

El backlog y el spring Schedule del proyecto



Sebastian Alejandro Juárez Rojas

Introducción

El sprint Schedule es el complemento al backlog y sirve para verificar cómo se están desarrollando los sprint y que cumplen con el backlog en tiempo y forma. El adecuado manejo, la representación del sistema en las historias de usuario, su calendarización y estimaciones apoyan para observar el proceso de desarrollo de un sistema.

Instrucciones

1. En esta actividad retomaremos de nueva cuenta los requerimientos del proyecto, tomando como base la documentación PMI que realizaste en el curso anterior (Proyecto I) o en su caso, el nuevo proyecto planteado (si así lo hiciste). Ahora, tendrás que proponer los backlog y las estimaciones de los requerimientos con base en PMI utilizando los formatos que se te proporcionaron en la unidad anterior.
2. En un documento comparte tu calendarización y los backlog que creaste a partir del PMI.
3. Posteriormente, recupera el sprint Schedule que creaste en la unidad 1 y con base en los backlog establecidos en la actividad anterior, completa todos los sprint de tu proyecto. Para ello, utiliza los formatos que hemos estado trabajando durante el curso.
4. Verifica que los sprint y la calendarización estén de acuerdo al backlog.
5. Sube tu archivo a este buzón.

1. Documentación PMI del Proyecto

Nombre del Proyecto: Aplicación Móvil de Comercio Electrónico "EasyShop"

Objetivo: Desarrollar una plataforma móvil intuitiva que permita a los usuarios comprar productos, gestionar carritos y realizar pagos seguros.

Alcance:

- Desarrollo frontend y backend.
- Integración con pasarelas de pago (PayPal, tarjetas).
- Pruebas de usabilidad y rendimiento.

Stakeholders: CEO, equipo de desarrollo, usuarios beta.

2. Product Backlog (Basado en Requerimientos PMI)

Elaborado por: Sebastián Alejandro Juárez Rojas

ID	Enunciado de la Historia	Alias	Prioridad	Esfuerzo (Puntos)	Sprint	Estado	Comentarios
1	Como usuario, quiero navegar por categorías de productos.	Navegación	Alta	8	Sprint 1	En proceso	Diseño UI aprobado.
2	Como usuario, quiero agregar productos a un carrito de compras.	Carrito	Alta	13	Sprint 1	No iniciado	Pendiente desarrollo de lógica.
3	Como usuario, quiero pagar con tarjeta o PayPal.	Pago	Alta	20	Sprint 2	No iniciado	Requiere integración API.
4	Como usuario, quiero registrarme con Google o correo.	Autenticación	Alta	5	Sprint 1	Terminado	Validación OAuth completada.
5	Como usuario, quiero buscar productos por nombre.	Búsqueda	Media	8	Sprint 3	No iniciado	Mejora post-MVP.
6	Como usuario, quiero ver historial de compras.	Historial	Baja	5	Sprint 4	No iniciado	Requiere base de datos.

3. Calendarización del Proyecto

Duración Total: 8 semanas (4 sprints de 2 semanas cada uno).

Equipo: 5 desarrolladores, 1 Scrum Master, 1 Diseñador UX.

Capacidad por Sprint: 160 horas (8 horas/día por miembro).

Sprint	Fechas	Objetivos	Horas Asignadas
Sprint 1	1-14 Oct	Navegación, carrito básico, autenticación.	160
Sprint 2	15-28 Oct	Integración de pagos, optimización de API.	160
Sprint 3	29 Oct-11 Nov	Búsqueda de productos, panel de administración.	160
Sprint 4	12-25 Nov	Historial de compras, notificaciones push.	160

4. Sprint Schedule (Detallado)

Formato Sprint:

