

# Metodología de Desarrollo para el SIM

<b>Metodología de desarrollo para el SIM</b>	<b>1</b>
Roles	1
El tablero del sprint	1
Backlog	2
Reuniones	2
Ejecución de las historias de usuario	3

Con el objetivo de garantizar el desarrollo de las herramientas del Sistema de Información Misional de la Comisión se implementa la metodología Scrum como parte del desarrollo de software(Ver <https://proyectosagiles.org/que-es-scrum/>), adaptandola a las necesidades de la Comisión. Cabe señalar que la implementación es facilitada por la herramienta Jira(<https://www.atlassian.com/es/software/jira>). En este sentido se describe a continuación la forma en que se adapta para el SIM.

## Roles

**Dueño del producto:** Encargado de planear las funcionalidades, convertirlas en historias de usuario y planearlas en conjunto con el equipo de desarrollo. Este rol lo desempeñó en su momento el Coordinador del SIM y actualmente la Dirección de Conocimiento y el Coordinador del Proyecto Archivo del Esclarecimiento.

**Scrum Master:** Facilitador del trabajo de los desarrolladores. Este rol lo desempeña el líder del equipo de Desarrollo.

**Desarrollador:** Persona que ejecuta las actividades necesarias para completar el desarrollo solicitado en las reuniones de planeación. Actualmente, el equipo de Desarrollo del SIM cuenta con 18 personas. En este equipo de desarrollo, se involucran las personas de Infraestructura. Para el año 2022, se incluye el equipo de visualización.

**Analista de pruebas:** Encargado de validar la calidad de los desarrollos entregados. Actualmente, el equipo de desarrollo del SIM cuenta con 3 analistas de pruebas.

## El tablero del sprint

El tablero permite avanzar las tareas asignadas durante el sprint y permiten conocer el estado de las asignaciones, sus estados son:

- Por Hacer: Estado inicial desde el backlog, llegan a este punto durante la planeación.
- En Curso: Indican que el desarrollador está trabajando la funcionalidad.
- QA: Indica que el desarrollador concluyó la tarea y que está lista para ser validada.
- Reabierta: Indica que la tarea no cumplió en gran medida con lo requerido.
- Validada: Indica que la tarea ya fue validada y está lista para ser desplegada.

## Sprint 50

3 days remaining Complete sprint

Incidenias Bloqueantes Historias de Usuario Bugs Only My Issues Insights

POR HACER 10	EN CURSO 11	REABIERTA 3	QA 6	VALIDADA 3
<p>▼ Alejandro Ramírez Marín 3 issues</p> <div> <p>Automatización de pruebas del formulario de catalogación de recursos: Opciones sobre recursos</p> <p>Automatización de pruebas</p> <p>04/Mar/22 10:26 AM</p> <p>MCDLV-874</p> </div>	<div> <p>Automatización de pruebas del formulario de catalogación de recursos: Árbol de fondos, listado de recursos y filtros</p> <p>Automatización de pruebas</p> <p>04/Mar/22 10:21 AM</p> <p>MCDLV-871</p> </div>		<div> <p>Asociar rol Tesoro a la funcionalidad del Gestor de Listas</p> <p>Today 7:34 AM</p> <p>MCDLV-873</p> </div>	
<p>▼ Andres Santacruz 3 issues</p> <div> <p>Crear rol de Revisión bibliográfica</p> <p>None</p> <p>MCDLV-868</p> </div> <div> <p>Generar versiones de formato de soporte para audio, video, imagen y documento en el formulario FAI</p> <p>None</p> <p>MCDLV-840</p> </div> <div> <p>Generar slug e ident simplificado al crear un recurso en el Módulo de Catalogación</p> <p>None</p> <p>MCDLV-861</p> </div>				
<p>▼ Fabián León 2 issues</p> <div> <p>Asegurar api de servicios de administración</p> <p>None</p> <p>MCDLV-870</p> </div> <div> <p>Crear caché para el primer nivel del árbol de fondos</p> <p>None</p> <p>MCDLV-869</p> </div>				

Por otra parte, está el tablero de Soporte, en el cual se registran los bugs (errores) identificados sobre cada herramienta:

## Kanban board

Release

Only My Issues Recently Updated

TO DO 24	IN PROGRESS 14	QA 1	DONE 2
<p>▼ Alejandro Ramírez Marín 1 issue</p>			<div> <p>Error al eliminar una opción de lista</p> <p>EEAP-193</p> </div>
<p>▼ fabian.rios 2 issues</p> <div> <p>Error en el funcionamiento de la miga de pan</p> <p>491,662</p> <p>EEAP-175</p> </div>			<div> <p>Error al visualizar webcomponents</p> <p>EEAP-163</p> </div>
<p>▼ jamith.bolanos 10 issues</p> <div> <p>No se visualizan todas las posibilidades para el campo destino - Pantalla homologación</p> <p>EEAP-161</p> </div> <div> <p>No se muestran las visualizaciones de exploración de entrevistas</p> <p>EEAP-165</p> </div> <div> <p>No se visualiza resultado al dejar la opción de todos en la pestaña de visualización de metadatos</p> </div>	<div> <p>Ajustar búsqueda por palabra exacta</p> <p>EEAP-86</p> </div> <div> <p>Ajustar términos en la pantalla de exploración por Tesoro</p> <p>EEAP-103</p> </div> <div> <p>No se obtiene como primer resultado la búsqueda por ident</p> <p>EEAP-112</p> </div>		

## Backlog

Contiene el listado completo de funcionalidades proyectadas a desarrollar en la forma de historias de usuario. Está basado en las conversaciones que el “Dueño del producto” tiene con las áreas usuarias y busca atender sus necesidades en cuanto al producto. Este es el punto principal del Dueño del Producto.

## Backlog

Share ...

Incidenias Bloqueantes
Historias de Usuario
Bugs
Only My Issues

Insights

Backlog	25 issues	Create sprint	...
<input checked="" type="checkbox"/>	Habilitar el campo autorización de derechos de uso como base64 a nivel de recursos	MCDLV-855	-
<input checked="" type="checkbox"/>	Agregar el usuario al campo history del reporte de metadatos	MCDLV-856	-
<input checked="" type="checkbox"/>	Incluir listas configuradas desde el gestor en la pantalla de homologación de términos	MCDLV-857	-
<input checked="" type="checkbox"/>	Permitir modificar la cobertura geográfica por medio de la actualización masiva de recursos	MCDLV-844	-
<input checked="" type="checkbox"/>	Definir tipo de dato Int68 para el nivel de acceso a nivel de resources	MCDLV-836	-
<input checked="" type="checkbox"/>	Validar mongoSchema de resources con respecto al formulario de cargas	MCDLV-834	-
<input checked="" type="checkbox"/>	Ajustar filtro enfoque de acuerdo con la configuración de listas	MCDLV-832	-
<input checked="" type="checkbox"/>	Ajustar filtros de búsqueda avanzada de acuerdo con la configuración de listas	MCDLV-831	-
<input checked="" type="checkbox"/>	Adicionar enlace para redireccionar al documento duplicado	MCDLV-839	-
<input checked="" type="checkbox"/>	Validar las extensiones no soportadas por el buscador	MCDLV-837	-
<input checked="" type="checkbox"/>	En la vista de NGram mostrar por defecto "Todos" en el filtro de fondos	MCDLV-835	-
<input checked="" type="checkbox"/>	Permitir crear un enlace como recurso	MCDLV-818	-
<input checked="" type="checkbox"/>	En botones OR y AND Mostrar mensaje de ayuda cuando no se ha escrito una búsqueda	MCDLV-648	3
<input checked="" type="checkbox"/>	Como gestor catalogador debo poder definir los nombres de un campo	MCDLV-724	13
<input checked="" type="checkbox"/>	Como gestor catalogador debo poder definir a partir de las listas de catalogación el campo y los valores destino en la interfaz de estandarización de metadatos	MCDLV-723	8
<input checked="" type="checkbox"/>	Ajustes viz: ArcGis y Tableau	MCDLV-716	5
<input checked="" type="checkbox"/>	Como gestor catalogador quiero asociar una lista a un campo de recurso	MCDLV-721	-
<input checked="" type="checkbox"/>	Ajustar formularios de catalogación para obtener nombre de los campos de forma dinámica	MCDLV-725	-
<input checked="" type="checkbox"/>	Permitir pasar la IP del usuario a través del proxy reverse	MCDLV-711	-
<input checked="" type="checkbox"/>	Manejo jerarquico de filtros	MCDLV-719	-

A cada una de las actividades se le asigna una prioridad, se estima la complejidad con técnicas como *Planning Poker* y de acuerdo a ello se planea el *Sprint*.

## Reuniones

Scrum implementa *Sprint* periódicos en los que se definen las tareas que los desarrolladores ejecutarán durante un periodo determinado, buscando que se avance de acuerdo a estos períodos. Normalmente se definen sprint de 8 días o 15 días dependiendo de la necesidad.

Se implementan tres tipos de reuniones periódicas:

### Planeación

Se estiman las historias de usuario a construir y se enmarcan en los periodos de tiempo en que deben ser desarrollados. Cada desarrollador recibe un conjunto de historias de usuario que corresponden a puntos de historia respectivos. Estas historias quedan asignadas a los desarrolladores y es su responsabilidad que sean entregadas en los tiempos estimados.

