Texto

Descripción generada automáticamente con confianza media

USUARIO

PROYECTO ii

**Actividad integradora. Sprint, el corazón de SCRUM**

**Implementación de SCRUM**

SCRUM es un marco de trabajo ágil que facilita la entrega incremental de valor en ciclos cortos llamados sprints.

**Roles clave:**

•Product Owner: Define la visión del producto y prioriza los requisitos.

•Scrum Master: Facilita el proceso SCRUM y elimina impedimentos.

•Development Team: Desarrolladores encargados de construir y probar el producto.

**Artefactos principales:**

•Product Backlog: Lista priorizada de funcionalidades y mejoras.

•Sprint Backlog: Selección de tareas del Product Backlog para el sprint actual.

•Incremento: Producto funcional al final de cada sprint.

**Eventos SCRUM:**

1.Sprint Planning (Planificación del Sprint)

2.Daily Scrum (Reuniones diarias de seguimiento)

3.Sprint Review (Revisión del Sprint)

4.Sprint Retrospective (Retrospectiva del Sprint)

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

El código se desarrollará en React, utilizando herramientas modernas como Vite para la configuración del entorno y GitHub para la gestión del código fuente.

**Pasos iniciales:**

1.Configuración del proyecto

2.Desarrollo de componentes: Se modularizan las secciones del sitio web en React.

3.Integración con APIs (si aplica): Para gestionar datos dinámicos.

4.Pruebas y refinamiento: Se aplican pruebas unitarias con Jest y React Testing Library.

**Publicación del Código en GitHub**

1.Crear un repositorio en GitHub.

2.Subir el código

Estructura del proyecto en React:

|  |
| --- |
| mi-proyecto/  │── src/  │ ├── components/ # Componentes reutilizables  │ │ ├── Header.jsx  │ │ ├── Footer.jsx  │ │ ├── Home.jsx  │ │ └── Products.jsx  │ ├── App.jsx # Componente principal  │ ├── main.jsx # Punto de entrada de la app  │ ├── index.css # Estilos globales  │── public/ # Archivos estáticos  │── package.json # Dependencias y configuración  │── vite.config.js # Configuración de Vite  │── README.md # Documentación del proyecto |

Componente funcional

Barra de navegación (Header.jsx):

|  |
| --- |
| import React from "react";  const Header = () => {  return (  <header style={{ background: "#333", padding: "10px", color: "white" }}>  <h1>Mi Sitio Web</h1>  <nav>  <a href="/">Inicio</a> | <a href="/productos">Productos</a>  </nav>  </header>  );  };  export default Header; |

Componente principal

Organiza la pagina (App.jsx):

|  |
| --- |
| import React from "react";  import Header from "./components/Header";  import Footer from "./components/Footer";  const App = () => {  return (  <div>  <Header />  <main>  <h2>Bienvenido a nuestro sitio web</h2>  <p>Explora nuestros productos y descubre lo que ofrecemos.</p>  </main>  <Footer />  </div>  );  };  export default App; |

Compartir el enlace del repositorio en el documento final.

**Documento Integrado**

El documento incluirá:

•Acceso al repositorio GitHub: Enlace directo.

•Product Backlog: Lista de características y tareas priorizadas.

•Fichas de Backlog: Descripción detallada de cada funcionalidad con criterios de aceptación.

•Calendarización de los Sprints: Cronograma con fechas de inicio, duración y entregables por sprint.

**Ejemplo de Product Backlog**

|  |  |  |
| --- | --- | --- |
| **Historia de usuario** | **Prioridad** | **Criterios de aceptación** |
| Como usuario, quiero ver una página de inicio con un diseño atractivo | Alta | Debe contener un banner, un botón de contacto y un footer |
| Como usuario, quiero una página de productos con filtros | Media | Se debe filtrar por categoría y precio |

**Ejemplo de Sprint Backlog**

|  |  |  |
| --- | --- | --- |
| **Tarea** | **Responsable** | **Estado** |
| Diseñar la UI de la página de inicio | Juan | En progreso |
| Implementar la lógica de filtrado en React | Ana | Pendiente |

**Calendarización del Sprint**

El sprint durará 2 semanas, con reuniones diarias de 15 minutos.

|  |  |  |
| --- | --- | --- |
| **Semana** | **Actividad** | **Responsable** |
| 1 | Configuración del entorno y diseño de UI | Equipo |
| 2 | Implementación de lógica, pruebas y ajustes | Equipo |

Referencias

* Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., … & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance.
* Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.
* Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley.
* McKenna, P. (2016). Scrum: A Pocket Guide. Van Haren Publishing.
* Freeman, E., & Robson, E. (2020). Head First Learn to Code: A Learner’s Guide to Coding and Computational Thinking. O’Reilly Media.
* React Documentation. (2024). React: A JavaScript library for building user interfaces. Meta Open Source.
* Fitzgerald, B., & Stol, K. (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123, 176-189.
* Martin, R. C. (2011). The Clean Coder: A Code of Conduct for Professional Programmers. Prentice Hall.