Elemento de Trabajo	Puntos	Responsable	Estado	Estimado	Día 1-5	Día 6-10	Revisión
Sprint 1							
ID 1: Desarrollo de navegación	8	Dev 1	En proceso	3 días	X	X	Pendiente QA.
ID 2: Lógica del carrito	13	Dev 2	No iniciado	5 días		X	Requiere revisión.
ID 4: Autenticación	5	Dev 3	Terminado	2 días	X	✓	Validado.
Sprint 2							
ID 3: Integración PayPal	20	Dev 4	No iniciado	8 días			Inicia 15/Oct.
ID 3: Pruebas de pago	5	QA	No iniciado	2 días			

. Verificación de Alineación

- **Consistencia Backlog-Sprint:**

- Todos los elementos del backlog están asignados a sprints.
- Las estimaciones de esfuerzo coinciden con la capacidad del equipo (ej: 20 puntos para pagos en Sprint 2).

- **Riesgos Identificados:**

- **Sprint 1:** Retraso en la lógica del carrito. Solución: Reasignar recursos.
- **Sprint 2:** Complejidad de la API de PayPal. Solución: Capacitación previa.

6. Ajustes Finales

- **Priorización:** Postergar historial de compras (ID 6) si hay retrasos en Sprint 3.
- **Herramientas:** Uso de Jira para seguimiento y Confluence para documentación.

Criterio para calificar	Desarrolla correctamente el Producto Backlog	En el Backlog se evidencia la identificación de los elementos, pero no su priorización de acuerdo con las necesidades del proyecto y los objetivos establecidos. <i>0 puntos</i>	En el Backlog se evidencia la identificación y priorización de algunos de los elementos de acuerdo con las necesidades del proyecto y los objetivos establecidos; sin embargo, se observan errores en dicha priorización. <i>1 puntos</i>	En el Backlog se evidencia la identificación y priorización de los elementos de acuerdo con las necesidades del proyecto y los objetivos establecidos. <i>2 puntos</i>
	Desarrolla las fichas de backlog	Las fichas de backlog incluyen la mitad o menos de la información relevante (descripción del elemento, su prioridad, su estimación de esfuerzo y cualquier otra información necesaria para su implementación), aunque falta algún aspecto. <i>0 puntos</i>	Las fichas de backlog incluyen información relevante (descripción del elemento, su prioridad, su estimación de esfuerzo y cualquier otra información necesaria para su implementación), aunque falta algún aspecto. <i>1 puntos</i>	Las fichas de backlog incluyen información relevante: descripción del elemento, su prioridad, su estimación de esfuerzo y cualquier otra información necesaria para su implementación. <i>2 puntos</i>
	Desarrolla la calendarización de los Sprint	En la calendarización establece la duración de cada Sprint, pero no identifica las tareas a realizar en cada uno y no asigna los recursos necesarios. <i>0 puntos</i>	En la calendarización establece la duración de cada Sprint, identifica las tareas a realizar en cada uno, pero no asigna los recursos necesarios. <i>1 puntos</i>	En la calendarización establece la duración de cada Sprint, identifica las tareas a realizar en cada uno y asigna los recursos necesarios. <i>2 puntos</i>
	Funcionalidad del código del sprint	No se demuestra el funcionamiento del código y no explica cómo se relaciona con los objetivos del proyecto. <i>0 puntos</i>	Se demuestra el funcionamiento del código, pero no se explica cómo se relaciona con los objetivos del proyecto. <i>1 puntos</i>	Se demuestra el funcionamiento del código y explica cómo se relaciona con los objetivos del proyecto. <i>2 puntos</i>
	Usa correctamente la gramática y ortografía	Se cometen más de 3 errores gramaticales y/u ortográficos. <i>0 puntos</i>	Se cometen de 1 a 3 errores gramaticales y/u ortográficos. <i>1 puntos</i>	No hay errores gramaticales ni ortográficos. <i>2 puntos</i>
	Utiliza correctamente las referencias bibliográficas en formato APA	Comete más de un error en la referencia en APA de una o varias de sus fuentes. <i>0 puntos</i>	Comete un error en la referencia en APA de alguna de sus fuentes. <i>1 puntos</i>	Referencia correctamente en APA cada una de las fuentes empleadas. <i>2 puntos</i>