En esta reunión se da inicio al sprint en Jira indicando el tiempo que durará y especificando la siguiente reunión de planeación.

Esta reunión se realiza de manera semanal.

### Daily

En estas reuniones se verifica el avance de lo proyectado, para esto se verifica el tablero y los estados de las historias de usuarios o bugs. Estas reuniones están agendadas en el calendario interno de la Comisión. El marco de referencia *Scrum* recomienda hacer estas reuniones de manera diaria con una duración de máximo 5 minutos, sin embargo, por los

tiempos de la Comisión, estas reuniones se practican 2 veces por semana con una duración de 20 minutos máximo.

### Temas puntuales

Son reuniones que se agendan con algunas personas del equipo para validar requerimientos que soliciten las diferentes áreas de la Comisión o cuando ocurren situaciones técnicas que se deban solucionar.

## Procedimiento de desarrollo

El procedimiento que formaliza el trabajo del equipo de desarrollo se construye en conjunto con el equipo de monitoreo del SIM y el líder del equipo, presentándolo al área de Planeación de la COMISIÓN. El documento definitivo se puede consultar en el siguiente enlace:

📄 P8.GC Desarrollo de Software V1.pdf

Adicionalmente, se definen 2 formatos de acuerdo con la complejidad del requerimiento solicitado al equipo de desarrollo:

- Requerimiento sobre herramientas existentes: Corresponde a mejoras a funcionalidades (historias de usuario). Formato: [📄 F1.P8.GC Alcance Especificación Requerimientos Funcionalidades Software V...](#)
- Requerimientos complejos que implican nueva herramienta o procedimiento: Formato [📄 F2.P8.GC Documento Requerimientos Herramientas o Software SIM ...](#)

Por otra parte, los usuarios quieren requieran solicitar un requerimiento o reportar un error, lo pueden realizar por medio de la herramienta Ayuda SIM (<https://ayudasim.comisiondelaverdad.co/scp/>), en la cual se crean tickets que son gestionados por la profesional en Soporte del equipo de desarrollo. El manual de uso de esta aplicación se puede consultar en el siguiente enlace:

📄 MANUAL DE APLICATIVO DE REQUERIMIENTOS DE DESARROLLOS SIM y CONT...

## Herramientas

- ☐ Basado en código fuente y herramientas libres

## METODOLOGÍA

Una vez contextualizado el marco de referencia del equipo de desarrollo, la ejecución se realiza de la siguiente manera:

1. Definición de células dentro del equipo de desarrollo: Considerando que la operación del equipo es transversal a varias áreas de la Comisión, existen varios frentes de trabajo, lo cual implica en distribuir el trabajo en varios sub equipos especializados, llamados células. Éstas son:

- Soporte Metabuscador: 3 desarrolladores + 1 analista de calidad.
- Soporte Archivo de Esclarecimiento: 3 diseñadores + 2 desarrolladores de visualizaciones + 3 desarrolladores + 1 analista de calidad.
- Soporte CKAN: 1 desarrollador + 1 analista de calidad.
- Integración de datos: 4 desarrolladores + 1 analista de calidad.
- Preparación de herramientas para el Legado: 3 desarrolladores + 1 analista de calidad.
- Modelos *machine learning*: 1 desarrollador + 1 analista de calidad.
- Infraestructura: 2 desarrolladores.

#### Notas:

- Con cada célula se realizan las ceremonias (reuniones) descritas anteriormente.
  - Algunos desarrolladores están involucrados en 1 o 2 células dependiendo la prioridad de los requerimientos asociados.
  - Cada célula tiene su propio tablero en Jira para gestionar los requerimientos solicitados por las áreas.
2. El dueño de producto define la prioridad de los requerimientos, la cual se registra en el *backlog* de cada célula.
  3. Semanalmente, el Líder de desarrollo realiza la planeación con cada célula creando *sprints* de 8 días.
  4. Dentro de cada *sprint* los desarrolladores y analistas de calidad actualizan en los tableros de Jira el estado de sus tareas:

4.1 Cuando el desarrollador inicia la actividad, modifica el estado a “En Curso”. Allí, se realiza la implementación de código necesario para atender el requerimiento. Si se presentan inquietudes, se socializa en el *daily* o de ser necesario se escala la inquietud al usuario.

