

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>1 de 21</b>

## **METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE**

**SEPTIEMBRE DE 2025**

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>2 de 21</b>

## **CONTROL DE CAMBIOS**

<b>VERSIÓN</b>	<b>SECCIÓN</b>	<b>TIPO</b>	<b>FECHA (DD/MM/AAAA)</b>	<b>AUTOR</b>	<b>OBSERVACIONES</b>
1.0	Todas	Creación	30/09/2025	GIT de Apoyo Informático	Se implementa la metodología basada en scrum.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>3 de 21</b>

## CONTENIDO

<b>1.</b>	<b>Contexto .....</b>	4
<b>2.</b>	<b>Glosario .....</b>	5
<b>3.</b>	<b>Justificación .....</b>	5
<b>3.1</b>	<b>Alcance .....</b>	7
<b>3.2</b>	<b>Audiencia .....</b>	7
<b>4.</b>	<b>SCRUM (Sutherland, 2020) .....</b>	7
<b>4.1</b>	<b>Scrum en CGN .....</b>	8
<b>4.1.1</b>	<b>Roles .....</b>	8
<b>4.1.2</b>	<b>Eventos .....</b>	9
<b>4.1.2.1</b>	<b>Planificación inicial (Sprint 0) .....</b>	9
<b>4.1.2.1.1</b>	<b>Contenidos .....</b>	10
<b>4.1.2.2</b>	<b>Iteraciones (Sprint) .....</b>	10
<b>4.1.2.2.1</b>	<b>Planeación de la iteración (Sprint Planning) .....</b>	11
<b>4.1.2.2.2</b>	<b>Implementación .....</b>	11
<b>4.1.2.2.2.1</b>	<b>Daily Meeting .....</b>	12
<b>4.1.2.2.3</b>	<b>Revisión de la iteración (Sprint Review) .....</b>	12
<b>4.1.2.2.4</b>	<b>Retrospectiva de la iteración (Sprint Retrospective) .....</b>	13
<b>4.1.3</b>	<b>Artefactos .....</b>	13
<b>4.1.3.1</b>	<b>Reserva de requerimientos del producto (Product Backlog) .....</b>	13
<b>4.1.3.2</b>	<b>Reserva de requerimientos de la iteración (Sprint Backlog) .....</b>	14
<b>4.1.3.3</b>	<b>Incremento (Increment) .....</b>	14
<b>4.1.3.4</b>	<b>Artefactos CGN .....</b>	14
<b>5.</b>	<b>Ciclo de vida de desarrollo de software - SDLC .....</b>	16
<b>5.1</b>	<b>Fases del SDLC .....</b>	17
<b>5.1.1</b>	<b>Planificación y Análisis .....</b>	17
<b>5.1.2</b>	<b>Diseño .....</b>	17
<b>5.1.3</b>	<b>Desarrollo (Implementación) .....</b>	18
<b>5.1.4</b>	<b>Pruebas .....</b>	19
<b>5.1.5</b>	<b>Instalación o despliegue .....</b>	19
<b>5.1.6</b>	<b>Uso y mantenimiento .....</b>	19
<b>6.</b>	<b>Consideraciones generales .....</b>	20
<b>7.</b>	<b>Referencias .....</b>	21

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>4 de 21</b>

## 1. Contexto

En nuestro país, los lineamientos en términos de transformación digital se definieron a través de la Política de Gobierno Digital, la cual ha sido actualizada y regulada en el Decreto 767 de 2022 por el Ministerio de Tecnologías de la Información y Comunicaciones (MinTIC), como un instrumento para las entidades públicas en todos sus niveles (municipal, departamental y nacional). Esta política busca que las entidades generen valor público al implementar tecnologías digitales que permitan mejorar la eficiencia, transparencia y participación de sus procesos promoviendo una gestión moderna y efectiva.

La U. A. E. Contaduría General de la Nación (UAE CGN) es el organismo técnico de regulación de la contabilidad para el sector público en el país, que en desarrollo del mandato constitucional y legal es responsable de:

- Determinar las políticas, principios y normas de contabilidad que deben regir en el país para el sector público.
- Uniformar, centralizar y consolidar la información contable.
- Elaborar el Balance General Consolidado de la Nación.

Bajo el contexto normativo mencionado, y como se pudo evidenciar previamente, la misionalidad de la entidad no está orientada hacia el desarrollo de software, aun así, se deben garantizar la disponibilidad, la integridad y la confidencialidad de su información; los cuáles son sólo algunos de los atributos de calidad del software. Las herramientas tecnológicas de la UAE CGN, ya sean adquiridas, adaptadas o desarrolladas (parcial o totalmente), son críticas para el cumplimiento de las funciones asociadas a sus procesos misionales, estratégicos, de apoyo y de evaluación. La Política de Gobierno Digital establece como una responsabilidad institucional el coordinar, adoptar, implementar y hacer seguimiento y verificación de su implementación.

