

| | | |
|---|--|-----------------------------|
|  | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 1 de 36 |

UNIVERSIDAD DE CUNDINAMARCA

MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL

**FUSAGASUGÁ
2020**

| | | |
|---|--|-----------------------------|
|  | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 2 de 36 |

UNIVERSIDAD DE CUNDINAMARCA

MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL

Elaborado por: DIRECCIÓN DE SISTEMAS Y TECNOLOGÍA

**FUSAGASUGÁ
2020**

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 3 de 36 |

TABLA DE CONTENIDO

| | |
|--|--------------------------------------|
| 1. INTRODUCCION..... | ¡Error! Marcador no definido. |
| 2. OBJETIVO..... | 7 |
| 2.1 OBJETIVO ESPECÍFICO | 7 |
| 3. ALCANCE | 8 |
| 4. TÉRMINOS Y DEFINICIONES | 8 |
| 5. APLICABILIDAD | 9 |
| 6. ROLES Y RESPONSABILIDAD..... | 9 |
| 6.1 Equipo Scrum..... | 9 |
| 6.2 Facilitador (Scrum Master)..... | 10 |
| 6.3 Analista..... | 10 |
| 6.4 Equipo de Desarrollo | 11 |
| 6.5 Stakeholders | ¡Error! Marcador no definido. |
| 7. METRICAS..... | ¡Error! Marcador no definido. |
| 7.1 Java..... | 12 |
| 7.2 Apache Tomcat | 12 |
| 7.3 Oracle..... | 12 |
| 7.4 Postgresql | 13 |
| 7.5 Netbeans | 13 |
| 8. METODOLOGÍA SCRUM..... | 13 |
| 9. RECOLECCIÓN DE REQUERIMIENTOS | 17 |
| 9.1 Obtener Información De Los Usuarios..... | 17 |
| 9.2 Especificar Los Requerimientos Técnicos Y Tecnológicos | 17 |
| 10. DESARROLLO DE SOFTWARE..... | 19 |
| 10.1 Análisis..... | 19 |

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 4 de 36 |

| | | |
|--------|--|----|
| 10.2 | Recolección De Historias De Usuario | 19 |
| 10.3 | Diseño | 19 |
| 10.4 | Realizar Diagrama Entidad – Relación | 19 |
| 10.5 | Codificación | 19 |
| 10.5.1 | Nomenclatura De Objetos | 20 |
| 10.5.2 | Manejo De Los Corchetes | 20 |
| 10.5.3 | Indentación..... | 21 |
| 10.5.4 | Objetos Web..... | 21 |
| 10.5.5 | Comentarios | 22 |
| 10.5.6 | Patrones Para El Desarrollo De Software | 22 |
| 10.5.7 | Estructura Del Directorio De Archivos..... | 22 |
| 10.5.8 | Lenguajes De Programación | 23 |
| 10.6 | Pruebas | 23 |
| 10.7 | Implementación | 30 |
| 10.7.1 | Especificaciones Entregables Del Proyecto..... | 30 |
| 11. | ADMINISTRACIÓN SISTEMAS DE INFORMACIÓN | 30 |
| 11.1 | Pruebas De Afinamiento:..... | 30 |
| 11.2 | Implantación: | 31 |
| 11.2.1 | Desarrollo Interno:..... | 31 |
| 11.2.2 | Aplicación externa (nuevas funcionalidades): | 31 |
| 11.3 | Parametrización:..... | 31 |
| 11.4 | Soporte:..... | 32 |
| 12. | TIPOS DE SOLUCION INFORMATICA | 33 |
| 12.1.1 | Actualización | 33 |
| 12.1.2 | Módulo | 33 |

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 5 de 36 |

| | | |
|--------|-----------------------------------|----|
| 12.1.3 | Aplicación | 33 |
| 12.1.4 | Sitio Web | 33 |
| 13. | BASES DE DATOS | 33 |
| 14. | DATOS ABIERTOS..... | 33 |
| 15. | SEGURIDAD DE LA APLICACIÓN | 34 |
| 15.1 | Identificación | 34 |
| 15.2 | Autenticación | 34 |
| 15.3 | Autorización..... | 34 |
| 16. | LINEAMIENTOS DE LEY | 35 |
| 16.1 | Normatividad | 35 |
| 16.2 | Acuerdo de Confidencialidad | 35 |
| 17. | BIBLIOGRAFIA Y WEBGRAFÍA..... | 35 |

