



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

***Proyecto **Aplicativo Web para la Gestión de
Acervos Digitales en el Gobierno Regional de
Tacna*****

Curso: Construcción de Software II

Docente: Mag. Ricardo Eduardo Valcárcel Alvarado

Integrantes:

<i>Arenas Paz Soldan, Miguel Jesús</i>	<i>(2017059282)</i>
<i>Céspedes Medina, Christian Alexander</i>	<i>(2010036257)</i>

**Tacna – Perú
2025**

***Sistema *Aplicativo Web para la Gestión de Acervos**
Digitales en el Gobierno Regional de Tacna
Documento Informe de Proyecto utilizando
metodología ágil SCRUM**

Versión {1.0}

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MAPS, CACM	CACM	MAPS	17/06/2025	Versión Base

INDICE GENERAL

Objetivos:	5
1. Introducción	6
1.1. Propósito del documento	6
1.2. Alcance del proyecto	6
1.3. Audiencia	6
1.4. Estructura del documento	6
2. Visión	7
2.1. Resumen del producto	7
2.2. Principales características del producto	7
2.3. Objetivos del negocio	7
3. Organización y roles	8
3.1. Equipo SCRUM	8
3.1.1. Product Owner	8
3.1.2. Scrum Master	8
3.1.3. Equipo de Desarrollo	8
3.2. Stakeholders	8
4. Backlog del Producto	9
4.1. Definición del Product Backlog	9
4.2. Historias de usuario clave	9
4.2.1. Historia de usuario 1	9
4.2.2. Historia de usuario 2	9
4.3. Priorización de las historias de usuario	9
5. Planificación de Sprints	10
5.1. Ciclo de vida de los sprints	10
5.2. Objetivos de Sprint	10
5.3. Sprint Backlog	10

5.4.	<i>Reuniones clave del sprint</i>	11
5.4.1.	<i>Planificación del Sprint (Sprint Planning)</i>	11
5.4.2.	<i>Daily Standup (Daily Scrum)</i>	11
5.4.3.	<i>Revisión del Sprint (Sprint Review)</i>	11
5.4.4.	<i>Retrospectiva del Sprint (Sprint Retrospective)</i>	11
6.	Definición de "Hecho" (Definition of Done - DoD)	12
6.1.	Criterios de aceptación	12
6.2.	Requisitos de calidad	12
7.	Gestión de Impedimentos	13
7.1.	<i>Proceso para identificar y resolver impedimentos</i>	13
7.2.	<i>Escalación de problemas</i>	13
8.	Iteraciones	13
8.2.	<i>Revisión de entregas</i>	14
9.	Métricas y seguimiento de progreso	14
9.1.	<i>Burndown Chart</i>	14
9.2.	<i>Velocidad del equipo</i>	14
9.3.	<i>Revisión de progreso</i>	14
10.	Plan de Pruebas y control de calidad	14
10.1.	<i>Pruebas en el marco de SCRUM</i>	14
10.2.	<i>Automatización de pruebas</i>	14
10.3.	<i>Criterios de aceptación de historias</i>	15
11.	Riesgos y Gestión de cambios	15
11.1.	<i>Riesgos del proyecto</i>	15
11.2.	<i>Gestión del cambio</i>	15
12.	Conclusiones	15
12.1.	<i>Reflexiones finales</i>	16
12.2.	<i>Próximos pasos</i>	16

Informe de Proyecto utilizando metodología ágil SCRUM

La documentación de un proyecto de software utilizando la metodología ágil SCRUM debe ser ligera, enfocada en la colaboración y la entrega continua de valor. A diferencia de las metodologías tradicionales, SCRUM promueve la iteración rápida y la flexibilidad, por lo que la documentación tiende a ser más adaptativa y funcional.

Sin embargo, a pesar de la flexibilidad, es importante que haya documentación que guíe el proyecto y facilite la toma de decisiones, especialmente para garantizar la trazabilidad, la calidad y el cumplimiento de los requisitos. A continuación, se plantea una estructura recomendada para un documento que pueda ser utilizado en un proyecto ágil SCRUM.

Documentación ligera: SCRUM promueve la documentación mínima necesaria, enfocándose más en la comunicación y la colaboración entre los miembros del equipo y stakeholders. Por lo tanto, la documentación debe ser accesible, pero sin ser redundante.

Iteración continua: Como SCRUM es un proceso iterativo, la documentación puede ser actualizada de manera continua y evolutiva según el avance del proyecto.

