `

    <select id="servicioSelect"></select>

    <select id="barberoSelect"></select>

  `;

  loadSelects();

  const servicioSelect = document.getElementById('servicioSelect');

  const barberoSelect = document.getElementById('barberoSelect');

  expect(servicioSelect.firstChild.textContent).toBe('Selecciona un servicio');

  expect(servicioSelect.firstChild.disabled).toBe(true);

  expect(servicioSelect.firstChild.selected).toBe(true);

  expect(barberoSelect.firstChild.textContent).toBe('Selecciona un barbero');

  expect(barberoSelect.firstChild.disabled).toBe(true);

  expect(barberoSelect.firstChild.selected).toBe(true);

});

test('muestra mensaje si no hay reservas', () => {

  document.body.innerHTML = `

    <ul id="bookingList"></ul>

  `;

  const reservas = [];

  renderBookings(reservas);

  expect(bookingList.children).toHaveLength(1);

  expect(bookingList.children[0].textContent).toBe('No hay reservas registradas.');

});

test('renderiza reservas correctamente', () => {

  const reservas = [

    {

      nombreCliente: 'Fran',

      fecha: '2025-08-10',

      hora: '15:00',

      nombreBarbero: 'Joaquín Rojas',

      ciCliente: '1234',

      servicio: 'Corte',

    },

  ];

  document.body.innerHTML = `

    <ul id="bookingList"></ul>

  `;

  renderBookings(reservas);

  const bookingList = document.getElementById('bookingList');

  expect(bookingList.children).toHaveLength(1);

  const li = bookingList.children[0];

  expect(li.querySelector('.booking-client').textContent).toBe('Fran');

  expect(li.querySelector('.booking-date').textContent).toBe('2025-08-10 - 15:00');

  expect(li.querySelector('.booking-barber').textContent).toBe('Barbero: Joaquín Rojas');

});

test('creates and stores a new reservation correctly', () => {

  document.body.innerHTML = `

      <input id="fecha" value="2025-08-10">

      <select id="horaSelect">

        <option value="15:00" selected>15:00</option>

      </select>

      <select id="barberoSelect">

        <option value="Joaquín Rojas" selected>Joaquín Rojas</option>

      </select>

      <input id="ciCliente" value="12345678">

      <input id="nombreCliente" value="Fran">

      <select id="servicioSelect">

        <option value="Corte" selected>Corte</option>

      </select>

      <div id="bookingFeedback"></div>

      <form id="bookingForm"></form>

    `;

  const reservas = [];

  reserva(reservas);

  expect(reservas).toHaveLength(1);

  expect(reservas[0]).toBeInstanceOf(Reserva);

  expect(reservas[0].nombreCliente).toBe('Fran');

  expect(reservas[0].ciCliente).toBe('12345678');

  expect(reservas[0].fecha).toBe('2025-08-10');

  expect(reservas[0].hora).toBe('15:00');

  expect(reservas[0].nombreBarbero).toBe('Joaquín Rojas');

  expect(reservas[0].servicio).toBe('Corte');

  const storedReservas = JSON.parse(localStorage.getItem('reservas'));

  expect(storedReservas).toHaveLength(1);

  expect(storedReservas[0].nombreCliente).toBe('Fran');

  const feedback = document.getElementById('bookingFeedback');

  expect(feedback.textContent).toBe('¡Reserva realizada con éxito!');

  expect(feedback.classList.contains('show-feedback')).toBe(true);

});

describe('Función reserva()', () => {

  let reservas = [];

  beforeEach(() => {

    document.body.innerHTML = `

      <form id="bookingForm">

        <input id="fecha" value="2025-08-10">

        <select id="horaSelect">

          <option value="15:00" selected>15:00</option>

        </select>

        <select id="barberoSelect">

          <option value="Joaquín Rojas" selected>Joaquín Rojas</option>

        </select>

        <input id="ciCliente" value="12345678">

        <input id="nombreCliente" value="Fran">

        <select id="servicioSelect">

          <option value="Corte" selected>Corte</option>

        </select>

        <div id="bookingFeedback"></div>

      </form>

    `;

    reservas.length = 0;

    localStorage.clear();

