**Proyecto ii**

App para la lavandería

Clean – Clean

**product backlog**

Sebastian Alejandro Juárez Rojas

**App de Clean – Clean App:**

[**https://sebastianudg.github.io/CleanApp/**](https://sebastianudg.github.io/CleanApp/)

**Repositorio de Clean - Clean App:** [**https://github.com/SebastianUDG/CleanApp.git**](https://github.com/SebastianUDG/CleanApp.git)

**1. Desarrollo y Aplicación de Scrum**

**Roles en Scrum**

* **Product Owner (PO)**: Sebastian Alejandro Juárez Rojas (tú).
* **Scrum Master (SM)**: Sebastian Alejandro Juárez Rojas (tú).
* **Equipo de Desarrollo**: Sebastian Alejandro Juárez Rojas (tú).

**Eventos de Scrum**

1. **Sprint Planning**:
   * Reunión para planificar el sprint.
   * Seleccionar las historias del backlog que se trabajarán en el sprint.
   * Dividir las historias en tareas y estimar el esfuerzo.
2. **Daily Scrum**:
   * Reunión diaria de 15 minutos para sincronizar el trabajo.
   * Preguntas clave:
     + ¿Qué hice ayer?
     + ¿Qué haré hoy?
     + ¿Hay algún impedimento?
3. **Sprint Review**:
   * Al final del sprint, revisar el trabajo completado.
   * Demostrar el funcionamiento del código.
4. **Sprint Retrospective**:
   * Reflexionar sobre el sprint.
   * Identificar qué salió bien, qué se puede mejorar y acciones concretas.

**2. Desarrollo del Código y Documentación de Sprint Planning**

**Sprint Planning**

* **Objetivo del Sprint**: Implementar las historias de usuario priorizadas (H1, H2 y H3).
* **Duración del Sprint**: 1 semana (5 días).
* **Tareas**:
  + **H1: Registro de usuario**:
    - Diseñar formulario de registro (2 horas).
    - Implementar validación de datos (2 horas).
    - Conectar con base de datos (1 hora).
  + **H2: Solicitud de servicio**:
    - Diseñar interfaz de selección de servicio (3 horas).
    - Implementar integración con API de pedidos (4 horas).
    - Agregar opción de programación de recolección (1 hora).
  + **H3: Pagos en línea**:
    - Integrar pasarela de pago (4 horas).
    - Diseñar pantalla de confirmación de pago (2 horas).
    - Implementar encriptación de datos (2 horas).

**Documentación del Sprint**

* **Objetivos**:
  + Implementar el registro de usuarios.
  + Permitir la solicitud de servicios de lavandería.
  + Integrar pagos en línea.
* **Recursos**:
  + Tiempo estimado: 31 horas.
  + Herramientas: React, Node.js, Firebase (para autenticación y base de datos), Stripe (para pagos).
* **Riesgos**:
  + Retrasos en la integración con la API.
  + Problemas de compatibilidad con la pasarela de pagos.

**3. Product Backlog**

**Tabla de Product Backlog**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Enunciado de la Historia | Alias | Estado | Esfuerzo | Prioridad | Responsable | Dependencias | Criterios de Aceptación |
| H1 | Como usuario, quiero registrarme en la app. | Registro | Pendiente | 5 | Alta | Sebastian Juárez | Ninguna | El usuario debe poder registrarse y autenticarse con éxito. |
| H2 | Como usuario, quiero solicitar un servicio. | Solicitud | Pendiente | 8 | Alta | Sebastian Juárez | H1 | El usuario debe poder seleccionar el tipo de lavado y confirmar el pedido. |
| H3 | Como usuario, quiero realizar pagos en línea. | Pagos | Pendiente | 8 | Media | Sebastian Juárez | H2 | El pago debe procesarse sin errores y generar un comprobante. |
| H4 | Como administrador, quiero ver los pedidos. | Dashboard | Pendiente | 5 | Media | Sebastian Juárez | H2 | El administrador debe visualizar los pedidos en un dashboard. |
| H5 | Como usuario, quiero recibir notificaciones. | Notificaciones | Pendiente | 5 | Baja | Sebastian Juárez | H2 | El usuario debe recibir notificaciones en tiempo real sobre el estado del pedido. |

**4. Fichas de Backlog**

**Ficha de H1: Registro de Usuario**

* **Descripción**: Como usuario, quiero registrarme en la app con mi correo y contraseña para acceder a los servicios.
* **Prioridad**: Alta.
* **Esfuerzo**: 5 puntos.
* **Criterios de Aceptación**:
  + El usuario debe poder registrarse con correo y contraseña.
  + El sistema debe validar los datos ingresados.
  + Los datos deben almacenarse en una base de datos.

**Ficha de H2: Solicitud de Servicio**

* **Descripción**: Como usuario, quiero solicitar un servicio de lavandería indicando la cantidad de prendas y el tipo de lavado.
* **Prioridad**: Alta.
* **Esfuerzo**: 8 puntos.
* **Criterios de Aceptación**:
  + El usuario debe poder seleccionar el tipo de lavado.
  + El sistema debe confirmar el pedido y generar un ID único.

**Ficha de H3: Pagos en Línea**

* **Descripción**: Como usuario, quiero realizar pagos en línea de forma segura.
* **Prioridad**: Media.
* **Esfuerzo**: 8 puntos.
* **Criterios de Aceptación**:
  + El pago debe procesarse sin errores.
  + El sistema debe generar un comprobante de pago.

