**Épica 1: Gestión de Usuarios**

**Historia 1: Inicio de sesión de usuarios**

* **Como** cliente registrado,
* **Quiero** poder iniciar sesión en mi cuenta,
* **Para** acceder a mi historial de pedidos y realizar nuevas compras.
* **Criterios de aceptación**:
  + El sistema debe validar correctamente el correo y la contraseña.
  + Mostrar mensajes de error si las credenciales son incorrectas.
  + Redirigir al cliente a la página de inicio tras un inicio de sesión exitoso.

**Historia 2: Registro de nuevos usuarios**

* **Como** nuevo cliente,
* **Quiero** registrarme en la plataforma,
* **Para** poder realizar pedidos y guardar mis datos de contacto.
* **Criterios de aceptación**:
  + Los campos obligatorios incluyen nombre, correo y contraseña.
  + Mostrar mensajes de error si los datos son incompletos o inválidos.
  + Confirmar el registro exitoso y redirigir al cliente al inicio de sesión.

**Épica 2: Gestión de Productos**

**Historia 3: Explorar productos disponibles**

* **Como** cliente,
* **Quiero** ver una lista de los productos disponibles,
* **Para** decidir qué comprar.
* **Criterios de aceptación**:
  + Mostrar los productos con nombre, precio e imagen.
  + Los productos deben estar organizados de manera atractiva.

**Historia 4: Añadir productos al carrito**

* **Como** cliente,
* **Quiero** poder añadir productos al carrito desde la página de productos,
* **Para** preparar mi compra de manera sencilla.
* **Criterios de aceptación**:
  + Cada producto debe tener un botón para añadirlo al carrito.
  + Los productos añadidos deben reflejarse inmediatamente en el carrito.

**Historia 5: Visualizar detalles de los productos**

* **Como** cliente,
* **Quiero** ver más información sobre un producto,
* **Para** entender sus características antes de comprarlo.
* **Criterios de aceptación**:
  + Cada producto debe tener una página o modal con detalles adicionales como descripción, ingredientes y tamaño.
  + Mostrar un botón para añadir al carrito desde la vista detallada.

**Épica 3: Gestión del Carrito**

**Historia 6: Ver el contenido del carrito**

* **Como** cliente,
* **Quiero** visualizar los productos en mi carrito,
* **Para** revisar lo que planeo comprar antes de realizar el pago.
* **Criterios de aceptación**:
  + Mostrar el nombre, cantidad y precio de cada producto.
  + Calcular y mostrar el precio total del carrito.

**Historia 7: Modificar el contenido del carrito**

* **Como** cliente,
* **Quiero** poder eliminar productos del carrito,
* **Para** ajustar mi pedido antes de pagar.
* **Criterios de aceptación**:
  + Cada producto debe tener un botón para eliminarlo del carrito.
  + El precio total debe actualizarse tras la eliminación de un producto.

**Historia 8: Finalizar la compra**

* **Como** cliente,
* **Quiero** proceder al pago desde el carrito,
* **Para** completar mi compra de manera segura y rápida.
* **Criterios de aceptación**:
  + El botón de "Finalizar compra" debe redirigir a una página de pago.
  + Mostrar un mensaje de confirmación al completar el pago.

**Épica 4: Gestión de Pedidos**

**Historia 9: Ver historial de pedidos**

* **Como** cliente registrado,
* **Quiero** acceder a un historial de mis compras anteriores,
* **Para** revisar mis pedidos y volver a pedir fácilmente.
* **Criterios de aceptación**:
  + Mostrar una lista de pedidos previos con fecha, productos y precios.
  + Ofrecer un botón para "Reordenar" productos de un pedido anterior.

**Historia 10: Seguimiento de pedidos en tiempo real**

* **Como** cliente,
* **Quiero** rastrear el estado de mi pedido después de comprarlo,
* **Para** saber cuándo llegará.
* **Criterios de aceptación**:
  + Mostrar estados como "Preparando", "En camino" y "Entregado".
  + Actualizar el estado en tiempo real y enviar notificaciones visuales.

**3. Tecnología para aplicar**

**Herramientas y Lenguajes de Programación:**

* **Frontend**:
  + **HTML5**: Para la estructura de las páginas.
  + **CSS3**: Para el diseño y estilos de las páginas.
  + **JavaScript (Vanilla JS)**: Para la lógica y funcionalidad del cliente.
* **Backend**:
  + **Node.js con Express.js**: Para gestionar las APIs del servidor.
* **Base de datos**:
  + **MongoDB**: Para almacenar información de usuarios, productos, pedidos y carritos.
* **Herramientas de gestión**:
  + **Git y GitHub**: Control de versiones.
  + **Jira o Azure DevOps**: Gestión de tareas y seguimiento del proyecto.
  + **Postman**: Pruebas de APIs.