Esta estructura ayuda a mantener el proyecto ágil, enfocado en la entrega continua de valor y adaptado a los cambios que puedan surgir durante el ciclo de vida del producto.

Objetivos:

- *Proveer una guía para documentar un proyecto de software utilizando una metodología ágil.*
- *Elaborar la documentación en el desarrollo de software utilizando la metodología ágil SCRUM.*

1. Introducción

1.1. Propósito del documento

Este documento tiene como propósito establecer las directrices y lineamientos para el desarrollo del Aplicativo Web para la Gestión de Acervos Digitales en el Gobierno Regional de Tacna utilizando la metodología ágil SCRUM. Sirve como guía de referencia para todos los miembros del equipo de desarrollo, stakeholders y partes interesadas, asegurando una comprensión común del alcance, objetivos y procesos del proyecto.

1.2. Alcance del proyecto

El proyecto busca desarrollar una solución web integral que permita al Gobierno Regional de Tacna digitalizar, organizar, preservar y facilitar el acceso a su patrimonio documental histórico y administrativo.

El sistema permitirá:

- Digitalización y catalogación de documentos históricos
- Búsqueda avanzada de documentos
- Gestión de permisos y accesos
- Preservación digital a largo plazo
- Generación de reportes y estadísticas

1.3. Audiencia

Este documento está dirigido a:

- **Equipo de desarrollo:** Desarrolladores, testers y diseñadores
- **Product Owner:** Representante del Gobierno Regional de Tacna
- **Scrum Master:** Facilitador del proceso ágil
- **StakeHolders:** Directivos del GRT, personal administrativo

1.4. Estructura del documento

El documento se organiza en 12 secciones principales que cubren desde la visión del producto hasta las conclusiones, incluyendo la organización del equipo, planificación de sprints, métricas de seguimiento y gestión de riesgos, siguiendo las mejores prácticas de SCRUM

2. Visión

2.1. Resumen del producto

El Aplicativo Web para la Gestión de Acervos Digitales es una plataforma tecnológica que transformará la manera en que el Gobierno Regional de Tacna gestiona su patrimonio documental.

El sistema resolverá los problemas actuales de:

- Deterioro físico de documentos históricos
- Dificultad de acceso a la información
- Falta de organización sistemática
- Pérdida de documentación importante
- Limitado alcance de consulta ciudadana

Está destinado a funcionarios públicos, investigadores, estudiantes y ciudadanos interesados en acceder al patrimonio documental de la región.

2.2. Principales características del producto

- **Sistema de Catalogación:** Metadatos estructurados según estándares archivísticos.
- **Motor de Búsqueda Avanzada:** Búsqueda por texto completo, fechas, categorías
- **Gestión de Usuario y Permisos:** Control de acceso basado en roles
- **Panel de Administración:** Gestión integral del sistema
- **API REST:** Integración con otros sistemas del GRT
- **Respaldo y Recuperación:** Sistema de backup automático

2.3. Objetivos del negocio

- **Preservación del Patrimonio:** Salvaguardar documentos históricos mediante digitalización
- **Mejora de la Eficiencia:** Reducir en 80% el tiempo de búsqueda de documentos
- **Transparencia:** Facilitar el acceso ciudadano a información pública
- **Modernización:** Posicionar al GRT como referente en gestión documental digital
- **Cumplimiento Normativo:** Alinearse con las políticas nacionales de gobierno digital

3. Organización y roles

3.1. Equipo SCRUM

3.1.1. Product Owner

Responsable: Representante del Área de Archivo Central del GRT

Responsabilidades:

- Definir y priorizar el Product Backlog
- Validar los criterios de aceptación
- Aprobar o rechazar los entregables de cada sprint
- Comunicar la visión del producto al equipo
- Gestionar las expectativas de los stakeholders

3.1.2. Scrum Master

Responsable: Miguel Jesús Arenas Paz Soldan

Responsabilidades:

- Facilitar las ceremonias SCRUM
- Eliminar impedimentos del equipo
- Asegurar el cumplimiento de las prácticas ágiles
- Proteger al equipo de interrupciones externas
- Promover la mejora continua

3.1.3. Equipo de Desarrollo

Integrantes:

- Christian Alexander Céspedes Medina (Desarrollador Full Stack)
- Miguel Jesus Arenas Paz Soldan (Desarrollador Frontend)

Responsabilidades:

- Desarrollo de las funcionalidades
- Pruebas unitarias y de integración
- Documentación técnica
- Estimación de historias de usuario

3.2. Stakeholders

- Director Regional de Administración: Sponsor del proyecto
- Jefe de Archivo Central: Usuario principal
- Personal administrativo: Usuarios operativos
- Investigadores y académicos: Usuarios externos

4. Backlog del Producto

4.1. Definición del Product Backlog

El Product Backlog se organiza como una lista priorizada de funcionalidades, mejoras y correcciones. Se utiliza la técnica MoSCoW (Must have, Should have, Could have, Won't have) para la priorización, considerando:

- Valor de negocio
- Dependencias técnicas
- Riesgos asociados
- Complejidad de implementación

4.2. Historias de usuario clave

4.2.1. Historia de usuario 1: Digitalización de documentos

Como archivista del GRT

Quiero registrar documentos en el sistema

Para mantener un inventario digital organizado y accesible

Criterios de aceptación:

- El sistema permite registrar documentos con información completa
- El sistema genera un código único de identificación
- El sistema permite compartir con otros usuarios
- El sistema encripta los archivos para mayor seguridad

4.2.2. Historia de usuario 2 Búsqueda avanzada

Como archivista del GRT

Quiero registrar documentos en el sistema

Para mantener un inventario digital organizado y accesible

Criterios de aceptación:

- Búsqueda por fecha
- Resultados ordenables y paginados
- Gestionar archivos

4.3. Priorización de las historias de usuario

Prioridad Alta (Must Have):

- Autenticación
- Gestión de usuarios
- Registro de acervos
- Búsqueda básica

5. Planificación de Sprints

5.1. Ciclo de vida de los sprints

- Duración: 2 semanas
- Inicio: lunes
- Fin: viernes de la segunda semana

5.2. Objetivos de Sprint

Sprint 1: Configuración del entorno y autenticación básica

Sprint 2: Módulo de registro y catalogación

Sprint 3: Sistema de metadatos y categorización

Sprint 4: Búsqueda básica y listados

Sprint 5: Gestión de préstamos y consultas

Sprint 6: Panel de administración

Sprint 7: Reportes y estadísticas

Sprint 8: Pruebas finales y despliegue

5.3. Sprint Backlog

Se seleccionan historias del Product Backlog considerando:

- Dependencias técnicas
- Balance entre frontend y backend
- Capacidad real del equipo
- Objetivos del sprint

5.4. Reuniones clave del sprint

5.4.1. Planificación del Sprint (Sprint Planning)

- Duración: 4 horas
- Participantes: Todo el equipo SCRUM
- Agenda:
 - Revisión del Product Backlog
 - Definición del objetivo del sprint
 - Selección de historias
 - Descomposición en tareas
 - Estimación y compromiso

5.4.2. Daily Standup (Daily Scrum)

- Duración: 15 minutos
- Formato: De pie, frente al tablero kanban
- Preguntas:
 - ¿Qué hice ayer?
 - ¿Qué haré hoy?
 - ¿Hay impedimentos?

5.4.3. Revisión del Sprint (Sprint Review)

- Duración: 2 horas
- Participantes: Equipo SCRUM + Stakeholders
- Actividades:
 - Demostración de funcionalidades
 - Feedback de stakeholders
 - Actualización del Product Backlog
 - Métricas del sprint

5.4.4. Retrospectiva del Sprint (Sprint Retrospective)

- Duración: 1.5 horas
- Participantes: Equipo SCRUM
- Formato: Start-Stop-Continue
- Resultado: Plan de mejora para el siguiente sprint

6. Definición de "Hecho" (Definition of Done - DoD)

6.1. Criterios de aceptación

Una historia se considera “hecha” cuando:

- Código:
 - Código completo y funcional
 - Cumple con los estándares de codificación
 - Code Review aprobado
 - Integrado en la rama principal
- Pruebas:
 - Pruebas unitarias
 - Pruebas de integración ejecutadas
 - Pruebas de aceptación
- Documentación:
 - Código comentado
 - Documentación técnica actualizada
 - Manual de usuario actualizado si aplica
- Despliegue:
 - Desplegado en ambiente de staging
 - Sin bugs críticos

6.2. Requisitos de calidad

- **Performance:** Tiempo de respuesta
- **Seguridad:** Cumplimiento OWASP
- **Accesibilidad:** WCAG 2.1 Nivel AA
- **Compatibilidad:** Chrome, Firefox, Edge (últimas 2 versiones)
- **Responsive:** Adaptable a dispositivos móviles