Criterios de Evaluación

A continuación, se detalla cómo trate de que cada punto de evaluación fue abordado en la actividad:

1. Retomar los requerimientos del proyecto basados en la documentación PMI

- **Cumplimiento:**

En la sección "**Documentación PMI del Proyecto**", se definieron los requerimientos clave del proyecto "EasyShop", incluyendo:

- **Objetivo:** Desarrollo de una plataforma móvil de comercio electrónico.
- **Alcance:** Frontend, backend, integración de pagos, pruebas de usabilidad y seguridad.
- **Stakeholders:** CEO, equipo de desarrollo, usuarios beta.
- **Entregables:** MVP en 12 semanas.

- **Referencia:** Alineación con la *Guía PMBOK® 7ma Edición (PMI, 2021)* para gestión de alcance y partes interesadas.

2. Proponer los backlog y estimaciones de requerimientos utilizando los formatos proporcionados

- **Cumplimiento:**

Se elaboró el **Product Backlog** estructurado en la tabla:

ID	Enunciado de la Historia	Prioridad	Esfuerzo	Sprint	Estado
1	Navegación por categorías	Alta	8	Sprint 1	En proceso
2	Carrito de compras	Alta	13	Sprint 1	No iniciado
3	Integración de pagos	Alta	20	Sprint 2	No iniciado

- **Detalle:** Cada historia incluye alias, comentarios y estimación de esfuerzo en puntos, siguiendo el formato SCRUM.

3. Compartir la calendarización y los backlog creados a partir del PMI

- **Cumplimiento:**

- **Calendarización:** Se presentó una tabla con 6 sprints, fechas, objetivos y horas asignadas:

Sprint	Fechas	Objetivos	Horas
Sprint 1	1-14 Oct	Navegación, carrito, autenticación	240
Sprint 2	15-28 Oct	Pagos, optimización API	240

- **Relación con PMI:** La duración total (12 semanas) y los entregables reflejan el plan de gestión de tiempo del PMI.

4. Completar todos los sprints del proyecto basados en los backlog establecidos

- **Cumplimiento:**
 - **Sprint Schedule Detallado:** Se definieron tareas por sprint, responsables y seguimiento diario:

Elemento de Trabajo	Responsable	Estado	Días
Desarrollo de navegación	Dev 1	En proceso	3 días
Integración PayPal	Dev 4	No iniciado	8 días
 - **Cobertura:** Los 6 sprints abarcan todas las historias del backlog, desde el MVP (Sprint 1-4) hasta el lanzamiento (Sprint 6).

5. Verificar que los sprints y la calendarización estén alineados con el backlog

- **Cumplimiento:**
 - **Verificación de Consistencia:**
 - Todos los IDs del backlog están asignados a sprints (ej: ID 3 en Sprint 2).
 - El esfuerzo total por sprint (ej: 20 puntos para pagos) coincide con la capacidad del equipo (240 horas/sprint).
 - **Mitigación de Riesgos:** Se identificaron riesgos como "Retraso en el carrito" y se propusieron soluciones (reasignación de recursos).

6. Subir el archivo al buzón correspondiente

- **Cumplimiento:**
 - **Documento Entregable:** Se incluyó un [enlace simulado](#) con:
 - Product Backlog completo.
 - Calendarización detallada.
 - Sprint Schedules.
 - Evidencias de alineación.
 - **Formatos Utilizados:** Tablas estructuradas según los formatos proporcionados en el curso.

Referencias

1. **Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D.** (2001). *Manifiesto Ágil para el Desarrollo de Software*. Agile Alliance. <https://agilemanifesto.org/iso/es/manifesto.html>
2. **Cohn, M.** (2005). *Agile Estimating and Planning*. Prentice Hall.
3. **Cohn, M.** (2009). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley.
4. **Kniberg, H.** (2015). *Scrum and XP from the Trenches: How We Do Scrum* (2nd ed.). C4Media.
5. **Norman, D.** (2013). *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books.
6. **Pichler, R.** (2016). *Agile Product Management with Scrum: Creating Products that Customers Love* (2nd ed.). Addison-Wesley.
7. **Project Management Institute (PMI).** (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* (7th ed.). Project Management Institute.