En la práctica, para las entidades es difícil responder de manera estructurada a los requerimientos de la evolución tecnológica. En particular, si no se cuenta con una metodología clara, y con soluciones proactivas, articuladas y trazables. Esto puede resultar en sobrecarga para los equipos técnicos, dependencia del conocimiento tácito y dificultades para mantener la sostenibilidad de las soluciones a mediano plazo. Por tanto, resulta necesario implementar una metodología de trabajo ágil, planificada y orientada a la solución de las necesidades tecnológicas de la entidad. Esta metodología debe estar alineada con la Política de Gobierno Digital —especialmente con los principios de articulación, competitividad, innovación, prospectiva y resiliencia tecnológicas— y adaptarse a las particularidades de una entidad pública no dedicada al

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>5 de 21</b>

desarrollo de software, pero sí altamente dependiente de herramientas digitales para su operación y evolución.

La UAE CGN adopta la metodología ágil *SCRUM* para ser utilizada en los proyectos de desarrollo de software, la cual se centra en la colaboración, la iteración y la adaptación continua, así como en la entrega de valor de forma incremental y a través de ciclos iterativos llamados *sprints*.

## 2. Glosario

- **Administrador de Aplicaciones:** Persona encargada de la gestión y mantenimiento de un sistema o aplicación específica, incluyendo instalación, configuración, actualización y resolución de problemas del aplicativo.
- **Administrador de Requerimientos:** Responsable de la gestión de los requerimientos de un proyecto, cuyas funciones incluyen la recopilación, documentación, análisis, validación y gestión de cambios de los requerimientos de los usuarios.
- **Artefactos:** Son los formatos o instrumentos específicos utilizados por la UAE CGN para la documentación en el proceso de desarrollo de software.
- **Definición de Hecho (*DoD – Definition of Done*):** Descripción formal de las condiciones que deben cumplirse para que un producto, historia de usuario o tarea se considere "hecha", implicando que el código esté funcional y listo para ser entregado.
- **Épica:** Característica o funcionalidad de alto nivel que aporta valor significativo a los usuarios finales. Se define como un conjunto grande de trabajo que se desglosa en tareas más pequeñas llamadas historias de usuario. Son fundamentales para la planificación y gestión de proyectos ágiles.
- **Mantenibilidad:** Se refiere a la facilidad con la que un sistema o aplicación puede ser mantenido y actualizado. Un sistema mantible es fácil de entender, modificar y solucionar problemas.
- **Repositorio:** Ubicación donde se almacena el código fuente del software.
- **TDD (Test Driven Development):** Método de Desarrollo ágil donde los desarrolladores escriben pruebas automatizadas para una característica deseada antes de escribir el código.

## 3. Justificación

La UAE CGN requiere generar una metodología de trabajo para desarrollar y mantener el software y los sistemas de información misionales, de apoyo, estratégicos y de evaluación.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>6 de 21</b>

El CHIP —como la mayoría de los sistemas de información existentes en la UAE CGN— se encuentra en constante mantenimiento, actualización y/o soporte, esto quiere decir que está en una mejora continua de sus funcionalidades y tecnologías, así como en el seguimiento a su disponibilidad e integridad. Además, no es raro que en ocasiones surjan necesidades que involucran el desarrollo de características o aplicativos nuevos para los procesos estratégicos, de apoyo y de evaluación.

En el contexto de la entidad, se presentan dificultades particulares para el mantenimiento de software bajo desarrollo con metodologías ágiles. Por ejemplo, a medida que las herramientas tecnológicas aumentan en tamaño y complejidad, se incrementa la dificultad para realizar pruebas que garanticen su correcto funcionamiento por varios factores como la cantidad de personas que pueden dedicarse exclusivamente a esta tarea, la falta de implementación de herramientas automatizadas enfocadas a pruebas y la multiplicidad de responsabilidades que se deben asumir sobre los distintos desarrollos. Si, además agregamos que cuando el software tiene un tiempo de vida prolongado es común que acumule múltiples modelos de arquitectura, estilos de desarrollo e incluso enfoques de negocio, crece significativamente la complejidad de su mantenimiento. Dicha diversidad de factores sobre las herramientas no solo dificulta la implementación de nuevas funcionalidades, sino que son criterios a tener en cuenta al momento de identificar los tiempos y costos asociados a cualquier tipo de cambio o evolución.

Por estos motivos, se plantea la presente metodología, la cual busca subsanar estas brechas mediante un enfoque estructurado que permita planificar, clasificar y atender los requerimientos con criterios objetivos, predecibles y medibles. Se inspira en prácticas de gestión ágil y mantenimiento de software, adaptadas al contexto de una entidad pública cuyo enfoque principal no es el desarrollo de software y en donde los recursos humanos para atender las herramientas tecnológicas pueden ser limitados.