7. Gestión de Impedimentos

7.1. Proceso para identificar y resolver impedimentos

- Identificación:
 - Durante Daily Scrum
 - Tablero de impedimentos visible
 - Comunicación directa al Scrum Master
- Clasificación:
 - Nivel 1: Resoluble por el equipo
 - Nivel 2: Requiere intervención del Scrum Master
 - Nivel 3: Requiere escalación a management
- Resolución:
 - Asignación de responsable
 - Plazo máximo: 48 horas
 - Seguimiento diario

7.2. Escalación de problemas

- Impedimentos técnicos: Arquitecto de Software del GRT
- Impedimentos de recursos: Director de Administración
- Impedimentos de alcance: Comité de Dirección del Proyecto

8. Iteraciones

8.1. Incrementos del producto

Cada sprint entrega un incremento potencialmente desplegable:

- Sprint 1: Sistema de login funcional
- Sprint 2: Capacidad de registrar y almacenar información de acervos
- Sprint 3: Documentos catalogados con metadatos completos
- Sprint 4: Sistema de búsqueda operativo
- Sprint 5: Control de préstamos implementado
- Sprint 6: Administración completa del sistema
- Sprint 7: Sistema con reportes funcionales
- Sprint 8: Producto final en producción

8.2. Revisión de entregas

- Validación funcional por el Product Owner
- Pruebas de aceptación de usuarios
- Verificación de cumplimiento de DoD
- Aprobación formal antes de pasar a producción

9. Métricas y seguimiento de progreso

9.1. Burndown Chart

- Actualización diaria
- Visible en el área de trabajo
- Análisis de tendencias
- Identificación temprana de desvíos

9.2. Velocidad del equipo

- Medición por sprint completado
- Promedio móvil de 3 sprints
- Ajuste de capacidad según velocidad real

9.3. Revisión de progreso

- Semanal: Revisión de burndown
- Por Sprint: Análisis de velocidad y calidad
- Mensual: Reporte de stakeholders
- KPIs principales:
 - Historias completadas vs planificadas
 - Bugs encontrados en producción
 - Satisfacción del Product Owner

10. Plan de Pruebas y control de calidad

10.1. Pruebas en el marco de SCRUM

Estrategia de pruebas:

- Pruebas continuas durante el desarrollo
- TDD (Test Driven Development) cuando aplique
- Pruebas automatizadas como parte del DoD

Tipos de pruebas:

- Unitarias (desarrolladores)
- Integración (automatizadas)

- Sistema (equipo QA)
- Aceptación (Product Owner)
- Rendimiento (herramientas automatizadas)

10.2. Automatización de pruebas

- Framework: Jest para pruebas unitarias
- E2E: Cypress para pruebas end-to-end
- CI/CD: GitHub Actions
- Cobertura mínima: 80%

10.3. Criterios de aceptación de historias

- Funcionalidad completa según descripción
- Sin regresiones en funcionalidades existentes
- Performance dentro de parámetros
- Accesible y usable
- Documentación actualizada

11. Riesgos y Gestión de cambios

11.1. Riesgos del proyecto

Riesgo	Probabilidad	Impacto	Mitigación
Cambios en requerimientos	Alta	Alta	Revisiones frecuentes
Falta de disponibilidad	Media	Alto	Delegación de decisiones
Problemas de infraestructura	Baja	Alto	Plan de contingencia
Resistencia al cambio	Media	Medio	Capacitación temprana

11.2. Gestión del cambio

- Cambios evaluados en Sprint Planning
- No se aceptan cambios durante el sprint
- Impacto analizado antes de aceptar
- Comunicación transparente de consecuencias
- Product Backlog actualizado inmediatamente

12. Conclusiones

12.1. Reflexiones finales

La implementación de SCRUM para este proyecto permite:

- Entregas incrementales de valor
- Adaptación rápida a cambios
- Colaboración estrecha con el GRT
- Transparencia en el progreso
- Mejora continua del proceso

El éxito dependerá del compromiso del equipo y la participación activa de los stakeholders.

12.2. Sprint 5: Control de préstamos **Próximos pasos**

- Inmediato: Kick-off meeting con todos los stakeholders
- Semana 1: Configuración de ambientes de desarrollo
- Semana 2: Sprint Planning del Sprint 1
- Continuo: Refinamiento del Product Backlog
- Mensual: Revisión de la hoja de ruta con stakeholders