# **Herramientas**

**1. Herramientas para Documentación de Requisitos**

**Matriz de Evaluación**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Criterio** | **Peso** | **Jira + Confluence (0-10)** | **Po** | **Azure DevOps (0-10)** | **Po** | **GitHub Issues + Wiki (0-10)** | **Po** | **Google Docs + Sheets (0-10)** | **Po** | **Microsoft Office 365**  **(0-10)** | **Po** |
| **Facilidad de Uso** | 4 | 8 | 32 | 8 | 32 | 10 | 40 | 10 | 40 | 8 | 32 |
| **Colaboración** | 4 | 10 | 40 | 10 | 40 | 8 | 32 | 10 | 40 | 8 | 32 |
| **Integración** | 5 | 10 | 50 | 10 | 50 | 6 | 30 | 6 | 30 | 10 | 50 |
| **Trazabilidad** | 5 | 10 | 50 | 10 | 50 | 6 | 30 | 4 | 20 | 3 | 15 |
| **Control de Versiones** | 4 | 8 | 32 | 10 | 20 | 10 | 40 | 6 | 24 | 6 | 24 |
| **Costo** | 3 | 6 | 18 | 8 | 24 | 10 | 40 | 10 | 30 | 6 | 18 |
| **Escalabilidad** | 4 | 10 | 40 | 10 | 40 | 8 | 32 | 6 | 24 | 8 | 32 |
| **Ponderado** | 29 (Suma de Po) |  | 262 |  | 266 |  | 244 |  | 208 |  | 203 |

**Justificación**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criterio** | **Jira + Confluence** | **Azure DevOps** | **GitHub Issues + Wiki** | **Google Docs + Sheets** | **Microsoft Office 365** |
| **Facilidad de Uso** | Requiere familiarización con su estructura de proyectos y configuración. No es muy intuitiva para principiantes. | Interfaz moderna y relativamente sencilla para equipos técnicos. | Intuitiva para usuarios de GitHub. | Muy fácil de usar, familiar para la mayoría. | Familiar para muchos usuarios. |
| **Colaboración** | Alta colaboración entre equipos con historial de cambios y comentarios. | Muy buena colaboración con boards ágiles, work items y comentarios. | Buena colaboración mediante issues y comentarios. | Excelente en tiempo real. | Buena, pero no tan fluida como Google Docs. |
| **Integración** | Se integra bien con otros productos de Atlassian y herramientas de desarrollo. | Excelente integración con Git, CI/CD, VS Code, GitHub, etc. | Limitada comparada con otras herramientas más específicas. | Limitada con herramientas de desarrollo. | Limitada en entornos de desarrollo puro. |
| **Trazabilidad** | Excelente trazabilidad entre requisitos, tareas y bugs. | Alta trazabilidad entre requerimientos, código y pruebas. | Limitada sin plugins o configuraciones adicionales. | Manual, difícil de mantener en proyectos grandes. | Muy baja. |
| **Control de Versiones** | Limitado, depende de integraciones externas como Bitbucket. | Soporte nativo para Git. | Nativo en el repositorio. | Básico mediante historial de versiones. | Disponible con OneDrive, pero poco ágil. |
| **Costo** | Licencia paga, aunque con versión gratuita limitada. | Versión gratuita robusta para equipos pequeños. | Gratuito para la mayoría de los casos. | Gratuito. | Requiere suscripción. |
| **Escalabilidad** | Escalable para entornos empresariales complejos. | Altamente escalable para proyectos grandes y organizaciones. | Útil para proyectos medianos o personales. | Poco adecuado para proyectos complejos. | Más útil en entornos corporativos no técnicos. |

## **Conclusión**

**Azure DevOps** es la herramienta más equilibrada, combinando trazabilidad, colaboración y control de versiones con una curva de aprendizaje accesible. Supera a herramientas más básicas como Google Docs o GitHub Wiki en funcionalidad técnica, y es más integrada y gratuita en comparación con Jira.

**2. Herramientas para Arquitectura y Diseño**

**Matriz de Evaluación**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Criterio** | **Peso** | **Lucidchart (0-10)** | **Po** | **Draw.io (diagrams.net) (0-10)** | **Po** | **MS Visio (0-10)** | **Po** | **PlantUML (0-10)** | **Po** | **Miro (0-10)** | **Po** |
| **Facilidad de Uso** | 4 | 10 | 40 | 8 | 32 | 6 | 24 | 4 | 16 | 10 | 40 |
| **Colaboración** | 3 | 10 | 30 | 8 | 24 | 6 | 18 | 6 | 18 | 10 | 30 |
| **Integración** | 5 | 8 | 40 | 8 | 40 | 8 | 40 | 10 | 50 | 6 | 30 |
| **Trazabilidad** | 5 | 6 | 30 | 4 | 20 | 2 | 10 | 8 | 40 | 4 | 20 |
| **Control de Versiones** | 4 | 8 | 32 | 10 | 40 | 6 | 24 | 10 | 40 | 6 | 24 |
| **Costo** | 3 | 6 | 18 | 10 | 30 | 4 | 12 | 10 | 30 | 6 | 18 |
| **Escalabilidad** | 4 | 8 | 32 | 8 | 24 | 8 | 32 | 10 | 40 | 8 | 32 |
| **Ponderado** | 28 (Suma de Po) |  | 222 |  | 210 |  | 170 |  | 234 |  | 194 |