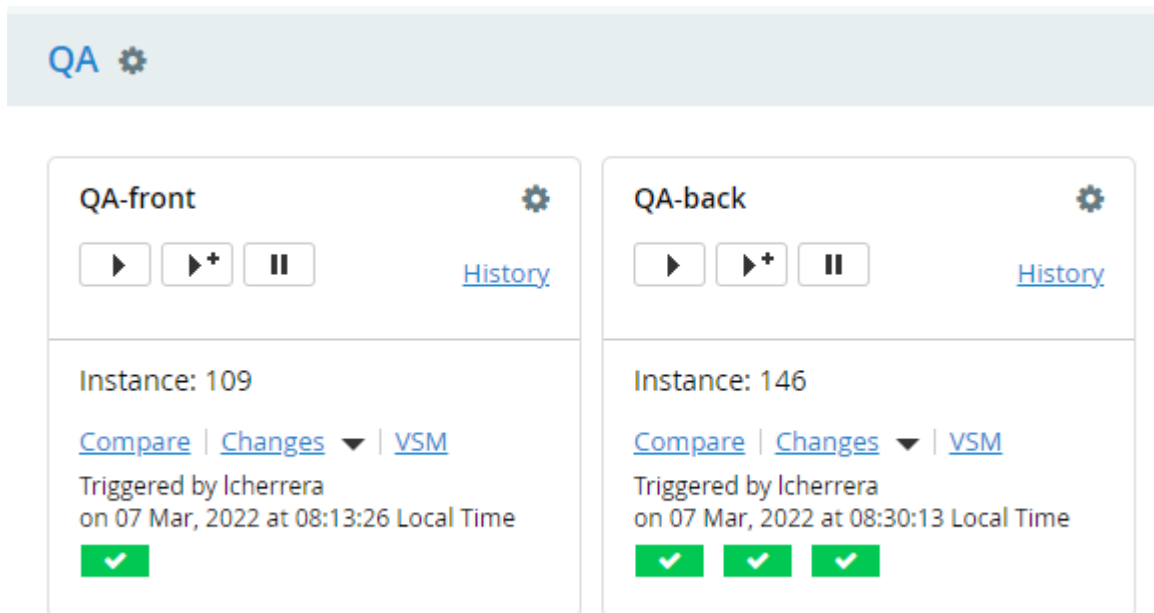
\*Antes de modificar el código (si aplica), el desarrollador debe tener configurado el ambiente de desarrollo en su equipo local y descargar los fuentes del repositorio en GitLab

\**GitLab* es el repositorio de código seleccionado para almacenar y gestionar las versiones de código de todas las herramientas, procedimientos e integraciones del SIM.

4.2 Al finalizar la modificación, el desarrollador sube el código al repositorio en GitLab correspondiente (*commit*). Posteriormente, genera una solicitud de integración a la rama de desarrollo (*develop*) creando un *merge request* en Git, el cual es aprobado por una analista de calidad o líder del equipo. Finalmente, actualiza el estado de la tarea en Jira a “QA”.

4.3 Una vez aprobado el *merge request* del desarrollador, el analista de calidad recibe la tarea y despliega la rama *develop* en el ambiente de pruebas de la

herramienta. Este despliegue se realiza por medio de la herramienta [GoCD](#), la cual automatiza este proceso.



\* Los despliegues en pruebas los realizan solamente los analistas de calidad

4.4 Al finalizar el despliegue en el ambiente de pruebas, el analista de calidad realiza los escenarios de prueba y los documenta en el formato de Casos de Prueba. Allí identifica posibles escenarios de borde. En esta validación pueden ocurrir dos resultados:

- La implementación del desarrollador es correcta y se da cumplimiento al requerimiento: En este caso, se actualiza en Jira el estado de la tarea a “Done” y se confirma su despliegue a producción.
- La implementación del desarrollador no satisface completamente el requerimiento: En este caso, se actualiza en Jira el estado de la tarea a “Reabierto”, se notifica al desarrollador y éste realiza los ajustes pertinentes. Al finalizar la corrección, se retoma al punto 4.2 de este flujo.

5. En los *daily* se revisa el estado actual de la ejecución de la planeación y con base a ello se define la fecha de despliegue a producción. Generalmente se aplica cada 15 días, de la siguiente manera:

5.1 Se valida con el equipo de analistas de calidad que los cambios son satisfactorios y tenemos una versión estable y segura para aplicar en producción.

5.2 Se agenda con anterioridad para avisar a las áreas de la Comisión la no disponibilidad de la o las herramientas. Los cambios en producción se aplican después de las 6 de la tarde en una ventana de mantenimiento de 2 horas ó los fines de semana de acuerdo con la complejidad del cambio.

5.5 Se realiza la integración de cambios de la rama de desarrollo (*develop*) a la rama de producción (*master*) en GitLab.

5.4 Usando la herramienta [GoCD](#) se despliega la rama máster del código al ambiente de producción. Este despliegue solamente lo realizan los ingenieros de soporte o líder del equipo.

5.5 Se documenta el control de cambios en la herramienta Ayuda Sim.

5.6 Se notifica a los usuarios sobre los cambios realizados en las herramientas.

## Infraestructura requerida

## Stack tecnológico

## Línea de tiempo del equipo de desarrollo

 Línea del tiempo