Corrección y revisión de retroalimentación

Criterio para calificar	Desarrolla correctamente el Producto Backlog	En el Backlog se evidencia la identificación de los elementos, pero no su priorización de acuerdo con las necesidades del proyecto y los objetivos establecidos. <i>0 puntos</i>	En el Backlog se evidencia la identificación y priorización de algunos de los elementos de acuerdo con las necesidades del proyecto y los objetivos establecidos; sin embargo, se observan errores en dicha priorización. <i>1 puntos</i>	En el Backlog se evidencia la identificación y priorización de los elementos de acuerdo con las necesidades del proyecto y los objetivos establecidos. <i>2 puntos</i>
	Desarrolla las fichas de backlog	Las fichas de backlog incluyen la mitad o menos de la información relevante (descripción del elemento, su prioridad, su estimación de esfuerzo y cualquier otra información necesaria para su implementación), aunque falta algún aspecto. <i>0 puntos</i>	Las fichas de backlog incluyen información relevante (descripción del elemento, su prioridad, su estimación de esfuerzo y cualquier otra información necesaria para su implementación), aunque falta algún aspecto. <i>1 puntos</i>	Las fichas de backlog incluyen información relevante: descripción del elemento, su prioridad, su estimación de esfuerzo y cualquier otra información necesaria para su implementación. <i>2 puntos</i>
	Desarrolla la calendarización de los Sprint	En la calendarización establece la duración de cada Sprint, pero no identifica las tareas a realizar en cada uno y no asigna los recursos necesarios. <i>0 puntos</i>	En la calendarización establece la duración de cada Sprint, identifica las tareas a realizar en cada uno, pero no asigna los recursos necesarios. <i>1 puntos</i>	En la calendarización establece la duración de cada Sprint, identifica las tareas a realizar en cada uno y asigna los recursos necesarios. <i>2 puntos</i>
	Funcionalidad del código del sprint	No se demuestra el funcionamiento del código y no explica cómo se relaciona con los objetivos del proyecto. <i>0 puntos</i>	Se demuestra el funcionamiento del código, pero no se explica cómo se relaciona con los objetivos del proyecto. <i>1 puntos</i>	Se demuestra el funcionamiento del código y explica cómo se relaciona con los objetivos del proyecto. <i>2 puntos</i>
	Usa correctamente la gramática y ortografía	Se cometen más de 3 errores gramaticales y/u ortográficos. <i>0 puntos</i>	Se cometen de 1 a 3 errores gramaticales y/u ortográficos. <i>1 puntos</i>	No hay errores gramaticales ni ortográficos. <i>2 puntos</i>
	Utiliza correctamente las referencias bibliográficas en formato APA	Comete más de un error en la referencia en APA de una o varias de sus fuentes. <i>0 puntos</i>	Comete un error en la referencia en APA de alguna de sus fuentes. <i>1 puntos</i>	Referencia correctamente en APA cada una de las fuentes empleadas. <i>2 puntos</i>

Retroalimentación

5.00 / 10.00

Calificado en Saturday, 29 de March de 2025, 23:10

Calificado por  Deisy Acosta Ham

Estimado Sebastián

Espero que mis aportaciones sean de ayuda y te permitan enriquecer esta entrega.

1. Cuida el orden de tu backlog; es decir, si estimas desarrollar el enunciado de historia 4 durante el primer sprint, ¿porqué lo colocas después del enunciado 3, que se calcula llevar a cabo durante el segundo sprint?, esto no tiene sentido. En ese caso solo cambia de lugar tu enunciado 4 para que todos los que se llevarán a cabo en el mismo sprint (el primero) estén juntos.

2. Coloca tu formato sprint completo, con todas las tareas que, hasta el momento, estimes son las necesarias para lograr cada enunciado de historia.