Además, se espera implementar buenas prácticas que permitan fortalecer la gobernanza digital de la entidad, mejorar la capacidad de respuesta ante nuevas necesidades tecnológicas y garantizar que los desarrollos, adaptaciones o mantenimientos se ejecuten de manera eficiente y alineados a los objetivos institucionales.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>7 de 21</b>

### **3.1 Alcance**

En el documento se concretan las actividades a realizar por los integrantes del equipo de sistemas de información e información al implementar un nuevo proyecto. Desde la obtención de requerimientos hasta su puesta en producción, incluyendo reuniones de *SCRUM*, documentación, roles de los integrantes, iteraciones, planeaciones de *Sprint*, entre otros.

Dentro de la ejecución de la metodología, se tendrá en cuenta el ciclo de vida de desarrollo que cuenta con las fases de especificación de requerimientos, diseño, desarrollo, pruebas y despliegue. Sin embargo, no hay que confundirse con la aplicación tradicional de las fases, al contrario, lo que se busca es rescatar los puntos fuertes de estas y tenerlos en cuenta en cada uno de los desarrollos como etiquetas (tags) que permitan la clasificación de historias de usuario, fortaleciendo que el enfoque no sea únicamente al desarrollo, pero también los elementos transversales a este.

### **3.2 Audiencia**

Este documento está dirigido principalmente a los miembros del equipo de sistemas de información e información, al coordinador del GIT de Apoyo Informático y demás funcionarios de la UAE CGN interesados.

## **4. SCRUM (Sutherland, 2020)**

Scrum es un proceso en el que se aplican de forma periódica un conjunto de actividades para trabajar colaborativamente, en equipo, y alcanzar el mejor resultado posible de un proyecto. Se realizan entregas parciales del producto final, priorizadas por el beneficio que aportan al receptor del proyecto.

Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados confiables; donde los requisitos son muy variables o poco definidos; y, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

También se utiliza para resolver situaciones en que no se está entregando al usuario final lo que necesita; cuando las entregas están tomando más tiempo de lo estimado, o simplemente la calidad no es suficiente; cuando la moral del equipo es baja; y, cuando es necesario definir y solucionar ineficiencias en las tareas.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>8 de 21</b>

#### 4.1 Scrum en CGN

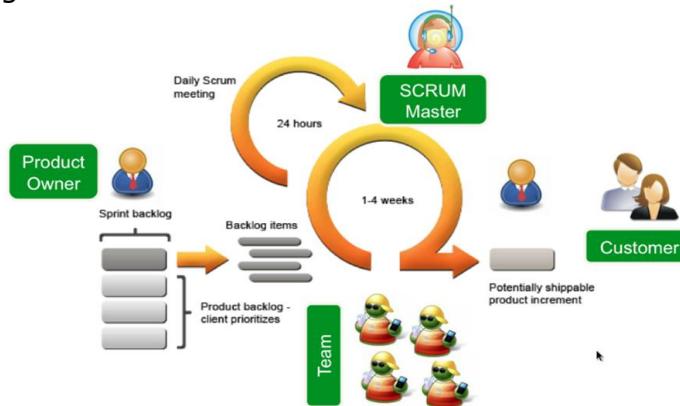
La metodología principal que se va a adoptar, para todos los desarrollos, en la CGN es **Scrum**. Esto, por las fortalezas que ofrece principalmente con respecto a entregas constantes, refactorización de requerimientos, y flexibilidad en los equipos de trabajo. Dicho esto, teniendo en cuenta los artefactos que se encuentran definidos en la entidad y la importancia de estos para una documentación robusta de los aplicativos, se tendrán en cuenta las fases del ciclo de vida del desarrollo de software - SDLC (descritas en el [numeral 4](#)), que si bien, están enfocadas a metodologías más tradicionales, no pierden valor en su implementación dentro de *Scrum*, dado que funcionan como herramientas que se deben tener en cuenta, pero no pasos consecutivos que se deben ejecutar para llevar a buen término un desarrollo.

Para la implementación de la metodología *Scrum* se debe tener en cuenta lo siguiente:

- Previo al inicio del ciclo de iteraciones (*sprints*), se debe realizar una actividad preliminar de planificación inicial en la que se recibe la solicitud, se realiza el análisis de viabilidad y se define la primera versión de la reserva de requerimientos (*Product Backlog*), compuesto por las historias de usuario identificadas.
- Se deben ejecutar los 3 eventos principales (Planeación de la iteración (*Sprint Planning*), Revisión de la iteración (*Sprint Review*) y Retrospectiva de la iteración (*Sprint Retrospective*)).