**Justificación**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criterio** | **Lucidchart** | **Draw.io (diagrams.net)** | **MS Visio** | **PlantUML** | **Miro** |
| **Facilidad de Uso** | Muy fácil e intuitivo. | Muy intuitivo, sin curva de aprendizaje. | Interfaz más técnica, puede ser compleja. | Requiere conocimientos técnicos (código). | Muy visual e intuitivo. |
| **Colaboración** | Colaboración en tiempo real con permisos detallados. | Permite edición en conjunto vía Google Drive o GitHub. | Limitada en comparación con herramientas web. | A través de repositorios Git. | Excelente para equipos creativos. |
| **Integración** | Buena con Google Drive, Slack, Confluence. | Compatible con plataformas comunes. | Funciona bien con Microsoft 365. | Excelente con herramientas de CI/CD y documentación. | Múltiples plugins disponibles. |
| **Trazabilidad** | Limitada. | Muy básica. | No especializada. | Alta, al estar basado en texto. | Limitada para uso técnico. |
| **Control de Versiones** | Versión básica de historial. | Depende del almacenamiento externo (Google Drive, GitHub). | Limitado, depende del almacenamiento. | Nativo al trabajar con Git. | Muy básico. |
| **Costo** | Tiene plan gratuito limitado. | Completamente gratuito. | Requiere licencia de pago. | Gratuito y open source. | Plan gratuito con restricciones. |
| **Escalabilidad** | Apto para equipos pequeños y medianos. | Apto para cualquier tipo de proyecto. | Apto para grandes empresas. | Muy alta. | Bueno para brainstorming, limitado para documentación técnica. |

## **Conclusión**

PlantUML ofrece trazabilidad, control de versiones y automatización. Draw.io, por su parte, brinda facilidad y colaboración visual. La combinación cubre necesidades técnicas y de diseño conceptual, superando a herramientas con licencias costosas o con funciones limitadas como Visio o Miro.

**3. Herramientas para Informes de Pruebas**

**Matriz de Evaluación**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Criterio** | **Peso** | **Jira + Xray (0-10)** | **Po** | **GitHub Issues + Actions (0-10)** | **Po** | **Confluence + TestRail (0-10)** | **Po** | **Excel / Google Sheets (0-10)** | **Po** | **Allure Reports (0-10)** | **Po** |
| **Facilidad de Uso** | 4 | 8 | 32 | 8 | 32 | 6 | 24 | 10 | 40 | 6 | 24 |
| **Colaboración** | 3 | 10 | 30 | 8 | 24 | 8 | 24 | 6 | 18 | 6 | 18 |
| **Integración** | 5 | 10 | 50 | 10 | 50 | 8 | 40 | 4 | 20 | 10 | 50 |
| **Trazabilidad** | 5 | 10 | 50 | 8 | 40 | 10 | 50 | 2 | 10 | 8 | 40 |
| **Control de Versiones** | 4 | 8 | 32 | 10 | 40 | 6 | 24 | 4 | 16 | 8 | 32 |
| **Costo** | 3 | 4 | 12 | 10 | 30 | 4 | 12 | 10 | 30 | 10 | 30 |
| **Escalabilidad** | 4 | 10 | 40 | 8 | 32 | 10 | 40 | 4 | 16 | 8 | 32 |
| **Ponderado** | 28 (Suma de Po) |  | 246 |  | 248 |  | 214 |  | 150 |  | 226 |

**Recomendación: GitHub Issues + Actions**

**Justificación**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criterio** | **Jira + Xray** | **GitHub Issues + Actions** | **Confluence + TestRail** | **Excel / Google Sheets** | **Allure Reports** |
| **Facilidad de Uso** | Requiere configuración avanzada. | Intuitiva para usuarios de GitHub. | Requiere familiarización. | Muy fácil. | Requiere configuración técnica. |
| **Colaboración** | Muy buena integración con equipos ágiles. | Muy buena, especialmente en proyectos open source. | Buena, especialmente en grandes equipos. | Google Sheets destaca en este aspecto. | Limitada. |
| **Integración** | Alto nivel con pipelines y otros plugins. | Altísima con CI/CD y testing automático. | Limitada fuera del ecosistema Atlassian. | Mínima. | Excelente con entornos de pruebas automatizadas. |
| **Trazabilidad** | Alta, enlaza pruebas con historias y errores. | Muy buena, desde código hasta ejecución. | Buena, pero con configuración adicional. | Muy baja. | Alta si se integra correctamente. |
| **Control de Versiones** | Limitado. | Totalmente integrado con Git. | Limitado. | Básico. | Depende del entorno de pruebas. |
| **Costo** | Requiere licencia para Xray. | Gratuito. | Licencia de TestRail es elevada. | Gratuito. | Gratuito. |
| **Escalabilidad** | Apto para grandes empresas. | Funciona en proyectos de todos los tamaños. | Alta. | Difícil de mantener en proyectos grandes. | Buena para entornos automatizados. |

## **Conclusión**

GitHub Issues + Actions combina automatización, trazabilidad y colaboración sin costos adicionales. Es ideal para equipos modernos que buscan eficiencia en pruebas sin comprometer integración ni control.