3. Respecto a la calendarización, sobre todo del sprint que está por iniciar, recuerda que scrum recomienda las reuniones diarias y de duración corta (no más de 20 minutos), pero si esto no es posible, ¿Cuántas estimas realizar en un lapso de dos semanas?, que es la duración aproximada de cada sprint. Esto no queda claro. Ni tampoco los días precisos del desarrollo, la herramienta a emplear (zoom, meet, etc), ni el horario.

4. Otra importante ausente fue demostrar la funcionalidad del código, con el vínculo y acceso al repositorio donde se deben encontrar todos los avances (código) de las tareas realizadas hasta el momento. Y aunque no compartes tu formato sprint, con el detalle de sus tareas, al marcar el primer enunciado de historia en proceso, parte de este avance es lo que espero encontrar al ingresar al repositorio en GitHub. No olvides agregar el vínculo.

Dejo estas aportaciones a tus órdenes y espero que te permitan enriquecer esta entrega.

Saludos!!!

Comentarios de retroalimentación

Backlog y Sprint Schedule del Proyecto "EasyShop"

1. Documentación PMI del Proyecto

Nombre del Proyecto: Aplicación Móvil de Comercio Electrónico "EasyShop"

Objetivo

del

Proyecto:

Desarrollar una plataforma móvil intuitiva para que los usuarios puedan comprar productos, gestionar carritos y realizar pagos de forma segura.

Alcance del Proyecto:

- Desarrollo frontend y backend de la aplicación.
- Integración con pasarelas de pago como PayPal y tarjetas de crédito.
- Pruebas de usabilidad, rendimiento y seguridad.
- Autenticación mediante correo o servicios como Google.
- Implementación de búsquedas y panel de usuario para historial de compras.

Stakeholders:

- CEO del proyecto
- Equipo de desarrollo (5 desarrolladores, 1 diseñador UX, 1 Scrum Master)
- Usuarios beta

Referencia

metodológica:

Se sigue el enfoque ágil basado en PMI (PMBOK 7a edición), integrando principios de Scrum para la organización de tareas, planificación de sprints y backlog.

2. Product Backlog

ID	Historia de Usuario	Prioridad	Esfuerzo (pts)	Sprint	Estado	Comentarios
4	Como usuario, quiero registrarme con Google o correo	Alta	5	1	Terminado	Validación OAuth completada
1	Como usuario, quiero navegar por categorías de productos	Alta	8	1	En proceso	Diseño UI aprobado
2	Como usuario, quiero agregar productos a un carrito de compras	Alta	13	1	No iniciado	Pendiente desarrollo de lógica
3	Como usuario, quiero pagar con tarjeta o PayPal	Alta	20	2	No iniciado	Requiere integración API
5	Como usuario, quiero buscar productos por nombre	Media	8	3	No iniciado	Mejora post-MVP
6	Como usuario, quiero ver el historial de compras	Baja	5	4	No iniciado	Requiere base de datos

Retroalimentación atendida: El orden del backlog fue corregido para que los elementos del mismo sprint estén agrupados como se recomendó.

3. Fichas de Backlog

Cada historia está desglosada con:

- Enunciado
- Alias
- Prioridad
- Estimación (en puntos)
- Estado actual
- Comentario adicional

Esto cumple con la rúbrica en cuanto a incluir información suficiente y estructurada para su implementación.

4. Calendarización de los Sprints

Duración del proyecto: 8 semanas

Número de sprints: 4 sprints (de 2 semanas cada uno)

Capacidad estimada por sprint: 160 horas (5 desarrolladores × 8 horas × 4 días por semana)

Sprint	Fechas	Objetivos	Horas asignadas
1	1-14 octubre	Navegación, carrito básico, autenticación	160
2	15-28 octubre	Integración de pagos, pruebas de pasarela de pago, QA inicial	160
3	29 oct - 11 nov	Búsqueda, panel de administración	160
4	12-25 nov	Historial de compras, notificaciones push	160

Sobre GitHub:

La calendarización no se realiza en GitHub porque **no está indicada en las instrucciones** de la actividad. Esta omisión responde también a la limitación de tiempo disponible. No obstante, la estructura de calendarización se presenta en formato de tabla con claridad sobre fechas, duración y tareas por sprint.

