

**Proyecto para el desarrollo de la Pagina Web para la
Gestión de Acciones
Estándares**

Versión 0.9

Gestión de Acciones	Versión: 0.9
Documento de Estándares	Fecha: 23/11/2024

Historial de Revisiones

Fecha	Versión	Descripción	Autor
23/11/2024	0.9	Definición de los Estándares Utilizados	Grupo DEVSQUAD: Pamela Cruz Alexandra Martínez Loreley Pazmiño Joffre Salvador

Tabla de Contenidos

Gestión de Acciones	Versión: 0.9
Documento de Estándares	Fecha: 23/11/2024

1	Introducción	4
1.1	Propósito:	4
1.2	Alcance	4
1.3	Audiencia	4
2	Estándares de Codificación	4
2.1	Lenguajes de programación utilizados	4
2.1.1	Formato de código:	4
2.1.2	Comentarios y documentación:	4
2.2	Gestión de Versiones	4
2.2.1	Sistema de control de versiones:	4
2.2.2	Convención de ramas:	4
2.2.3	Commits:	5
2.3	Revisión de Código	5
2.3.1	Proceso de revisión: Explicar cómo y quién revisa los cambios (e.g., pull requests).	5
2.3.2	Criterios de aceptación:	5
2.4	Pruebas	5
2.4.1	Tipos de pruebas: Unitaria, de integración, de sistema, etc.	5
2.4.2	Herramientas de prueba: Enumerar herramientas utilizadas (e.g., JUnit, pytest, Selenium).	5
2.4.3	Cobertura de pruebas:	5
2.5	Documentación	5
2.5.1	Estandarización	5
2.5.2	Contenido requerido:	5
2.6	Gestión de Configuración	5
2.6.1	Archivos de configuración:	5
2.6.2	Variables sensibles	6
2.7	Control de Calidad	6
2.7.1	Herramientas de análisis estático:	6
2.7.2	Criterios de calidad del código	6
2.8	Seguridad	6
2.8.1	Prácticas seguras de desarrollo:	6
2.8.2	Pruebas de seguridad:	6
2.9	Herramientas y Tecnologías	6
2.10	Revisión y Actualización	6
2.10.1	Periodicidad	6
2.10.2	Responsables	6

Gestión de Acciones	Versión: 0.9
Documento de Estándares	Fecha: 23/11/2024

1 Introducción

1.1 Propósito:

El objetivo de este documento es unificar las prácticas de desarrollo dentro del proyecto "Gestión de Acciones", asegurando una alta calidad de código, consistencia en las entregas y facilitando la colaboración entre los miembros del equipo.

1.2 Alcance

Este documento aplica a todas las áreas del proyecto, incluyendo:

- Desarrollo de código.
- Documentación técnica y de usuario.
- Pruebas de calidad y seguridad.
- Gestión de versiones y configuración.

1.3 Audiencia

Los destinatarios principales de este documento son:

- Product owner.
- Scrum Master.
- Equipo de desarrollo (Development team).

2 Estándares de Codificación

2.1 Lenguajes de programación utilizados

- Frontend: HTML, CSS (SASS), JavaScript (React.js).
- Backend: Java (Spring Boot).
- Bases de Datos: MySQL.

2.1.1 Formato de código:

- Estilo de nombres: camelCase para variables y métodos; PascalCase para clases.
- Límites de longitud de línea: 80 caracteres.
- Indentación: 4 espacios por nivel.

2.1.2 Comentarios y documentación:

- Comentarios en línea para lógica compleja o no trivial.
- Uso de Javadoc para documentar métodos, clases y componentes.

2.2 Gestión de Versiones

2.2.1 Sistema de control de versiones:

Herramienta: Git

2.2.2 Convención de ramas:

Branching Strategy

Main Branches (Ramas principales):

- main: Rama principal que contiene la versión estable y lista para producción.
- develop: Rama base para el desarrollo donde se integran nuevas funcionalidades antes de fusionarlas en

Gestión de Acciones	Versión: 0.9
Documento de Estándares	Fecha: 23/11/2024

main.

- Documentation: Rama para subir toda la documentación.

Feature Branches (Ramas de funcionalidades):

- feature/nombre: Nuevas funcionalidades.
- bugfix/nombre: Corrección de errores.

2.2.3 **Commits:**

- Mensajes: [Tipo]: Descripción breve.
Ejemplo: [Feature]: Agregar funcionalidad de registro.
- Documentación: versión_ "Nombre del documento"

2.3 Revisión de Código

2.3.1 **Proceso de revisión:**

- Cambios revisados a través de pull requests en GitHub.
- Cada cambio debe ser aprobado por al menos un miembro del equipo.

2.3.2 **Criterios de aceptación:**

- Pruebas exitosas.
- Cobertura de código mínima del 80%.
- Cumplimiento de los estándares establecidos.

2.4 Pruebas

2.4.1 **Tipos de pruebas:**

- Pruebas Unitarias: Validación de métodos individuales.
- Pruebas de Integración: Validación de componentes trabajando juntos.
- Pruebas de Sistema: Validación de funcionalidades completas dentro del sistema.

2.4.2 **Herramientas de prueba:**

- Las herramientas de prueba serán definidas y configuradas según las necesidades del equipo.

2.4.3 **Cobertura de pruebas:**

- Nivel mínimo aceptable: 80%.

2.5 Documentación

2.5.1 **Estandarización**

Herramientas: Markdown para guías de usuario.
Javadoc para documentar el código fuente.

2.5.2 **Contenido requerido:**

Diagramas UML de arquitectura.
Manuales técnicos.

2.6 Gestión de Configuración

2.6.1 **Archivos de configuración:**

Convención de nombres: Los archivos de configuración deben seguir la convención de nombres **config/{entorno}.properties**.

Gestión de Acciones	Versión: 0.9
Documento de Estándares	Fecha: 23/11/2024

2.6.2 **Variables sensibles**

Almacenadas en archivos .env y encriptadas.

2.7 Control de Calidad

2.7.1 **Herramientas de análisis estático:**

SonarQube para análisis de código estático y evaluación de calidad.

2.7.2 **Criterios de calidad del código**

Complejidad ciclomática menor a 10.

2.8 Seguridad

2.8.1 **Prácticas seguras de desarrollo:**

- Validación de entradas y salidas.
- Uso de contraseñas hashadas con bcrypt.

2.8.2 **Pruebas de seguridad:**

- Escaneo de vulnerabilidades con OWASP ZAP.

2.9 Herramientas y Tecnologías

- IDE: Visual Studio Code para frontend y backend.
- Control de Versiones: GitHub.
- CI/CD: Jenkins.

2.10 Revisión y Actualización

2.10.1 **Periodicidad**

Actualización semanal o cuando se detecten necesidades específicas.

2.10.2 **Responsables**

Equipo DEVSQUAD liderado por Loreley Pazmiño