**4. Objetivo del Proyecto**

El sistema tiene como finalidad ofrecer una plataforma de comercio electrónico para la venta de comidas. Permitirá a los usuarios:

1. Registrarse e iniciar sesión.
2. Explorar y añadir productos a un carrito.
3. Realizar pagos de forma segura.
4. Gestionar pedidos y rastrearlos.

**5. Alcance del Proyecto**

**Inclusiones:**

* Registro e inicio de sesión de usuarios.
* Catálogo de 10 productos con imágenes y descripciones.
* Carrito de compras funcional con adición y eliminación de productos.
* Página de pago con resumen del pedido.
* Historial de pedidos y sistema de rastreo básico.

**Exclusiones:**

* Integración con servicios de entrega en tiempo real.
* Métodos de pago avanzados (se usará simulación para la primera versión).

**6. Cronograma del Proyecto**

| **Actividad** | **Plazo** | **Responsable** |
| --- | --- | --- |
| Diseño de la interfaz | 1 semana | Diseñador |
| Desarrollo del frontend | 2 semanas | Desarrollador 1 |
| Configuración del backend | 1 semana | Desarrollador 2 |
| Desarrollo de la base de datos | 1 semana | DBA |
| Integración del sistema | 1 semana | Equipo completo |
| Pruebas y ajustes | 1 semana | QA |

**7. Definición del Primer Release**

**Funcionalidades:**

1. Registro e inicio de sesión de usuarios.
2. Catálogo de productos con vista detallada.
3. Carrito funcional.
4. Página de pago con validación.
5. Historial de pedidos.

**Requerimientos Funcionales:**

* Registro de usuarios.
* Visualización de productos.
* Carrito de compras.
* Pago y confirmación.

**Requerimientos No Funcionales:**

* Interfaz intuitiva y rápida.
* Soporte para dispositivos móviles (responsive design).
* Seguridad en la gestión de contraseñas.

**Metodología Scrum**

**1. Definir Tareas a Ejecutar**

* Crear diseño de las páginas.
* Desarrollar componentes del frontend.
* Configurar API para el backend.
* Integrar base de datos.
* Pruebas unitarias y de integración.

**2. Definir el Equipo de Trabajo**

| **Rol** | **Habilidades Requeridas** | **Responsabilidades** |
| --- | --- | --- |
| Product Owner | Gestión de requisitos | Priorizar épicas |
| Scrum Master | Gestión ágil | Facilitar ceremonias |
| Desarrolladores | HTML, CSS, JavaScript, Node.js | Codificación |
| QA | Pruebas manuales y automatizadas | Validar calidad |

**3. Herramientas**

* **Gestión**: Jira, Azure DevOps.
* **Versionamiento**: GitHub.
* **Diseño**: Figma.
* **Pruebas**: Selenium, Postman.

**4. Definir las Épicas**

* Gestión de usuarios.
* Gestión de productos.
* Gestión del carrito.
* Gestión de pedidos.

**5. Ceremonias de Scrum**

| **Ceremonia** | **Frecuencia** | **Duración** |
| --- | --- | --- |
| Sprint Planning | Inicio de cada sprint | 1 hora |
| Daily Stand-Up | Diario | 15 minutos |
| Sprint Review | Final de sprint | 1 hora |
| Sprint Retrospective | Final de sprint | 45 minutos |

**6. Historias de Usuario**

Ya están definidas anteriormente. Asignar puntos de historia según complejidad.

**3. Plan de Pruebas**

**1. Lista de Requerimientos Funcionales y No Funcionales**

* Funcionales: Registro, inicio de sesión, carrito, pago.
* No funcionales: Velocidad, seguridad, interfaz responsiva.

**2. Criterios de Aceptación y Rechazo**

* Aceptación: Funcionalidad trabaja como se espera.
* Rechazo: Bugs o resultados inesperados.

**3. Herramientas de Pruebas**

* Selenium: Automatización.
* Postman: Pruebas de API.
* BrowserStack: Compatibilidad.

**4. Cronograma de Ejecución de Pruebas**

| **Actividad** | **Plazo** | **Responsable** |
| --- | --- | --- |
| Pruebas Unitarias | 1 semana | QA |
| Pruebas de Integración | 1 semana | QA |

**5. Plantillas para Casos de Pruebas**

* **ID Prueba**: Único.
* **Descripción**: Resumen.
* **Criterios**: Aceptación/Rechazo.
* **Resultado esperado**: Lo que debe ocurrir.

**6. Equipos de Pruebas y Responsabilidades**

| **Miembro** | **Responsabilidad** |
| --- | --- |
| QA | Pruebas manuales. |
| Desarrollador | Solución de bugs. |

**7. Plan de Automatización de Pruebas**

* **Herramienta**: Selenium.
* **Estrategia**: Scripts para pruebas repetitivas (inicio de sesión, agregar al carrito).