**Fase de Metodología - Proyecto SpeakUp**

|  |  |
| --- | --- |
| **Autores** | Luis Santiago Forero Heredia, Emerson Messi Vega Bustos, Angel Daniel Canchila Aparicio |
| **Fecha de creación** | 16/09/2025 |
| **Artefacto** | Historias de usuario versión 1.2 |
| **Asunto** | Historias de usuario de proyecto SpeakUp |

**Objetivo General del Proyecto**

|  |  |
| --- | --- |
| **HU01 – Crear denuncia válida** | |
| **Funcionalidad disponible:** Aplicación WEB / APP | |
| **Usuario:** Ciudadano | |
| **Nombre historia:** Crear denuncia | **Prioridad de Negocio:** Alta |
| **Puntos de Estimación: 5** | **Spring #: 1** |
| **Programador Asignado: Luis Santiago Forero Heredia** | |
| **Descripción:**  **Como ciudadano quiero registrar una denuncia con un estado inicial "pendiente" para poder informar situaciones que requieren revisión por parte de un moderador.** | |
| **Criterios de Aceptación:**   * **Al registrar una denuncia debe asignarse automáticamente el estado pendiente.** * **La denuncia debe incluir título, descripción, ubicación, categoría, gravedad y usuario.** * **El sistema debe validar que los campos obligatorios no estén vacíos.**  |  |  | | --- | --- | | **Dado** | que un ciudadano llena el formulario con título, descripción, ubicación, categoría, gravedad y usuario | | **Cuando** | hace clic en “Registrar denuncia” | | **Entonces** | la denuncia debe crearse con estado pendiente | | |
| **Observaciones:**  **Todas las denuncias nuevas inician como pendientes para que un moderador las revise.** | |

|  |  |
| --- | --- |
| **HU02 – Registrar log automáticamente** | |
| **Funcionalidad disponible:** Microservicio de auditoría | |
| **Usuario:** Sistema | |
| **Nombre historia:** Registro automático de logs | **Prioridad de Negocio:** Alta |
| **Puntos de Estimación: 3** | **Spring #: 1** |
| **Programador Asignado: Luis Santiago Forero Heredia** | |
| **Descripción:**  **Como sistema quiero registrar automáticamente un log de las acciones para poder tener un historial de auditoría y trazabilidad de cambios.** | |
| **Criterios de Aceptación:**   * **Al ejecutar cualquier acción importante (ej. creación de usuario), debe registrarse un log con usuarioId, acción y entidad.** * **El log debe incluir fecha y hora del evento.** * **El log debe almacenarse en el repositorio de auditoría.**  |  |  | | --- | --- | | **Dado** | que el sistema ejecuta una acción relevante (ej. creación de usuario) | | **Cuando** | se registra la acción | | **Entonces** | debe guardarse automáticamente un log con usuarioId, acción, entidad y fecha | | |
| **Observaciones:**  Los logs permiten auditar la aplicación y detectar problemas o malas prácticas. | |

|  |  |
| --- | --- |
| **HU03 – Crear notificación** | |
| **Funcionalidad disponible:** Aplicación WEB / Microservicio de notificaciones | |
| **Usuario:** Moderador | |
| **Nombre historia:** Recibir notificación de denuncia | **Prioridad de Negocio:** Alta |
| **Puntos de Estimación: 3** | **Spring #: 2** |
| **Programador Asignado: Luis Santiago Forero Heredia** | |
| **Descripción:**  **Como moderador quiero recibir notificaciones de denuncias nuevas para poder revisarlas oportunamente.** | |
| **Criterios de Aceptación:**   * **Al crear una notificación debe quedar en estado no leído (false).** * **El mensaje de la notificación debe estar asociado a una denuncia específica.** * **La notificación debe mostrar el mensaje enviado por el sistema.**  |  |  | | --- | --- | | **Dado** | que se genera una denuncia que requiere atención de un moderador | | **Cuando** | se crea la notificación | | **Entonces** | debe guardarse con el campo leído = false y mostrar el mensaje correspondiente | | |
| **Observaciones:**  **Las notificaciones no leídas son clave para que el moderador actúe rápidamente.** | |