##### 4.1.1 Roles

Se definen los siguientes roles:



Fuente: <https://www.ayselucus.es/sites/default/files/noticias/fotos/860f82cb->

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>9 de 21</b>

[d4bc-4b24-ac23-a878e4e4baab.png](#)

- **Dueño del producto (Product Owner):** Es el dueño del producto o la persona responsable de la definición, actualización y validación de los requisitos. Generalmente es quien solicita los cambios o nuevos desarrollos, está definido en la *Solicitud de servicio de TI*.
- **Scrum Master:** Es el que apoya al equipo de sistemas de información e información a orientar y ejecutar cada una de las reuniones y ayudará en los problemas que hayan surgido. El *Scrum Master* guía, facilita y enseña las prácticas de *Scrum* a todos los involucrados en el proyecto.
- **Equipo (Scrum Team):** Es el grupo de personas encargadas del desarrollo del proyecto. Está encargado de analizar, diseñar y probar cada uno de los requerimientos definidos por el dueño del producto (*Product Owner*). Está integrado por los miembros del equipo de sistemas de información e información del GIT de Apoyo Informático.
- **Partes interesadas o clientes (Stakeholders):** Son todas las personas que participan directa o indirectamente en el proyecto, y para las cuales el proyecto generará un impacto. Algunos de ellos son: El Coordinador del GIT de Apoyo Informático, el usuario funcional de la aplicación, Sub Contadores de la entidad según el caso, Coordinadores de los distintos GIT de la entidad según el impacto del proyecto y el equipo de Certificación de Software.

#### 4.1.2 Eventos

##### 4.1.2.1 Planificación inicial (Sprint 0)

Tiene como objetivo preparar todos los elementos necesarios para iniciar el primer *Sprint* de forma efectiva.

Se compone de las actividades que se deben ejecutar para la validación de la solicitud *Solicitud de servicio de TI (Mesa de servicio)*, la construcción del *Product Backlog* para el inicio de las iteraciones, así como con la definición del *Product Goal*. Dichas actividades como mínimo deberán ser:

- Revisión de la viabilidad y aprobación de la solicitud.
- Alistamiento de los elementos que se requieren para la implementación de la metodología, teniendo en cuenta los contenidos descritos a continuación.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>10 de 21</b>

#### **4.1.2.1.1 Contenidos.**

- Definición de la visión del producto**

La elabora el dueño del producto (*Product Owner*), y en esta se definen los objetivos generales, el valor que aporta y el problema que resuelve el producto.

- Identificación de interesados (stakeholders)**

Determinar las personas o grupos clave que se verán impactados por el producto o que tienen interés en él.

- Formación del equipo Scrum**

Identificar las personas y realizar la asignación de roles.

- Capacitación en Scrum**

Realizar una inducción a todos los miembros del equipo y partes interesadas sobre la metodología a implementar asegurando comprensión de los principios, roles y eventos de Scrum.

- Creación inicial de la reserva de requerimientos (*Product Backlog*)**

Recopilar los requisitos iniciales: épicas, historias de usuario, tareas; y priorizarlos con base en las necesidades del negocio.

- Definición del entorno de trabajo**

Preparar las herramientas de colaboración y desarrollo.

- Definición de arquitectura básica (opcional)**

Realizar un **análisis técnico** preliminar para establecer bases tecnológicas sólidas. Seleccionar frameworks, integraciones, estándares técnicos.

- Criterios de éxito y métricas**

Establecer los KPIs o indicadores clave de desempeño y definir cómo se medirá el avance y éxito del proyecto.

#### **4.1.2.2 Iteraciones (*Sprint*)**

Las iteraciones son la base de una metodología ágil, en este caso *Scrum*. También llamados *sprints*, son el periodo de tiempo en el cual se desarrollan cierta cantidad de requerimientos del proyecto. La duración y la cantidad de las tareas son definidas por el *Scrum Master* y el equipo de sistemas de información e información (*Scrum Team*), según la experiencia de estos y la prioridad de los requerimientos.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>11 de 21</b>

En la UAE CGN está definido que el periodo de cada *sprint* es de 2 semanas, el cual puede variar según recomendaciones del *Scrum Master*. El seguimiento de cada iteración se realiza mediante una herramienta de gestión de proyectos (como trello, jira, etc.).

#### **4.1.2.2.1 Planeación de la iteración (*Sprint Planning*)**

Es donde el equipo planifica la iteración, elabora la estrategia que le permitirá conseguir el mejor resultado posible. Esta actividad la realizan las personas integrantes del equipo, cada miembro es el responsable de organizar su trabajo y es quien mejor conoce cómo realizarlo.

Durante esta reunión se debe definir un *Sprint Goal*, que no es más que el objetivo a alcanzar durante el *sprint*, es decir, el producto, desarrollo o servicio deseado. Se crea durante el *Sprint Planning* mediante la colaboración de todo el *Scrum Team*. El *Product Owner* debe proponer un objetivo de negocio a conseguir, y en conjunto con lo definido en el *Product Backlog*, son la base para la definición de esa meta.

Se definen las tareas necesarias para poder completar cada objetivo o requerimiento, creando una lista de tareas incluyendo el esfuerzo y tiempo que se espera consumir por cada una de ellas (Reserva de requerimientos del producto (*Sprint Backlog*)), las cuales serán revisadas al final de cada iteración. Cada miembro del equipo se auto asigna a las tareas que puede realizar.

Si no es el primer *sprint*, se debe agregar un tiempo previo a la selección de tareas, para el refinamiento del *Product Backlog* (ver numeral 4.1.1).

El *Sprint Planning* es el primer evento, marca el inicio de una iteración y debe tener una duración máxima de 4 horas para las 2 semanas de *Sprint* planteadas.