**5. Calendarización de los Sprints**

**Sprint 1 (Semana 1)**

* **Duración**: 5 días.
* **Tareas**:
  + Diseñar formulario de registro (2 horas).
  + Implementar validación de datos (2 horas).
  + Conectar con base de datos (1 hora).
* **Objetivo**: Implementar el registro de usuarios.

**Sprint 2 (Semana 2)**

* **Duración**: 5 días.
* **Tareas**:
  + Diseñar interfaz de selección de servicio (3 horas).
  + Implementar integración con API de pedidos (4 horas).
  + Agregar opción de programación de recolección (1 hora).
* **Objetivo**: Permitir la solicitud de servicios.

**Sprint 3 (Semana 3)**

* **Duración**: 5 días.
* **Tareas**:
  + Integrar pasarela de pago (4 horas).
  + Diseñar pantalla de confirmación de pago (2 horas).
  + Implementar encriptación de datos (2 horas).
* **Objetivo**: Integrar pagos en línea.

**6. Funcionamiento del Código**

**Relación con los Objetivos del Proyecto**

* **Registro de Usuario**: Permite a los usuarios acceder a los servicios de la app.
* **Solicitud de Servicio**: Facilita la interacción del usuario con los servicios de lavandería.
* **Pagos en Línea**: Ofrece una forma segura y eficiente de realizar transacciones.

**Sprint 1 (Duración: 1 semana)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Elemento de Trabajo | Puntos | Responsable | Estado | Estimado | Día 1 | Día 2 | Día 3 | Día 4 | Día 5 | Revisión |
| H1: Registro de usuario | 5 | Sebastian Juárez | Pendiente | 5 horas |  |  |  |  |  |  |
| - Diseñar formulario | 2 | Sebastian Juárez | Pendiente | 2 horas | X |  |  |  |  |  |
| - Validación de datos | 2 | Sebastian Juárez | Pendiente | 2 horas |  | X |  |  |  |  |
| - Conexión con base de datos | 1 | Sebastian Juárez | Pendiente | 1 hora |  |  | X |  |  |  |

**Sprint 2 (Duración: 1 semana)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Elemento de Trabajo | Puntos | Responsable | Estado | Estimado | Día 1 | Día 2 | Día 3 | Día 4 | Día 5 | Revisión |
| H2: Solicitud de servicio | 8 | Sebastian Juárez | Pendiente | 8 horas |  |  |  |  |  |  |
| - Diseñar interfaz de selección | 3 | Sebastian Juárez | Pendiente | 3 horas | X |  |  |  |  |  |
| - Integración con API de pedidos | 4 | Sebastian Juárez | Pendiente | 4 horas |  | X | X |  |  |  |
| - Opción de programación | 1 | Sebastian Juárez | Pendiente | 1 hora |  |  |  | X |  |  |

**Explicación:**

* **Tareas Desglosadas**: Cada historia del backlog se divide en tareas concretas (por ejemplo, "Diseñar formulario de registro").
* **Estimación de Tiempo**: Cada tarea tiene un tiempo estimado para su realización.
* **Seguimiento**: Se marca el progreso diario (con una "X") y se actualiza el estado (Pendiente, En Proceso, Terminado).

**3. Relación entre Backlog y Sprint**

* **Backlog**: Captura los requerimientos del cliente y las necesidades de los usuarios en forma de historias.
* **Sprint**: Toma las historias del backlog y las divide en tareas concretas para un ciclo de trabajo específico.
* **Actualización de Estados**:
  + En el **Backlog**, las historias cambian de estado (Pendiente, En Proceso, Terminado) a medida que avanzan los sprints.
  + En el **Sprint**, las tareas se marcan como completadas y se ajustan según sea necesario (por ejemplo, si una tarea toma más tiempo del estimado).

**Ejemplo de mi código**

**// src/components/Registro.js**

**import React, { useState } from 'react';**

**function Registro() {**

**const [email, setEmail] = useState('');**

**const [password, setPassword] = useState('');**

**const handleSubmit = (e) => {**

**e.preventDefault();**

**// Lógica para enviar datos al backend**

**};**

**return (**

**<form onSubmit={handleSubmit}>**

**<input**

**type="email"**

**placeholder="Correo"**

**value={email}**

**onChange={(e) => setEmail(e.target.value)}**

**/>**

**<input**

**type="password"**

**placeholder="Contraseña"**

**value={password}**

**onChange={(e) => setPassword(e.target.value)}**

**/>**

**<button type="submit">Registrarse</button>**

**</form>**

**);**

**}**

**export default Registro;**

**Relación con los Objetivos del Proyecto**

* **Cada componente desarrollado (como Registro.js) está alineado con los objetivos del proyecto:**
  + **Registro de usuario: Permite a los usuarios acceder a los servicios.**
  + **Solicitud de servicio: Facilita la interacción del usuario con los servicios de lavandería.**
  + **Pagos en línea: Ofrece una forma segura de realizar transacciones*.***

***Referencias***

* *Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley.*
* *Cohn, M. (2010). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley.*
* *Pichler, R. (2012). Agile Product Management with Scrum: Creating Products That Customers Love. Addison-Wesley.*
* *Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley.*
* *Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.*
* *Sommerville, I. (2020). Software Engineering (10th ed.). Pearson.*
* *Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.*
* *React Documentation. (2023). Recuperado de* [*https://reactjs.org/docs/getting-started.html*](https://reactjs.org/docs/getting-started.html)*.*
* *GitHub. (2023). Recuperado de* [*https://github.com*](https://github.com)*.*
* *Stripe Documentation. (2023). Recuperado de* [*https://stripe.com/docs*](https://stripe.com/docs)*.*