|  |  |
| --- | --- |
| **HU04 – Crear rol con valores por defecto** | |
| **Funcionalidad disponible:** Aplicación WEB / Administración | |
| **Usuario:** Administrador | |
| **Nombre historia:** Crear rol con estado activo | **Prioridad de Negocio:** Media |
| **Puntos de Estimación: 2** | **Spring #: 2** |
| **Programador Asignado: Luis Santiago Forero Heredia** | |
| **Descripción:**  **Como administrador quiero crear roles con estado activo por defecto para poder asignar funciones sin necesidad de activarlos manualmente.** | |
| **Criterios de Aceptación:**   * **Al crear un rol, el campo status debe ser 1 automáticamente.** * **Se debe almacenar el nombre y la descripción del rol.**  |  |  | | --- | --- | | **Dado** | que el administrador crea un nuevo rol con nombre y descripción | | **Cuando** | se registra el rol en el sistema | | **Entonces** | el rol debe crearse automáticamente con status = 1 (activo) | | |
| **Observaciones:**  **Evita roles inactivos desde su creación y asegura que estén listos para usarse.** | |

|  |  |
| --- | --- |
| **HU05 – Registrar usuario nuevo con contraseña hash** | |
| **Funcionalidad disponible:** Aplicación WEB / Servicio de autenticación | |
| **Usuario:** Nuevo usuario | |
| **Nombre historia:** Registro de usuario seguro | **Prioridad de Negocio:** Alta |
| **Puntos de Estimación: 5** | **Spring #: 2** |
| **Programador Asignado: Luis Santiago Forero Heredia** | |
| **Descripción:**  **Como nuevo usuario quiero registrarme con mi correo, nombre, contraseña y rol para poder ingresar a la aplicación de forma segura.** | |
| **Criterios de Aceptación:**   * **La contraseña debe almacenarse encriptada (hash).** * **No debe permitirse registrar usuarios con correos duplicados.** * **El sistema debe retornar confirmación del registro exitoso.**  |  |  | | --- | --- | | **Dado** | que un usuario ingresa sus datos (username, email, contraseña y rol) | | **Cuando** | hace clic en “Registrar” | | **Entonces** | el sistema debe guardar la contraseña encriptada y validar que el correo no esté duplicado | | |
| **Observaciones:**  La seguridad de contraseñas es fundamental para proteger cuentas de usuario. | |

|  |  |
| --- | --- |
| **HU06 – Crear categoría** | |
| **Funcionalidad disponible:** **Aplicación WEB / Administración** | |
| **Usuario:** Administrador | |
| **Nombre historia:** Crear categoría activa | **Prioridad de Negocio:** Alta / Media / Baja |
| **Puntos de Estimación: 2** | **Spring #: 2** |
| **Programador Asignado: Luis Santiago Forero Heredia** | |
| **Descripción:**  **Como administrador quiero crear categorías para clasificar denuncias para poder organizar los casos por temas.** | |
| **Criterios de Aceptación:**   * **Las categorías nuevas deben crearse con estado activo (status = 1).** * **Se deben almacenar nombre y descripción de la categoría.**  |  |  | | --- | --- | | **Dado** | que el administrador crea una nueva categoría con nombre y descripción | | **Cuando** | hace clic en “Registrar categoría” | | **Entonces** | la categoría debe crearse con status = 1 (activo) | | |
| **Observaciones:**  Las categorías ayudan a organizar denuncias y deben estar activas al crearse. | |

|  |  |
| --- | --- |
| **HU07 – Obtener usuario por email con rol** | |
| **Funcionalidad disponible:** Aplicación WEB / Backend | |
| **Usuario:** Sistema de autenticación | |
| **Nombre historia:** Obtener usuario con rol | **Prioridad de Negocio:** Alta |
| **Puntos de Estimación: 3** | **Spring #: 3** |
| **Programador Asignado: Emerson Messi Vega Bustos** | |
| **Descripción:**  **Como sistema de autenticación quiero consultar un usuario por email y obtener también su rol para poder asignar permisos correctamente.** | |
| **Criterios de Aceptación:**   * **La búsqueda por email debe retornar los datos del usuario.** * **El resultado debe incluir el nombre del rol asociado.**  |  |  | | --- | --- | | **Dado** | que el sistema busca un usuario por su correo electrónico | | **Cuando** | el usuario existe | | **Entonces** | debe devolver un objeto con sus datos y el nombre del rol asociado | | |
| **Observaciones:**  **El rol es indispensable para gestionar permisos y accesos en la aplicación.** | |