  });

  test('no crea reserva si falta la fecha', () => {

    document.getElementById('fecha').value = '';

    const result = reserva(reservas);

    expect(result).toBe(false);

    expect(reservas).toHaveLength(0);

    expect(localStorage.getItem('reservas')).toBeNull();

  });

  test('no sobrescribe reservas existentes', () => {

    reserva(reservas);

    document.getElementById('nombreCliente').value = 'Juan';

    reserva(reservas);

    const storedReservas = JSON.parse(localStorage.getItem('reservas'));

    expect(storedReservas).toHaveLength(2);

    expect(storedReservas[0].nombreCliente).toBe('Fran');

    expect(storedReservas[1].nombreCliente).toBe('Juan');

  });

  test('no crea reserva si falta el nombre del cliente', () => {

    document.body.innerHTML = `

      <form id="bookingForm">

        <input id="fecha" value="2025-08-10">

        <select id="horaSelect"><option value="15:00" selected>15:00</option></select>

        <select id="barberoSelect"><option value="Joaquín Rojas" selected>Joaquín Rojas</option></select>

        <input id="ciCliente" value="12345678">

        <input id="nombreCliente" value="">

        <select id="servicioSelect"><option value="Corte" selected>Corte</option></select>

        <div id="bookingFeedback"></div>

      </form>

    `;

    const reservas = [];

    const resultado = reserva(reservas);

    expect(resultado).toBe(false);

    expect(reservas).toHaveLength(0);

    expect(document.getElementById('bookingFeedback').textContent).toBe('Por favor complete todos los campos obligatorios');

  });

  test('resetea el formulario luego de hacer una reserva', () => {

    document.body.innerHTML = `

      <form id="bookingForm">

        <input id="fecha" value="2025-08-10">

        <select id="horaSelect"><option value="15:00" selected>15:00</option></select>

        <select id="barberoSelect"><option value="Joaquín Rojas" selected>Joaquín Rojas</option></select>

        <input id="ciCliente" value="12345678">

        <input id="nombreCliente" value="Fran">

        <select id="servicioSelect"><option value="Corte" selected>Corte</option></select>

        <div id="bookingFeedback"></div>

      </form>

    `;

    const reservas = [];

    const form = document.getElementById('bookingForm');

    let resetCalled = false;

    form.reset = () => { resetCalled = true; };

    reserva(reservas);

    expect(resetCalled).toBe(true);

  });

  test('renderiza la reserva luego de crearla', () => {

    document.body.innerHTML = `

      <ul id="bookingList"></ul>

    `;

    const reservas = [

      new Reserva("2025-08-10", "15:00", "Corte", "Joaquín Rojas", "12345678", "Fran")

    ];

    renderBookings(reservas);

    const li = document.querySelector('.booking-item');

    expect(li).not.toBeNull();

    expect(li.querySelector('.booking-client').textContent).toBe("Fran");

  });

  test('reserva se ejecuta correctamente con evento submit', () => {

    document.body.innerHTML = `

      <form id="bookingForm">

        <input id="fecha" value="2025-08-10">

        <select id="horaSelect"><option value="15:00" selected>15:00</option></select>

        <select id="barberoSelect"><option value="Joaquín Rojas" selected>Joaquín Rojas</option></select>

        <input id="ciCliente" value="12345678">

        <input id="nombreCliente" value="Fran">

        <select id="servicioSelect"><option value="Corte" selected>Corte</option></select>

        <div id="bookingFeedback"></div>

      </form>

    `;

    const reservas = [];

    const form = document.getElementById('bookingForm');

    let preventDefaultCalled = false;

    form.addEventListener('submit', (e) => {

      if (typeof e.preventDefault === 'function') {

        preventDefaultCalled = true;

        e.preventDefault();

      }

      reserva(reservas);

    });

    const event = new Event('submit', { bubbles: true, cancelable: true });

    form.dispatchEvent(event);

    expect(preventDefaultCalled).toBe(true);

    expect(reservas).toHaveLength(1);

  });

});

### Resultado de Tests

Texto

El contenido generado por IA puede ser incorrecto.