

Proyecto 2

Sistema implementado y reporte de las modificaciones realizadas

Sebastian Alejandro Juárez Rojas

Sistema implementado

Sistema Implementado: CLEAN CLEAN (Aplicación de Lavandería)

1. Introducción

CLEAN CLEAN es una aplicación web desarrollada bajo la metodología SCRUM, diseñada para simplificar la solicitud y gestión de servicios de lavandería. El sistema permite a los usuarios registrarse, personalizar pedidos, realizar pagos en línea y seguir el estado de sus pedidos, cumpliendo con los objetivos principales de usabilidad y eficiencia.

Enlace a la aplicación: <https://sebastianudg.github.io/CleanApp/>

Repositorio GitHub: <https://github.com/SebastianUDG/CleanApp>

2. Tecnologías y Herramientas

- **Frontend:** React (JavaScript) para la interfaz de usuario.
 - **Backend:** Firebase para autenticación de usuarios y base de datos en tiempo real.
 - **Diseño:** Canva para wireframes y prototipos.
 - **Gestión de Proyecto:** Google Docs, GitHub (control de versiones), y tableros SCRUM.
 - **Despliegue:** GitHub Pages.
-

3. Arquitectura del Sistema

- **Autenticación:** Integración con Firebase Auth para registro e inicio de sesión seguro.
- **Flujo de Pedidos:**
 - **Frontend:** React maneja la navegación en 5 pasos (selección de servicio, personalización, pago).

- **Backend:** Firebase Firestore almacena datos de usuarios, pedidos y transacciones.
- **Pagos:** Implementación básica de pasarela de pago (simulada) con validación de datos.

4. Funcionalidades Implementadas

Se completaron las **historias de usuario prioritarias (H1-H3)** en 3 sprints:

Historia	Funcionalidad	Detalles
H1	Registro y autenticación	Formulario con validación, conexión a Firebase Auth.
H2	Solicitud de servicio en 5 pasos	Selección de tipo de lavado (Básico, Express, Planchado, Seco, Premium) mediante botones intuitivos.
H3	Pago en línea	Integración de formulario de pago simulado, generación de comprobante.

Flujo completo del usuario:

1. Registro → 2. Inicio de sesión → 3. Selección de servicio → 4. Personalización → 5. Pago.

5. Proceso de Desarrollo (SCRUM)

- **Sprints:**
 - **Sprint 1:** Registro y autenticación (H1).
 - **Sprint 2:** Flujo de solicitud de servicio (H2).
 - **Sprint 3:** Integración de pagos (H3).
- **Ajustes clave:**

- Priorización de funcionalidades principales sobre secundarias (H4-H7 pospuestas).
- Rediseño del flujo de navegación para optimizar la experiencia de usuario (UX).

6. Desafíos y Soluciones

- **Desfase entre diseño y código:** Se realizaron pruebas tempranas y ajustes iterativos en colaboración entre el Scrum Master y el desarrollador.
- **Integración de Firebase:** Solucionado con documentación técnica y pruebas de conexión.
- **Priorización de tareas:** Uso de *Product Backlog* para reordenar historias según viabilidad y valor para el usuario.

7. Resultados y Métricas UX

- **Task Success:** 95% de usuarios completan el pedido en menos de 3 minutos.
- **Goal Completion:** 100% de funcionalidad principal operativa.
- **Task Efficiency:** Flujo reducido a 5 clics desde el inicio hasta el pago.

8. Conclusiones

- **Aprendizajes clave:**
 - La flexibilidad de SCRUM permitió adaptarse a cambios sin comprometer el objetivo final.
 - La documentación detallada fue esencial para la trazabilidad y mejora continua.
- **Próximos pasos:** Implementar H4-H7 (panel de administrador, notificaciones) en futuras iteraciones.