**4. Herramientas para Documentación de Usuario**

**Matriz de Evaluación**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Criterio** | **Peso** | **GitBook (0-10)** | **Po** | **Confluence (0-10)** | **Po** | **MkDocs + GitHub Pages (0-10)** | **Po** | **Notion (0-10)** | **Po** | **Swagger/OpenAPI (0-10)** | **Po** |
| **Facilidad de Uso** | 4 | 10 | 40 | 8 | 32 | 6 | 24 | 10 | 40 | 6 | 24 |
| **Colaboración** | 3 | 8 | 24 | 10 | 30 | 6 | 18 | 10 | 30 | 6 | 18 |
| **Integración** | 4 | 8 | 32 | 8 | 32 | 10 | 40 | 6 | 24 | 10 | 40 |
| **Trazabilidad** | 3 | 6 | 18 | 6 | 18 | 6 | 18 | 4 | 12 | 8 | 24 |
| **Control de Versiones** | 3 | 8 | 24 | 6 | 18 | 10 | 30 | 4 | 12 | 8 | 24 |
| **Costo** | 3 | 6 | 18 | 6 | 18 | 10 | 30 | 8 | 24 | 10 | 30 |
| **Escalabilidad** | 4 | 8 | 32 | 10 | 40 | 8 | 32 | 8 | 32 | 8 | 32 |
| **Ponderado** | 24 (Suma de Po) |  | 188 |  | 188 |  | 192 |  | 174 |  | 200 |

**Recomendación: MkDocs + Swagger**

**Justificación**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criterio** | **GitBook** | **Confluence** | **MkDocs + GitHub Pages** | **Notion** | **Swagger / OpenAPI** |
| **Facilidad de Uso** | Muy intuitivo, ideal para principiantes. | Requiere familiarización con su estructura. | Markdown sencillo, pero requiere conocimientos técnicos y manejo de Git. | Extremadamente fácil e intuitivo para cualquier usuario. | Técnica; requiere conocimientos sobre APIs y definición de endpoints. |
| **Colaboración** | Permite colaboración básica; limitada en el plan gratuito. | Alta colaboración con control de versiones y permisos. | Permite colaboración a través de repositorios Git, no en tiempo real. | Excelente colaboración en tiempo real con comentarios y menciones. | Puede compartirse, pero no permite colaboración simultánea en la edición. |
| **Integración** | Se conecta fácilmente con plataformas como Google Drive o Slack. | Se integra con herramientas del ecosistema Atlassian. | Alta integración con GitHub, CI/CD y despliegue automático (GitHub Pages). | Integración limitada con herramientas técnicas o pipelines de desarrollo. | Se integra con herramientas de desarrollo, APIs, pruebas automatizadas y Swagger UI. |
| **Trazabilidad** | Proporciona trazabilidad básica de cambios. | Buena trazabilidad si se configura correctamente. | Permite trazabilidad a través del control de versiones de Git. | Escasa; no ofrece herramientas específicas de seguimiento técnico. | Excelente trazabilidad técnica, especialmente para endpoints, respuestas y versiones. |
| **Control de Versiones** | Historial básico de edición, útil para cambios simples. | Ofrece historial, aunque limitado fuera de su ecosistema. | Control completo mediante Git, permite revertir o comparar versiones fácilmente. | Muy limitado, con opciones básicas de historial. | Control eficaz usando Git para mantener historial de documentación técnica. |
| **Costo** | Plan gratuito con funciones básicas, ampliables mediante suscripción. | Requiere licencia de pago para acceder a todas sus funcionalidades. | Gratuito y open source; se despliega sin costos adicionales en GitHub Pages. | Gratuito con funciones ampliadas en plan pago. | Completamente gratuito y de código abierto, sin restricciones de uso. |
| **Escalabilidad** | Escalable para proyectos pequeños y medianos. | Altamente escalable para empresas grandes con necesidades complejas. | Escalable y apto para documentaciones amplias y estructuradas técnicamente. | Escalable para documentación general, pero no para entornos técnicos avanzados. | Muy escalable y ampliamente adoptado en entornos empresariales y proyectos grandes. |

## **Conclusión**

La combinación MkDocs + Swagger cubre la documentación tanto general como técnica. Permite control total de versiones, despliegue automático y documentación interactiva para usuarios técnicos y no técnicos, siendo una opción potente, gratuita y profesional.