#### **4.1.2.2.2 Implementación**

Durante las siguientes dos semanas, se debe ir revisando y ajustando las historias de usuario asignadas al *Sprint Backlog*. Se debe tener en cuenta la clasificación de las historias de usuario, conforme con el **SDLC** y sus artefactos relacionados (ver sección 4).

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>12 de 21</b>

#### **4.1.2.2.2.1 Daily Meeting**

También llamadas reuniones diarias o periódicas; son reuniones muy cortas que realiza el *Scrum Team* liderados por el *Scrum Master* donde el objetivo es facilitar el paso de información y la colaboración entre los integrantes del equipo para exponer cuales son los inconvenientes que se le han presentado desde la última reunión en el avance de las tareas, y también para detallar las acciones a seguir con las actividades que está realizando; resumiendo, en lo posible ¿qué hizo desde ayer? ¿Qué tiene planeado hacer hoy? Y si ¿tiene algún problema que lo esté retrasando? Estas reuniones, se recomienda que no deben superar los 15 minutos.

Cada miembro del equipo de alguna forma está al tanto del trabajo que el resto del equipo está realizando para que, al finalizar la reunión se puedan hacer los cambios necesarios que permitan cumplir con el compromiso pactado para la iteración. Si se requiere, se deben plantear reuniones posteriores para la solución o aclaración de las tareas que presenten inconvenientes.

En el caso de la UAE CGN no se realizan reuniones diarias, en su lugar, deben ser 2 reuniones por semana.

#### **4.1.2.2.3 Revisión de la iteración (*Sprint Review*)**

El propósito del *Sprint Review* es inspeccionar el resultado del *Sprint* y determinar futuras adaptaciones. El *Scrum Team* presenta los resultados del trabajo realizado sobre los puntos clave para los *Stakeholders* y se discute el progreso de los objetivos planteados.

Durante el evento, el *Scrum Team* y los *Stakeholders* identifican qué se logró en el *Sprint* y qué ha cambiado en su planeación inicial. Basados en esta información, los asistentes definen qué se hará en la siguiente iteración. También se podrá ajustar el *Product Backlog* a partir de la identificación de nuevas necesidades o mejoras. El *Sprint Review* es una sesión de trabajo y el *Scrum Team* debe evitar limitarlo a una presentación.

El *Sprint Review* no debe ser para entregar incrementos (*increments*), es decir, no se realizan despliegues ni se muestran nuevas funcionalidades o características que no conozca el *Product Owner*.

El *Sprint Review* es el penúltimo evento y debe tener una duración máxima 2 horas para las 2 semanas de *Sprint* planteadas.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>13 de 21</b>

#### **4.1.2.2.4 Retrospectiva de la iteración (*Sprint Retrospective*)**

El objetivo es identificar los problemas que surgieron y encontrar maneras de gestionarlos para dar cumplimiento a las tareas planteadas durante el *Sprint Planning*.

El equipo debe evaluarse individualmente, sus interacciones, los procesos, las herramientas y su **DoD**. “Para ello, los integrantes del equipo tratan de responder a tres preguntas claves sobre el sprint que acaba de finalizar: ¿qué ha ido bien durante el sprint? ¿qué se puede mejorar? y ¿qué ha fallado?”<sup>1</sup>

El *Sprint Retrospective* es el último evento y debe tener una duración máxima de 2 horas para las 2 semanas de *Sprint* planteadas.

#### **4.1.3 Artefactos**

##### **4.1.3.1 Reserva de requerimientos del producto (Product Backlog)**

“El Product Backlog es una lista de nuevas funciones, mejoras, correcciones de errores, tareas o requisitos de trabajo necesarios para crear un producto. Se obtiene a partir de fuentes como la atención al cliente, los análisis de la competencia, las demandas del mercado y los análisis empresariales en general.”

El Product Backlog es un artefacto “vivo”, en el sentido de que se actualiza a medida que se dispone de nueva información. Se trata de un backlog compartido entre varios equipos que el propietario del producto mantiene y organiza entre los sprints y conforme van surgiendo nuevas ideas. Contiene tareas que una vez estuvieron en un sprint activo, pero que dejaron de ser prioritarias y pasaron de nuevo al backlog.”<sup>2</sup>

Para su construcción, se debe trabajar con la metodología Jira o Kanban, la cual “Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo”<sup>3</sup>. Como mínimo, se deben definir los estados de Pendiente, En curso y Hecho.

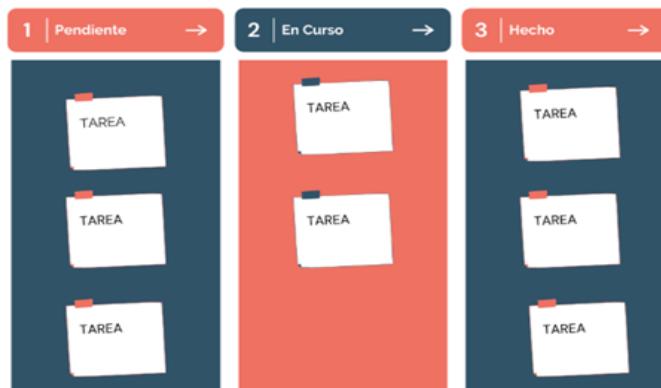
<sup>1</sup> (MacNeil, 2025)

<sup>2</sup> (Harris, s.f.)