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 6 de 36 |

1. INTRODUCCIÓN

El presente manual tiene como finalidad describir los estándares y lineamientos establecidos por la Dirección de Sistemas y Tecnología de la Universidad de Cundinamarca, para el desarrollo de productos y/o proyectos de software, elaborados por el área en mención, estudiantes y proveedores externos, aplicando las prácticas y normatividad vigente establecida por el Ministerio de Tecnologías de la Información y las Comunicaciones MINTIC en los criterios de Planeación, Diseño y Gestión de los sistemas de información establecidos en el Manual de Gobierno en línea (ahora Gobierno Digital), así como el marco de referencia de Arquitectura TI en la guía del dominio de Sistemas de Información.

El manual tendrá como función describir las métricas, metodologías, nomenclatura, seguridad y lineamientos de ley, que regularan la elaboración, modificación e implementación de software institucional.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 7 de 36 |

2. OBJETIVO

Construir una política de desarrollo de software unificada con base en los lineamientos, metodologías, infraestructura ya existente y proyecciones de Sistemas de Información de la institución, que permita mejorar la calidad y estandarizar la elaboración de productos de software dentro de la Universidad de Cundinamarca para dar cumplimiento con las disposiciones legales, proyecciones tecnológicas y metas estratégicas de la institución.

2.1 OBJETIVO ESPECÍFICO

- Establecer directrices que organicen, estructuren y agilicen el proceso de desarrollo de software para el Área de Sistemas de Información, estudiantes y proveedores externos que elaboren proyectos y/o productos para la Universidad de Cundinamarca.
- Unificar la documentación y lineamientos existentes en la Universidad de Cundinamarca, a fin de estructurar una única política de desarrollo de software, que cumpla con las disposiciones legales, proyecciones tecnológicas y metas estratégicas de la institución.
- Dar cumplimiento a la normatividad vigente en la Política de Gobierno en Línea (ahora Gobierno Digital) en cuanto a la Planeación, Diseño y Gestión de los sistemas de información.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 8 de 36 |

3. ALCANCE

El alcance del presente manual y sus políticas está destinado a todo el personal de desarrollo que labore en la Universidad de Cundinamarca, así como estudiantes y proveedores externos que deseen elaborar productos y/o proyectos de software para la institución.

4. TÉRMINOS Y DEFINICIONES

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Sprint: Es un intervalo prefijado durante el cual se crea un incremento de producto "Hecho o Terminado" utilizable, potencialmente entregable.

Product Backlog: Es una lista ordenada de todo lo que podría necesitarse en el producto y es la única fuente de requerimientos para los cambios que se realizarán en el producto.

Aplicativo Web: Aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

MVC: Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Base de datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Esquema BD: Puede referirse a una representación visual de una base de datos, a un conjunto de reglas que rige una base de datos, o bien, a todo el conjunto de objetos que pertenecen a un usuario en particular.

Librerías: Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

Subversión: Es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 9 de 36 |

5. APLICABILIDAD

La presente política será aplicable a todo producto y/o proyecto de software desarrollado por el Área de Sistemas de la información (Desarrollo y Asistencia a Usuarios), estudiantes y proveedores externos.

6. ROLES Y RESPONSABILIDAD

6.1 Equipo Scrum

El equipo Scrum es el equipo encargado de la realización del proyecto y que este cumpla con todas y cada una de las necesidades establecidas. Para llevar a cabo la construcción de proyectos de software, se establecen Siete (7) roles específicos; dichos roles son; Facilitador (Scrum Máster) o Miembro del Equipo de Desarrollo, los Interesados (Stakeholder), el analista y el equipo de desarrollo el cual están los roles de Desarrolladores y Codificadores.

Existen otros roles que pueden estar implicados en el proyecto, pero no se consideran "propios" del proyecto. Sin embargo, es necesario que estas personas mantengan cierto comportamiento para que un proyecto Scrum tenga éxito (por ejemplo, proveedores Externos de los cuales se requiera cierto tipo de acción o información para el avance del proyecto).

El cliente (sea interno o externo) también deberá entender y adoptar el marco Scrum, dado que tanto la manera de relacionarse con el equipo, como la manera en que el proyecto se entrega varían en Scrum.

