**UNIVERSIDAD PONTIFICIA CATOLICA DEL ECUADOR**

**WEB PARA GESTIONAR RESERVAS Y ADMINISTRACIÓN DE SALAS DE REUNIONES**

**NOMBRES**

**GERASAIR MALDONADO**

**ADAM MERO**

**13/07/2025**

**ÍNDICE DE CONTENIDO**

[1. ¿Por qué usar Jira en este proyecto? 3](#_heading=h.y887e7964bxq)

1.1 [Metodología Ágil y Scrum 3](#_heading=h.6whoy7ps0pwi)

1.2 [Seguimiento y Trazabilidad 3](#_heading=h.yprllah9p9ga)

1.3 [Integraciones Técnicas 3](#_heading=h.o79qbh9jk8ms)

1.4 [Visualización y Reportes 4](#_heading=h.u4ump2xue2a4)

2. [Gestión del Proyecto con Jira 5](#_heading=h.nsdrfunf2db9)

2.1 [Tableros y Flujos de Trabajo 5](#_heading=h.jigc5h6ibeuf)

2.2 [Organización de Issues 5](#_heading=h.9l76sju8e2tz)

3. [Conclusiones 7](#_heading=h.coqvz14g4l2r)

4. [Recomendaciones 8](#_heading=h.ao4a8zpnz7yk)

5. [Bibliografía 9](#_heading=h.n8ir76q3p0lb)

**ÍNDICE DE TABLAS**

[Clasificación mediante Etiquetas 6](#_heading=h.r92ca6btr00t)

[Seguimiento y Métricas 7](#_heading=h.pj2bkzicg4gp)

[Riesgos y Escalamiento 7](#_heading=h.295sazn2z2op)

**ÍNDICE DE FIGURAS**

[IMÁGENES 11](#_heading=h.aqbgzwhm8j9l)

# **¿Por qué usar Jira en este proyecto?**

El desarrollo del sistema SGSR —una aplicación de reservas de salas de reuniones— requiere una coordinación precisa de tareas, equipos y entregables. Jira fue elegido como plataforma de gestión por sus capacidades ágiles, trazabilidad, visualización, integración y escalabilidad.

## **Metodología Ágil y Scrum**

Jira permite implementar metodologías ágiles, especialmente Scrum, con funciones como:

* **Sprints de 2 semanas** con historias de usuario entregables.
* **Planificación colaborativa** a través de boards visuales.
* **Transparencia** para todos los miembros y partes interesadas.

**Figura 1**: Flujo de trabajo Scrum en Jira

## **Seguimiento y Trazabilidad**

* Cada historia de usuario está conectada con tareas, código (vía Git), pruebas y revisiones.
* Los cambios se documentan en tiempo real, facilitando auditoría y mantenimiento.

## **Integraciones Técnicas**

Jira se integró con herramientas como:

* **GitHub** (commits vinculados a issues)
* **CI/CD** con GitHub Actions
* **Testing** con Cypress, Jest

## **Visualización y Reportes**

Con dashboards personalizables se analizaron:

* Progreso por sprint
* Tiempos invertidos
* Cuellos de botella
* Calidad del código

**Figura 2**: Dashboard de métricas

# **Gestión del Proyecto con Jira**

## **Tableros y Flujos de Trabajo**

Se implementaron tableros **Scrum** con columnas:

* **Backlog** → **Sprint Ready** → **In Progress** → **Code Review** → **Testing** → **Done**

**Figura 3**: Ejemplo de tablero Kanban en Jira

## **Organización de Issues**

* **Epics**: Agrupaciones por módulo (reservas, calendario, reportes)
* **User Stories**: Casos funcionales centrados en el usuario
* **Tareas Técnicas** y **Bugs**

### 

### 

### 

### **Clasificación mediante Etiquetas**

**Tabla 2**: Clasificación de etiquetas en Jira  
 | Tipo | Ejemplo |  
  
 | Función | frontend, backend, database |  
 | Prioridad | critical, high, low |  
 | Componente | calendar-view, reservation-system |

### **Seguimiento y Métricas**

**Tabla 1**: Velocity por sprint  
 | Sprint | Puntos Completados |  
 |--------|--------------------|  
 | 1 | 34 |  
 | 2 | 38 |  
 | 3 | 42 |  
 | 4 | 36 |

Además:

* **Burndown charts** para seguimiento diario
* **Dashboards personalizados**
* **Time tracking** para analizar costos y productividad

### **Riesgos y Escalamiento**

Jira permitió:

* Reportar impedimentos rápidamente
* Escalar automáticamente al Scrum Master
* Resolver bloqueos antes de afectar el sprint

# **Conclusiones**

1. **Jira facilitó la coordinación ágil** del equipo mediante visibilidad y automatización.
2. Se logró **trazabilidad completa** desde requisitos hasta código.
3. **Los cambios en el proyecto** se manejaron con control y sin afectar la planificación.
4. **La calidad del software** mejoró gracias a integraciones CI/CD y seguimiento de bugs.
5. **Los stakeholders** accedieron a información en tiempo real, aumentando la confianza.
6. Se optimizó la **asignación de recursos y esfuerzo** gracias a métricas claras.
7. El proceso fue **escalable**, integrando nuevos miembros sin pérdida de rendimiento.

# **Recomendaciones**

1. **Invertir tiempo en la configuración inicial** de Jira: flujos, permisos y tableros.
2. Mantener **workflows simples pero efectivos**.
3. Establecer una **Definition of Done clara y compartida** por todo el equipo.
4. Usar **métricas clave (KPIs)** relevantes y evitar la sobrecarga de dashboards.
5. Automatizar transiciones, pero con sentido práctico.
6. Revisar periódicamente la herramienta y adaptarla al crecimiento del proyecto.
7. Formar un equipo de **gobernanza y mejora continua** sobre el uso de Jira.

## 

## 

# **Bibliografía**

1. Atlassian. (2024). *Jira Software Documentation*.<https://confluence.atlassian.com/jira/>
2. Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org
3. Cohn, M. (2010). *Succeeding with Agile*. Addison-Wesley
4. Rubin, K. (2012). *Essential Scrum*. Addison-Wesley
5. Atlassian. (2024). *Agile Project Management with Jira*.<https://www.atlassian.com/team-playbook/>
6. Stellman, A., & Greene, J. (2014). *Learning Agile*. O’Reilly
7. Beck, K., et al. (2001). *Manifesto for Agile Software Development*.<https://agilemanifesto.org>

#### **IMÁGENES**