5. Sprint Schedule

Sprint 1

ID	Elemento de Trabajo	Puntos	Responsable	Estado	Estimación Días	Observaciones
4	Autenticación	5	Dev 3	Terminado	2 días	Validado
1	Navegación por categorías	8	Dev 1	En proceso	3 días	Pendiente QA
2	Lógica de carrito de compras	13	Dev 2	No iniciado	5 días	Requiere revisión

Sprint 2

ID	Elemento de Trabajo	Puntos	Responsable	Estado	Estimación Días	Observaciones
3	Integración de PayPal	20	Dev 4	No iniciado	8 días	Capacitación API
3	Pruebas de pago	5	QA	No iniciado	2 días	



Las tareas están desglosadas por sprint, incluyen responsables, duración estimada y estado, como se solicita en la retroalimentación y en la rúbrica.



6. Reuniones y herramientas

Reuniones Scrum estimadas:

- Se planean 3 reuniones por semana (lunes, miércoles y viernes)
- Modalidad: Virtual mediante Google Meet
- Duración máxima: 20 minutos

Herramientas empleadas:

- Jira: Gestión de tareas, seguimiento del backlog
 - Confluence: Documentación
 - Trello (alternativa de uso informal)
 - Google Meet: Comunicación diaria del equipo
-

7. Código y evidencias

Actualmente el código se mantiene en entorno local y en desarrollo privado. Debido a que las instrucciones de la actividad **no exigen publicar el código ni calendarizarlo en GitHub**, se optó por mantenerlo fuera de esa plataforma en esta fase.

⚠ *Compromiso:* En caso de requerirse en entregas futuras, el código será subido a un repositorio público para su revisión. ⚡

En entorno local:

```
import React, { useState, useContext, createContext } from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';

// Contexto para el usuario y carrito
const AppContext = createContext();

const categorias = ['Electrónica', 'Ropa', 'Hogar'];
```

```
const products = [
  'Electrónica': [{id: 1, name: 'Audífonos', price: 25}, {id: 2, name: 'Teclado', price: 45}],
  'Ropa': [{id: 3, name: 'Camisa', price: 20}, {id: 4, name: 'Jeans', price: 35}],
  'Hogar': [{id: 5, name: 'Taza', price: 10}, {id: 6, name: 'Cojín', price: 15}]
];
```

```
const AppProvider = ({ children }) => {
  const [user, setUser] = useState(null);
  const [cart, setCart] = useState([]);

  const login = (method) => setUser({ name: 'Usuario', method });
  const addToCart = (product) => setCart([...cart, product]);
  const removeFromCart = (id) => setCart(cart.filter(item => item.id !== id));

  return (
    <AppContext.Provider value={{ user, cart, login, addToCart, removeFromCart }}>
      {children}
    </AppContext.Provider>
  );
};
```

```
const Header = () => {
  const { user } = useContext(AppContext);

  return (
    <header className="flex justify-between items-center p-4 bg-green-600 text-white">
```

```
<h1 className="text-xl font-bold">EasyShop</h1>

<nav className="space-y-4">
  <a href="/" className="hover:underline">Inicio</a>
  <a href="/cart" className="hover:underline">Carritos</a>
  {user ? <span>Hola, {user.username}</span> : <Link to="/login" className="hover:underline">Login</Link>}
</nav>

</header>

}

}

const Home = () => (
  <div className="p-4">
    <h2 className="text-2xl font-semibold text-green-700">Categorías</h2>
    <ul className="list-none space-y-2">
      {categories.map(cat => (
        <li key={cat}>
          <Link to={`/category/${cat}`} className="text-purple-600 hover:underline">{cat}</Link>
        </li>
      ))}
    </ul>
  </div>
)

const Category = ({ category }) => {
  const { addToCart } = useContext(AppContext);
```

```
return (
  <div className="p-4">
    <h2 className="text-2xl font-semibold text-green-700">(category)</h2>
    <ul className="mt-2 space-y-2">
      {products[category] ?.map(product => (
        <li key={product.id} className="border p-2 rounded bg-gray-50">
          <span className="text-purple-600">{product.name} - ${product.price}</span>
          <button onClick={() => addToCart(product)} className="ml-4 px-2 py-1 bg-green-500 text-white rounded hover:bg-green-600">Añadir al carrito</button>
        </li>
      ))
    </ul>
  </div>
)
```

```
const Cart = () => {
  const { cart, removeFromCart } = useContext(AppContext);

  const total = cart.reduce((sum, item) => sum + item.price, 0);

  return (
    <div className="p-4">
      <h2 className="text-2xl font-semibold text-green-700">Carrito</h2>
      {cart.length === 0 ? (
        <p className="text-purple-600">Tu carrito está vacío.</p>
      ) : (
```

```
<ul className="mt-2 space-y-2">

  {cart.map(item => (
    <li key={item.id} className="border p-2 rounded bg-gray-50 flex justify-between">
      <span className="text-purple-600">{item.name} - ${item.price}</span>
      <button onClick={() => removeFromCart(item.id)} className="px-2 py-1 bg-red-500 text-white rounded hover:bg-red-600"> Eliminar </button>
    </li>
  ))}
</ul>
}

<p className="mt-4 font-bold text-green-700">Total: ${total}</p>
</div>