Los Equipos Scrum son grupos Autoorganizados y Multifuncionales. Los Equipos Autoorganizados eligen la mejor manera de realizar su trabajo internamente, en lugar de ser dirigidos por externos al grupo. Los Equipos multifuncionales integran todas las competencias necesarias para realizar el trabajo, sin depender de recursos que no forman parte del equipo. El modelo de Equipo de Scrum está diseñado para potencia la flexibilidad, creatividad y productividad.

Los Equipos Scrum entregan productos de manera iterativa e incremental, facilitando las oportunidades de retroalimentación. Las entregas incrementales de producto "Terminados" garantizan que siempre está disponible una versión potencialmente útil del producto en desarrollo.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 10 de 36 |

6.2 Facilitador (Scrum Master)

El Facilitador es responsable de asegurar la correcta comprensión y aplicación de Scrum en su teoría, prácticas y normativas. También el Facilitador brinda soporte y liderazgo al equipo Scrum. Ayuda a los que están fuera del equipo Scrum a entender que interacciones con el equipo son útiles y cuáles no. También da soporte, tanto a externos como miembros del equipo, para dirigir estas interacciones y maximizar el valor creado.

El facilitador brindara el soporte necesario a todos y cada uno de los roles establecidos, en cuanto a los Interesados le brindara una asesoría con el fin de establecer técnicas para una gestión Eficiente de la Pila del Producto (Backlog de producto), entender la necesidad de registrar los elementos del Backlog de Productos de forma claras y concisa, garantizar que los Interesados puedan organizar la Pila del Producto (Backlog) para maximizar su valor y entender y practicar la agilidad y facilitar la entrega de Eventos Scrum.

El equipo de desarrollo estará guiado por el Facilitador con la finalidad de crear productos de alto valor, eliminar obstáculos para el avance del Equipo de Desarrollo, facilitar la entrega de Eventos Scrum, plasmar ideas organizacionales que no se han adoptado bajo la metodología Scrum y además de brindar capacitación para crear ambientes autoorganizados y multifuncionales.

En cuanto a la Universidad el facilitador brindara soporte en temas para la adopción de la metodología Scrum, liderando y capacitando a las Áreas Interesadas en la adopción de Scrum, planear la implementación de Scrum dentro de la organización, ayudar a los empleados y partes interesadas a comprender todos y cada uno de los artefactos utilizados para llevar a cabo el proyecto e incitar cambios que incrementen la productividad del equipo Scrum

6.3 Analista

El analista al igual que el Facilitador busca orientar y asegurar que los interesados y/o clientes comprendan y aplicación de Scrum en su teoría, prácticas y normativas. Además de brindar soporte a tema específicos como la recopilación de las necesidades, la importancia de registrar las necesidades del Backlog de Productos de forma clara, garantizar que los Interesados puedan organizar la Pila del Producto (Backlog); también será el encargado de efectuar la revisión de las tareas realizadas por parte del equipo de desarrollo con la finalidad de validar el funcionamiento e ir estableciendo un control de calidad.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 11 de 36 |

Además, una vez se termine la realización de un Sprint y este se considere un producto de calidad y totalmente funcional, el analista será el encargado de efectuar las reuniones necesarias con los interesados, con la finalidad de llevar a cabo la revisión de las necesidades ejecutadas e identificar si dicha iteración es un producto potencialmente entregable y que pueda ser totalmente funcional.

Debido a lo anterior el analista debe llevar el control de cada una de las tareas realizadas las cuales serán plasmadas en los artefactos generados para dicha gestión, se contará con una revisión diaria la cual se realizará de la mano con el equipo de desarrollo y las reuniones mensuales las cuales se ejecutarán en compañía del equipo Scrum.

6.4 Equipo de Desarrollo

El Equipo de Desarrollo está conformado por profesionales y técnicos encargados de realizar el trabajo para la entrega de productos incrementales “Culminados” al final de cada iteración (Sprint).

Los desarrolladores al igual que los codificadores están encargados de la construcción de los proyectos de software institucional, basado en las necesidades establecidas institucionalmente y la prioridad establecida por la alta dirección.

Con el fin de llevar un control y mantener documentado cada uno de los proyectos elaborados, el rol del desarrollador se encargará de llevar a cabo la administración de los proyectos en ambientes productivos, además de estar a cargo de primera mano de brindar soporte a los productos si es necesario, también llevan a cabo el proceso de documentar de proyectos en construcción, de la mano con el analista encargado del proyecto. Por último, los desarrolladores serán los guías para el personal encargado de la codificación en cuanto el Facilitador designe un “Encargado” del proyecto a elaborar.

