**Backend:**

1. **Lenguaje de Programación:**
   * **Node.js:** Es una opción sólida para construir el backend debido a su eficiencia y escalabilidad. Además, permite utilizar JavaScript tanto en el frontend como en el backend, lo que facilita la sincronización entre ambas partes del proyecto.
2. **Framework de Aplicaciones Web:**
   * **Express.js:** Es un framework de Node.js que simplifica el desarrollo de aplicaciones web y proporciona una amplia variedad de funcionalidades para construir APIs RESTful de manera rápida y sencilla.
3. **Base de Datos:**
   * **MongoDB:** Es una base de datos NoSQL que se integra bien con Node.js y es adecuada para proyectos donde se manejan datos no estructurados o semiestructurados, como historias de usuario y tareas en el Backlog.

**Frontend:**

1. **Lenguaje de Programación:**
   * **JavaScript (ES6+):** Es el lenguaje principal para el desarrollo frontend, especialmente cuando se utiliza en conjunto con frameworks modernos como React o Vue.js.
2. **Framework de JavaScript:**
   * **React.js o Vue.js:** Ambos son frameworks populares y potentes para construir interfaces de usuario interactivas y dinámicas. Tienen una curva de aprendizaje relativamente baja y una gran comunidad de desarrolladores, lo que facilita el desarrollo y la resolución de problemas.
3. **Gestión de Estado:**
   * **Redux (para React) o Vuex (para Vue.js):** Estas bibliotecas ayudan a manejar el estado de la aplicación de manera centralizada, lo que es útil para gestionar la información del Backlog y mantener la coherencia de los datos en toda la aplicación.
4. **Estilos y Diseño:**
   * **CSS con preprocesador (SASS o LESS):** Para una mejor organización y mantenibilidad de estilos.
   * **Bootstrap o Material-UI:** Estos son frameworks de diseño que ofrecen componentes preestilizados y un diseño receptivo que puede acelerar el desarrollo frontend.

**Pruebas y Despliegue:**

* **Pruebas:** Para pruebas unitarias y de integración, puedes usar frameworks como Jest (para pruebas de JavaScript) y Supertest (para pruebas de API).
* **Despliegue:** Docker es una excelente opción para contenerizar la aplicación y facilitar su despliegue. Además, herramientas como AWS, Google Cloud Platform o Heroku pueden ser útiles para el despliegue en la nube.