<sup>3</sup> (Martins, 2025)

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>14 de 21</b>

## KANBAN



Fuente: <https://josefacchin.com/wp-content/uploads/2021/07/tablero-kanban.png.webp>

### 4.1.3.2 Reserva de requerimientos de la iteración (Sprint Backlog)

El *Sprint Backlog* es un conjunto de tareas del *Product Backlog* que se han planteado para desarrollarse durante el *sprint* para lograr un incremento en la entrega del producto.

Los desarrolladores deben participar activamente en la planificación de los *Sprint Backlogs* de cara a los futuros entregables del producto (*Increments*).

### 4.1.3.3 Incremento (Increment)

Es la adición o modificación de una funcionalidad, desempeño, servicio, componente o característica del producto a entregar. Se pueden definir múltiples incrementos dentro de un *sprint*, sin embargo, tener en cuenta que sean lo suficientemente consistentes con los criterios del DoD. Se debe entregar el incremento antes de la finalización del *sprint*.

### 4.1.3.4 Artefactos CGN

Los formatos aquí descritos se puede encontrar en la intranet de la CGN en la ruta: [SIGI → Documentos SGC → Gestión TICs → Formatos](#).

- GTI07-FOR07 (Guiones de prueba de seguridad):** artefacto que detalla los pasos para verificar que el sistema o aplicación es resistente

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>15 de 21</b>

frente a amenazas, vulnerabilidades y ataques, simulando comportamientos maliciosos o condiciones inseguras. Su propósito es evaluar la robustez, integridad, confidencialidad y disponibilidad del software ante posibles riesgos de seguridad.

- **GTI06-FOR01 (Matriz de Seguimiento y Control de Versión):** artefacto que permite rastrear, documentar y controlar los cambios realizados en una herramienta de software a lo largo de su ciclo de vida. Su propósito es mantener un registro claro de las versiones, los cambios implementados, sus responsables y el estado actual de cada versión, facilitando la trazabilidad, auditoría y gestión de configuraciones del proyecto.
- **GTI06-FOR02 (Acta de recibo a satisfacción):** artefacto mediante el cual un cliente o usuario responsable certifica que ha recibido un producto, servicio o entrega parcial del proyecto de software, y que este cumple con las condiciones, requerimientos y criterios de calidad previamente establecidos. Su firma representa conformidad y aceptación, y puede marcar el cierre de una fase, hito o la totalidad del proyecto.
- **GTI06-FOR03 (Acta de Certificación de Versión):** artefacto que tiene como objetivo dejar constancia técnica y administrativa de la validación y aprobación de una nueva versión de software que será liberada o entregada. Este formato permite evaluar integralmente el estado funcional y técnico de la versión, asegurando que los productos entregados cumplan con los criterios de calidad, seguridad, funcionalidad y estabilidad establecidos.
- **Documento de despliegue:** artefacto que describe de manera detallada los pasos, procedimientos, configuraciones y consideraciones necesarias para instalar, configurar, poner en funcionamiento y verificar el correcto despliegue de una aplicación o sistema de software en uno o más entornos (desarrollo, pruebas, preproducción o producción).
- **Guiones de prueba:** artefacto que detalla los casos y pasos de prueba necesarios para verificar que una historia de usuario o requisito cumpla sus criterios de aceptación y requisitos no funcionales. Sirve como base de ejecución para QA/Dev, soporta la trazabilidad con requisitos y facilita la evidencia para la aceptación DoD, auditoría y mejora continua.
- **Historia de usuario:** artefacto que describe de forma concisa una

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>16 de 21</b>

necesidad del usuario o negocio, la razón de valor y los criterios de aceptación que permiten verificarla. Se usa para entender, acotar y preparar la historia antes de su construcción, asegurando trazabilidad con requisitos, reglas de negocio y políticas (seguridad, calidad, arquitectura).

## 5. Ciclo de vida de desarrollo de software - SDLC

Si bien se podría entender el ciclo de vida del desarrollo de software (SDLC por sus siglas en inglés) como parte de una metodología tradicional para la construcción de herramientas informáticas. La verdad es que es una parte integral de cualquier desarrollo dado que estructura las fases mínimas requeridas para que una herramienta se pueda construir y empezar a utilizar. Dicho esto, el enfoque del SDLC dentro de la CGN está orientado a su inclusión dentro de la filosofía enmarcada por las metodologías ágiles, especialmente con SCRUM.