|  |  |
| --- | --- |
| **HU08 – Validar actualización de logs** | |
| **Funcionalidad disponible:** Aplicación WEB / Backend de logs | |
| **Usuario:** Administrador | |
| **Nombre historia:** Actualizar logs | **Prioridad de Negocio:** Media |
| **Puntos de Estimación: 3** | **Spring #: 3** |
| **Programador Asignado: Emerson Messi Vega Bustos** | |
| **Descripción:**  **Como administrador quiero actualizar los logs existentes para poder mantener la trazabilidad con la información correcta.** | |
| **Criterios de Aceptación:**   * **Si no se envían datos en la actualización, debe retornar error 400.** * **Si el log no existe, debe retornar error 404.** * **No debe permitirse modificar el campo fecha.** * **Si es válido, debe retornar el log actualizado.**  |  |  | | --- | --- | | **Dado** | que un administrador intenta actualizar un log | | **Cuando** | envía un cambio válido (ejemplo: acción) | | **Entonces** | el log se actualiza correctamente | | |
| **Observaciones:**  No todos los campos deben poder actualizarse, la fecha está protegida. | |

|  |  |
| --- | --- |
| **HU09 – Transiciones de votos en denuncias** | |
| **Funcionalidad disponible:** Aplicación WEB / Módulo de denuncias | |
| **Usuario:** Ciudadano | |
| **Nombre historia:** Votar denuncias | **Prioridad de Negocio:** Media |
| **Puntos de Estimación: 5** | **Spring #: 3** |
| **Programador Asignado: Emerson Messi Vega Bustos** | |
| **Descripción:**  **Como ciudadano quiero votar a favor o en contra de una denuncia para poder apoyar o rechazar su relevancia en la comunidad.** | |
| **Criterios de Aceptación:**   * **El usuario puede votar +1 (upvote), -1 (downvote) o quitar el voto.** * **Los conteos de votos positivos y negativos deben actualizarse correctamente.** * **Votar dos veces lo mismo no debe cambiar el resultado (idempotencia).**  |  |  | | --- | --- | | **Dado** | que un ciudadano visualiza una denuncia | | **Cuando** | vota a favor (+1) o en contra (–1) | | **Entonces** | el sistema actualiza correctamente el conteo de votos | | |
| **Observaciones:**  El control de votos garantiza la relevancia justa de las denuncias. | |

|  |  |
| --- | --- |
| **HU10 – Health Check endpoint** | |
| **Funcionalidad disponible:** Microservicio de monitoreo | |
| **Usuario:** Administrador del sistema / Infraestructura | |
| **Nombre historia:** Verificar estado del sistema | **Prioridad de Negocio:** Alta |
| **Puntos de Estimación: 2** | **Spring #: 4** |
| **Programador Asignado: Emerson Messi Vega Bustos** | |
| **Descripción:**  **Como administrador del sistema quiero verificar el estado del servicio mediante el endpoint /health para poder garantizar que la aplicación y la base de datos estén operativas.** | |
| **Criterios de Aceptación:**   * **El endpoint debe responder con status: "ok" y código 200.** * **Si incluye la base de datos, debe indicar "connected" o "disconnected".**  |  |  | | --- | --- | | **Dado** | que un administrador consulta el endpoint /health | | **Cuando** | la aplicación está corriendo | | **Entonces** | debe responder con código 200 y body { status: "ok" } | | |
| **Observaciones:**  El health check permite monitorear la disponibilidad del sistema en producción. | |

Sprint 1 (Base de denuncias y logs iniciales) **Se implementan las funcionalidades mínimas para registrar denuncias y auditoría.**

* **HU01: Crear denuncia válida**
* **HU02: Registrar log automáticamente**

Sprint 2 (Gestión de usuarios, roles y categorías) **Se enfoca en asegurar la administración de usuarios y clasificación de denuncias.**

* **HU03: Crear notificación**
* **HU04: Crear rol con valores por defecto**
* **HU05: Registrar usuario nuevo con contraseña hash**
* **HU06: Crear categoría**

Sprint 3 (Consultas, validaciones y votos) **Se agregan funciones avanzadas para interacción con denuncias y manejo de errores.**

* **HU07: Obtener usuario por email con rol**
* **HU08: Validar actualización de logs**
* **HU09: Transiciones de votos en denuncias**

Sprint 4 (Monitoreo y estabilidad del sistema) **Se incluye la capacidad de supervisar la salud del sistema en producción.**

* **HU10: Health Check endpoint**

Puntos de estimación:

 1**-2 puntos:** tareas simples (crear un rol con status por defecto).

 3 **puntos:** tareas medianas (crear notificación, obtener usuario con rol).

 5 **puntos:** tareas complejas (registro de usuarios con seguridad, transiciones de votos).