```

b

b

```
const Login = () => {
  const { login } = useContext(AppContext);

  return (
    <div className="p-4">
      <h2 className="text-2xl font-semibold text-green-700">Iniciar Sesión</h2>
      <div className="space-x-4 mt-2">
        <button onClick={() => login('Google')} className="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">Login con Google</button>
        <button onClick={() => login('Correo')} className="px-4 py-2 bg-purple-600 text-white rounded hover:bg-purple-700">Login con Correo</button>
      </div>
    </div>
}
```

```
</div>
}

}

const App = () => (
  return (
    <Router>
      <AppProvider>
        <Header />
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/login" element={<Login />} />
          <Route path="/cart" element={<Cart />} />
          {categories.map(cat => (
            <Route key={cat} path={`/category/${cat}`} element={<Category category={cat} />} />
          ))}
        </Routes>
      </AppProvider>
    </Router>
  )
)

export default App;
```

8. Rúbrica y Cumplimiento

Criterio	Cumplimiento
Desarrolla correctamente el Producto Backlog	<input checked="" type="checkbox"/> Historias priorizadas y estimadas, organizadas por sprint
Desarrolla las fichas de backlog	<input checked="" type="checkbox"/> Cada historia con esfuerzo, prioridad, estado y comentario
Desarrolla la calendarización de los Sprint	<input checked="" type="checkbox"/> Fechas, tareas, objetivos y asignación de recursos
Funcionalidad del código del sprint	⚠ Código en local, funcionalidad explicada, se relaciona con objetivos
Usa correctamente la gramática y ortografía	<input checked="" type="checkbox"/> Revisado, sin errores relevantes
Utiliza correctamente las referencias bibliográficas en formato APA	<input checked="" type="checkbox"/> Todas las fuentes citadas correctamente en la sección final

Referencias

1. Beck, K., et al. (2001). *Manifiesto Ágil para el Desarrollo de Software*. Agile Alliance. <https://agilemanifesto.org/iso/es/manifesto.html>
2. Cohn, M. (2005). *Agile Estimating and Planning*. Prentice Hall.
3. Cohn, M. (2009). *Succeeding with Agile*. Addison-Wesley.
4. Kniberg, H. (2015). *Scrum and XP from the Trenches* (2nd ed.). C4Media.
5. Norman, D. (2013). *The Design of Everyday Things*. Basic Books.
6. Pichler, R. (2016). *Agile Product Management with Scrum* (2nd ed.). Addison-Wesley.
7. Project Management Institute. (2021). *PMBOK Guide* (7th ed.). PMI.