En el caso específico de la implementación del SDLC dentro de SCRUM, se debe tener en cuenta varios criterios:

1. Al ser una metodología ágil, se espera que la ejecución de las fases sea incremental, es decir, no se ejecutarán las fases completas ni se aplicarán cambios significativos en cada iteración. Al contrario, debe haber una modificación sencilla que permita ir evolucionando el producto.
2. La evaluación de requisitos debe ser constante y rápida para permitir al equipo identificar las necesidades propuestas antes de que lleguen a ser un problema grave.
3. Debe haber una participación constante de los clientes y partes interesadas, con el objetivo de refinar las entregas e ir dando victorias tempranas para su implementación. Sin embargo, no pueden ser los únicos con voz o responsabilidad sobre el proyecto, para evitar cambios sustanciales durante su implementación.
4. Debe ser **opcional**, es decir, NO es necesario ejecutar cada una de las fases en cada uno de los desarrollos planteados, esto permite agilizar el desarrollo y enfocar a los equipos en la entrega de soluciones de valor.

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>				
<b>PROCESO:</b>	GESTIÓN TIC			
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE			
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>	
30/09/2025	GTI07-MTD01	01	17 de 21	

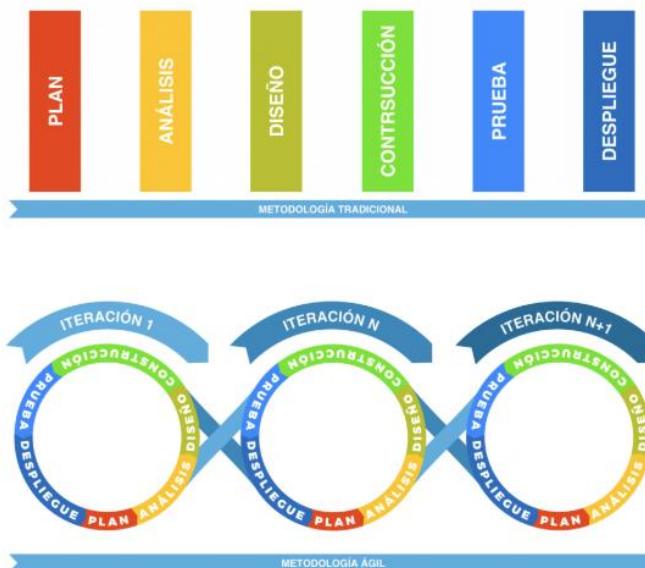


Ilustración 1. Tomado de <https://www.eniun.com/metodologia-scrum-desarrollo-software-agil/>

## 5.1 Fases del SDLC

### 5.1.1 Planificación y Análisis

Durante la **planificación del sprint** y las sesiones de **refinamiento del Product Backlog**, se analiza qué historias de usuario serán priorizadas y conforman el objetivo del sprint.

Se evita el análisis excesivo tradicional; en su lugar se mantiene un flujo continuo de refinamiento iterativo, con retroalimentación frecuente del *Product Owner* y *stakeholders* para ajustar alcance y valor de negocio.

#### Artefactos:

Solicitud de servicio de TI (Mesa de servicio) – Product Backlog – Sprint Backlog – Historias de usuario. Guiones de pruebas

### 5.1.2 Diseño

El diseño se realiza de forma **incremental y ligera**, dentro del sprint, como parte del desarrollo de cada historia. Se privilegia colaboración directa entre desarrolladores y *Product Owner*.

No existe una fase separada; el equipo refina y adapta diseños a medida que

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>18 de 21</b>

progesa, promoviendo prototipado rápido y decisiones orientadas al valor.

Es posible que la solución inicial no sea la más adecuada, por lo que en tal caso hay que refinarla.

#### **Artefactos:**

GTI07-FOR07 (Guiones de prueba de seguridad).

#### **5.1.3 Desarrollo (Implementación)**

Los desarrolladores trabajan sobre el *Sprint Backlog*, completando tareas pequeñas durante el *sprint* para alcanzar un incremento potencialmente entregable al final. Se **siguen las prácticas** de definición de listo (DoR por sus siglas en inglés) y de hecho (DoD por sus siglas en inglés).

El desarrollo es incremental y cruzado: **codificación, pruebas unitarias y documentación mínima** ocurren durante el sprint, garantizando calidad continua.

Como se plantea en *Intelequia*<sup>4</sup>: “En esta fase hay que elegir las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.”

Al programar, hay que intentar que el código no sea indescifrabla siguiendo distintas pautas como las siguientes:

- Evitar bloques de control no estructurados.
- Identificar correctamente las variables y su alcance.
- Elegir algoritmos y estructuras de datos adecuadas para el problema.
- Mantener la lógica de la aplicación lo más sencilla posible.
- Documentar y comentar adecuadamente el código de los programas.
- Facilitar la interpretación visual del código utilizando reglas de formato de código previamente consensuadas en el equipo de desarrollo.

También hay que tener en cuenta la adquisición de recursos necesarios para que el software funcione, además de desarrollar casos de prueba para comprobar el funcionamiento de este según se vaya programando.”

<sup>4</sup> (Intelequia, s.f.)

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>19 de 21</b>

#### **5.1.4 Pruebas**

Dentro de cada sprint se realiza **testing continuo**: unitario, integración, incluso pruebas automatizadas y de aceptación según el *DoD*.