6.5 Stakeholder

Los Interesados son grupos de personas que no están asociados directamente al proyecto de desarrollo pero que, si deben ser tenidos en cuenta, por ser personas interesadas en el mismo, tales como directivos, jefes, proveedores, entre otros.

| | | |
|---|--|-----------------------------|
|  | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 12 de 36 |

7. MÉTRICAS

La Universidad de Cundinamarca para llevar a cabo la ejecución de proyectos de software cuenta con una infraestructura orientada al uso de aplicaciones Web desarrolladas en Lenguaje Java, utilizando una estructura de codificación orientada al uso de patrones de diseño MVC y bases de datos relacionales. A continuación, nombramos las herramientas utilizadas y las versiones para contar con el soporte adecuado apoyado en la infraestructura tecnológica actual:

| HERRAMIENTA | VERSIÓN |
|---------------------|-----------------------|
| JAVA | JDK 1.6 |
| TOMCAT | Apache Tomcat 7 |
| ORACLE | Oracle Database 11g |
| POSTGRESQL | PGSQL 8.4.20 |
| IDE NETBEANS | NetBeans 8 o superior |

7.1 Java

El JDK es un entorno de desarrollo para crear aplicaciones, applets y componentes utilizando el lenguaje de programación Java. Para llevar a cabo el desarrollo de los sistemas de información, se utiliza dicho kit de herramientas de desarrollo en su versión 1.6, con el fin de realizar el compilado y ejecución sobre dicha versión y garantizar compatibilidad con la infraestructura de la Universidad de Cundinamarca.

7.2 Apache Tomcat

Apache Tomcat permite la implementación de Java Servlets y JavaServer Pages (JSP) para promover un entorno de servidor Java eficaz. Además de acceder a los recursos para la configuración y usar el lenguaje de marcado extensible (XML) para configurar proyectos. Igualmente debemos de trabajar con la versión 7 de Tomcat con el fin de brindar compatibilidad con la infraestructura de la Universidad de Cundinamarca.

7.3 Oracle

La Universidad de Cundinamarca dentro de la infraestructura tecnológica, cuenta con el motor de bases de datos Oracle en su versión 11g, dicho motor de base de datos

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 13 de 36 |

nos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de la Universidad de Cundinamarca. El uso del motor de bases de datos nos ayuda con la gestión de las bases de datos relacionales, con la ejecución de procedimientos y transacciones en línea o datos de procesamiento analíticos en línea. Se pueden crear tablas para almacenar datos y objetos de base de datos como índices, vistas y procedimientos almacenados para ver, administrar y proteger los datos.

7.4 Postgresql

Para la gestión de accesos mediante la plataforma institucional, se cuenta con el motor de bases de datos Postgresql en su versión 8.4.20, la cual se utiliza para la gestión de roles y permisos desde la plataforma institucional, con el fin de brindar diversos niveles de servicios y funcionalidades a cada uno de los interesados asociados a la Universidad.

7.5 Netbeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Este Entorno brinda diversos módulos para la gestión durante la elaboración de los proyectos de software, por medio de la gestión de servicios como apache y los motores de bases de datos relacionales.

Con la utilización de este Entorno de desarrollo se busca garantizar un estándar de proyectos y evitar problemas de compatibilidad, además de permitirnos utilizar diversas herramientas para el control de versiones como Git y Subversión.

8. METODOLOGÍA SCRUM

Cuando hablamos de SCRUM debemos saber que es un marco de trabajo para el desarrollo de proyectos de diferente índole, se caracteriza por ser un marco enfocado a la adaptabilidad, los tiempos de respuestas deben ser rápidos y de calidad, las entregas deben ser progresivas e iterativas, en cuando a las necesidades deben ser flexibles y que permitan al cliente solicitar cambios si es necesario, todo con el fin de mantener un nivel de aceptación, comunicación y confianza adecuados. Con la finalidad de crear entornos de trabajo amigable y transparente, además que se genere una responsabilidad mutua entre los interesados y el nivel de progreso sea continuo para garantizar proyectos de calidad.