Enlaces adicionales

Herramienta	Propósito	Enlace
Google Docs	Documentación del proyecto	Enlace
GitHub	Repositorio del código	CleanApp
Canva	Diseño de interfaces	Diseños
GitHub Pages	Despliegue de la app	App en producción

Reporte de Modificaciones Realizadas en el Proyecto "CLEAN CLEAN"

1. Modificaciones en la Planeación y Priorización

Fase 1: Inicio del Proyecto

- **Ajuste de Roles:**
 - Asignación clara de roles: *Ángeles Janneth Esparza Vargas* como **Scrum Master** y *Sebastián Alejandro Juárez Rojas* como **Desarrollador Principal**.
 - Documentación inicial de requerimientos técnicos, priorizando funcionalidades básicas (registro, solicitud de servicio, pago).

Fase 2: Consolidación Técnica

- **Reasignación de Sprints:**
 - La historia **H2** (solicitud de servicio en 5 clics) se movió entre el *Sprint 1* y *Sprint 2* para optimizar recursos.
 - **H4** (panel de administrador) se pospuso del *Sprint 4* al *Sprint 2*, pero finalmente se relegó a baja prioridad.
- **Reducción de Alcance:**

- Historias **H4-H7** (funcionalidades de administrador y notificaciones) se clasificaron como **baja prioridad** para enfocarse en el flujo principal del usuario.

2. Ajustes en Historias de Usuario

Historia H2: Solicitud de Servicio

- **Rediseño del Flujo:**
 - Original: 5 clics para seleccionar tipo de lavado.
 - Modificación: **1 clic** para elegir entre opciones predefinidas (*Básico, Express, Planchado, Seco, Premium*), manteniendo el flujo en 5 pasos totales.
 - Justificación: Mejorar la eficiencia del usuario sin sacrificar la experiencia (p. 17-18).

Historia H3: Pago en Línea

- **Simplificación de Tareas:**
 - Eliminación de tareas secundarias como "guardar tarjeta" y "generar comprobante automático".
 - Enfoque en integración básica de formulario de pago y pruebas de usuario (p. 29-30).
- **Ajuste Estético:**
 - Reubicación del formulario de pago para garantizar centrado visual y accesibilidad (p. 30).

3. Cambios en el Product Backlog

- **Repriorización Dinámica:**
 - **H3** elevada de prioridad *Media* a *Alta* para asegurar la finalización del flujo principal.
 - **H4-H7** relegadas a *Sprints 4-6* con prioridad *Baja* (p. 20-22).
- **Reescritura de Criterios de Aceptación:**

- Especificación detallada de botones y pasos para historias como **H2** (p. 18).

4. Adaptaciones Metodológicas (SCRUM)

- **Estimación de Tiempos:**
 - Ajuste de horas por tarea: De 25 horas iniciales en *Sprint 1* a bloques de 5 horas diarias para mayor realismo (p. 13-14).
- **Pruebas Tempranas:**
 - Implementación de pruebas de registro y flujo de acceso en *Sprint 1* para validar funcionalidades críticas (p. 16).
- **Documentación Iterativa:**
 - Uso de *Google Drive* y *GitHub* para versionado, facilitando la trazabilidad de cambios (p. 3-4).

5. Aprendizajes y Adaptaciones

- **Flexibilidad en SCRUM:**
 - Aceptación de cambios como parte del proceso (ej: reubicación de H2 y H3).
 - Uso de *retrospectivas* para ajustar prioridades (p. 42-43).
- **Trabajo en Equipo:**
 - Colaboración entre roles: *Scrum Master* lideró diseño UX, mientras el *Desarrollador* implementó la lógica técnica (p. 44-45).
- **Enfoque en MVP:**
 - Postergación de funcionalidades secundarias (H4-H7) para entregar un producto mínimo viable funcional (p. 32-34).

6. Resultado Final

- **Flujo Principal Implementado:**

- Registro → Inicio de sesión → Solicitud de servicio → Personalización → Pago.
- Indicadores UX alcanzados: *Task Success (95%), Goal Completion (100%)* (p. 27-28).

- **Repositorio y Despliegue:**

- Aplicación disponible en: <https://sebastianudg.github.io/CleanApp/>.
- Código fuente: [GitHub](#).