Al integrar pruebas desde el inicio, se evitan errores costosos detectados tardíamente. Esto puede incluir enfoques como TDD (Test Driven Development).

**Artefactos:**

GTI07-FOR07 (Guiones de prueba de seguridad) – GTI06-FOR03 (Acta de Certificación de Versión), Guiones de pruebas

#### **5.1.5 Instalación o despliegue**

El equipo entrega el incremento al *Product Owner* o *stakeholders* para valoración durante la *Sprint Review*, y se puede considerar su despliegue si cumple los criterios de entrega.

Despliegues frecuentes y semi-automáticos o automáticos (CI/CD) permiten recibir retroalimentación real y reducir riesgos, alineándose al flujo iterativo de Scrum.

**Artefactos:**

Solicitud de Cambio de TI – GTI06-FOR01 (Matriz de Seguimiento y Control de Versión) – GTI06-FOR02 (Acta de recibo a satisfacción).

#### **5.1.6 Uso y mantenimiento**

Después del despliegue, se abordan defectos, mejoras y nuevas solicitudes mediante elementos del *Product Backlog* para futuros *sprints*. El equipo refina continuamente el backlog y evoluciona el producto.

*“Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:*

*Es un ciclo ininterrumpido: cada sprint puede incluir mantenimiento, adaptaciones o nuevas funcionalidades según prioridades y feedback.*

*Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).*

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>20 de 21</b>

*Adaptarlo a nuevas necesidades (mantenimiento adaptativo).*

*Añadirle nuevas funcionalidades (mantenimiento perfectivo).*

*Aunque suene contradictorio, cuanto mejor es el software más tiempo hay que invertir en su mantenimiento. La principal razón es que se usará más (incluso de formas que no se habían previsto) y, por ende, habrá más propuestas de mejoras.*” (Intelequia, s.f.)

En conclusión, en *Scrum*, no se abandona la lógica del SDLC clásico, sino que se transforma: las fases se ejecutan como pequeñas iteraciones contenidas en *sprints*, con entrega incremental de valor, retroalimentación (feedback) frecuente y mejora continua. Esto hace del ciclo de vida una serie de mini-ciclos completos, más ágiles y adaptativos.

## 6. Consideraciones generales

La disponibilidad del cliente debe ser alta durante todo el proyecto, dado que participa de manera continua. Es importante su participación para asegurar que el producto final va a solucionar las problemáticas que impulsaron la solicitud inicial.

Cuando los cambios son muy grandes o tienen un impacto alto, es recomendable dividirlos en varios requerimientos para **minimizar el riesgo**.

Si bien se recomienda utilizar **iteraciones cortas de entre 2 y 4 semanas**, el tamaño ideal de una iteración puede depender de:

- Condiciones específicas del proyecto.
- Necesidad de retroalimentación temprana.
- **Equilibrio entre trabajo útil y carga operativa:** Las iteraciones deben ser lo suficientemente largas para que el tiempo dedicado a reuniones y otras tareas administrativas no domine el esfuerzo total.

Tener en cuenta que, con el tiempo, el equipo aprende a estimar su **velocidad de desarrollo**, es decir, cuánto trabajo puede completar en una iteración. Por eso, es recomendable que las iteraciones mantengan una **duración constante**, lo cual evita inconsistencias o extrapolaciones incorrectas.

Además, si es posible, hacer que las iteraciones coincidan con meses naturales puede facilitar la coordinación con otros departamentos y áreas de la

<b>METODOLOGÍA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE</b>			
<b>PROCESO:</b>	GESTIÓN TIC		
<b>PROCEDIMIENTO:</b>	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
<b>FECHA DE APROBACIÓN:</b>	<b>CÓDIGO:</b>	<b>VERSIÓN:</b>	<b>PÁGINA:</b>
30/09/2025	GTI07-MTD01	01	<b>21 de 21</b>

organización, se pueden estructurar sus mediciones de resultados y objetivos de forma mensual, trimestral o cuatrimestral conforme a las necesidades de la UAE CGN.

## 7. Referencias

- Harris, C. (s.f.). *Atlassian*. Obtenido de Artefactos del scrum ágil:  
<https://www.atlassian.com/es/agile/scrum/artifacts>
- Intelequia. (s.f.). *INTELEQUIA TECHNOLOGIES S.L.* Obtenido de intelequia.com: <https://intelequia.com/es/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>
- MacNeil, C. (07 de 02 de 2025). *asana*. Obtenido de Guía de retrospectiva Scrum: qué es y cómo hacerla: <https://asana.com/es/resources/sprint-retrospective>
- Martins, J. (19 de 01 de 2025). *asana*. Obtenido de ¿Qué es la metodología Kanban y cómo funciona?: <https://asana.com/es/resources/what-is-kanban>
- Sutherland, K. S. (2020). *scrumguides*. Obtenido de The 2020 Scrum GuideTM: <https://scrumguides.org/scrum-guide.html>