El marco de trabajo SCRUM tiene un punto diferencial de las metodologías tradicionales el cual lo orienta al conocimiento, ya que este se basa en la generación

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 14 de 36 |

de experiencia el cual es un punto a favor para la toma de decisiones, mediante la aplicación de repetición de procesos dando un enfoque iterativo e incremental lo que garantiza una mejora continua y ayuda con la reducción de los riesgos que se puedan presentar con la elaboración de los proyectos, también esta metodologías tiene un punto a favor ya que mejora la comunicación con los interesados y sobre todo con el cliente lo que nos demuestra ciertas particularidades:

- Una mejor tendencia a la respuesta y mejora continua.
- Elaboración de entregables funcionales y de forma iterativa e incremental.
- Comunicación más transparente por medio de comunicación directa con los interesados.
- Elaboración de productos o entregables necesarios.
- Un grupo de trabajo con motivación y compromiso ayuda a generar productos de manera más efectiva.

En busca de diseñar y desarrollar sistemas de información de calidad, respetando los plazos pactados, cumplir con las necesidades propuestas por el cliente y disminuir los tiempos de respuesta, con el fin de mejorar la atención de las diversas áreas y lograr satisfacer la demanda presentada se ha optado por la utilización de formatos bajo los instrumentos que propone la metodología SCRUM, en esta metodología se trabaja con dos(2) artefactos principales son los que orientan desde el inicio del proyecto con la consolidación de las tareas a realizar este artefacto se conoce como pila del Producto (Product Backlog), el otro artefacto se utiliza para establecer un conjunto de actividades a realizar con el fin de generar un entregable en un tiempo corto, con las actividades allí establecidas, este tiene como nombre pila del Sprint (Sprint Backlog), para diligenciar estos artefactos, la metodología recomienda la realización de reuniones periódicas con el fin de validar el estado del proyecto y establecer las actividades a realizar en cada Sprint creado, dichas reuniones son:

➤ **Reunión Inicial, levantamiento de necesidades**

Al momento de iniciar un proyecto se debe realizar una reunión con el equipo Scrum y los interesados, con el fin de establecer todas las necesidades y/o funcionalidades con las que va a contar el proyecto, todo esto para garantizar la mejora en los procesos internos con los que cuenta la Universidad de Cundinamarca.

Debido a lo anterior se procede a diligenciar en la pila del producto todas y cada una de las necesidades identificadas, brindando una descripción corta

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 15 de 36 |

de esta, se establece una prioridad y los responsables encargados de la ejecución, por último, se brinda un estimado de tiempo para su ejecución y entrega.

➤ **Planeación de la lista de tareas (Sprint Planning Meeting)**

Una vez aterrizadas las necesidades el equipo Scrum procede a realizar el estudio de las funcionalidades propuestas en la Pila del producto, con el fin de establecer tareas puntuales para la realización de las funcionalidades, estas deben ser priorizadas según lo requiera el Interesado del producto, además se deben establecer tiempos de ejecución y los responsables específicos de cada tarea, la recopilación de estas tareas se conoce como Sprint y son diligenciadas en la Pila del Sprint donde se establece un plazo máximo de 30 días para culminar el Sprint.

➤ **Reunión Diaria (Daily Scrum)**

Las reuniones diarias se deben realizar mediante el Sprint avanza, en esta se deben establecer 3 preguntas ¿Que realizo?, ¿Que va a Realizar? y ¿Tuvo algún impedimento?, estas preguntas debes ser resueltas por el personal encargado del desarrollo (Equipo de Desarrollo) y el Scrum master deberá dar solución a las incógnitas o problemas que pueda tener el equipo.

➤ **Revisión del Sprint terminado (Sprint Review)**

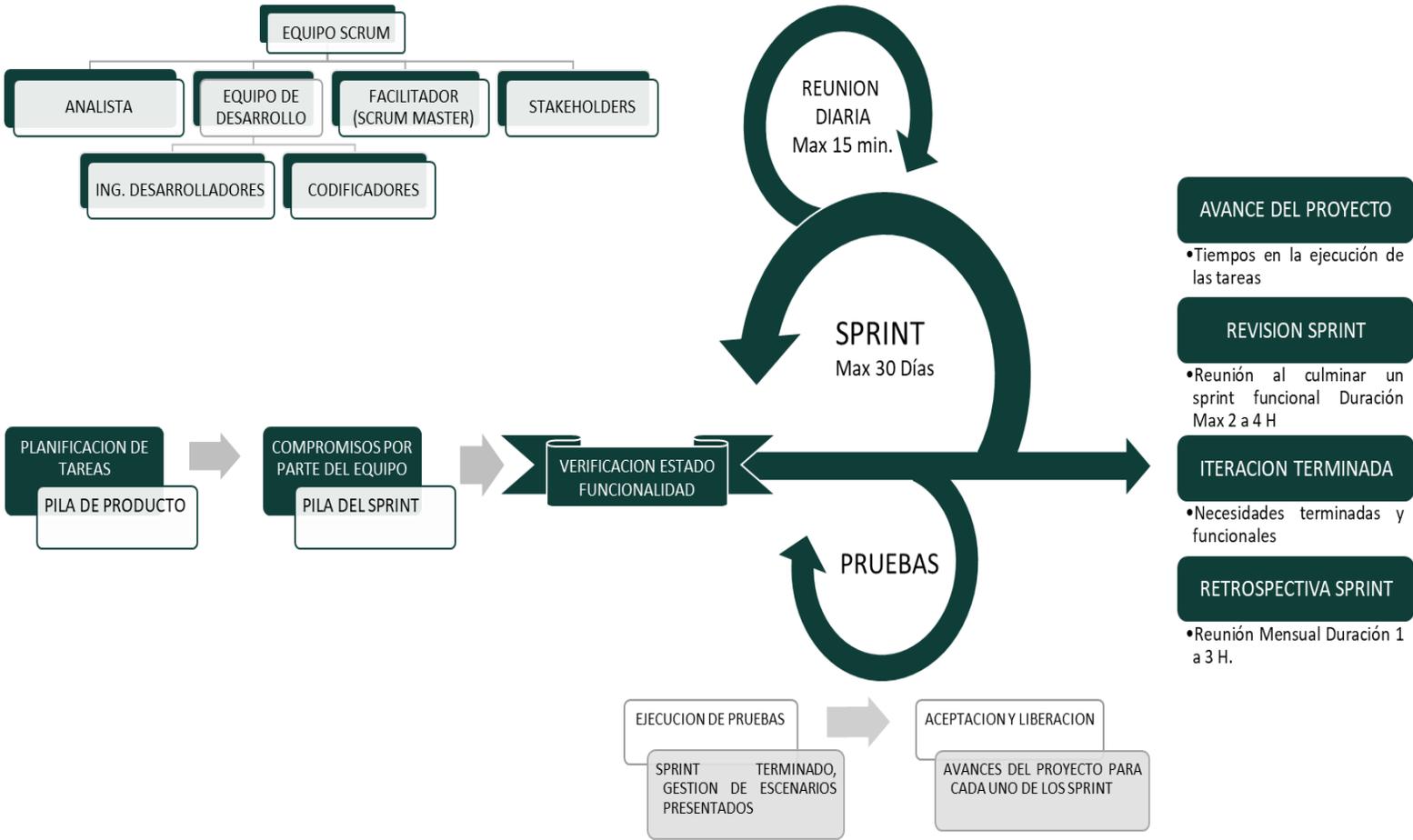
Al finalizar un Sprint se debe realizar la reunión con el fin de establecer y validar que dicho Sprint este completamente funcional y este pueda ser presentado a los interesados, cumpliendo con los pasos establecidos y el proceso de pruebas previamente realizado.

➤ **Revisión de las tareas culminadas (Sprint Retrospective)**

La reunión de retrospectiva busca realizar una revisión de los objetivos propuestos de los Sprint terminados. Se lleva un control de las buenas y malas cosas que se presentaron y de esta manera evitar cometer errores en procesos de desarrollo, pruebas y gestión del proyecto.

Una vez visto los artefactos y reuniones a utilizar, se generó un gráfico de comportamiento donde interactúan cada uno de estos elementos, se establece un tiempo para cada una de las reuniones y como son los ciclos durante cada Sprint. Ver ilustración

| | | |
|---|--|-----------------------------|
|  | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 16 de 36 |



| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 17 de 36 |

9. RECOLECCIÓN DE REQUERIMIENTOS

9.1 Obtener Información De Los Usuarios

Al momento de iniciar un proyecto de software se realiza la entrevista inicial con los interesados del proyecto o el Área solicitante y el equipo Scrum de manera que se pueda establecer el tipo de solución informática que pueda satisfacer las necesidades específicas del solicitante.

9.2 Especificar Los Requerimientos Técnicos Y Tecnológicos

Se hace un listado de cada requerimiento funcional, no funcional o externo, definidos así:

- Un requerimiento funcional puede ser una descripción de lo que un sistema debe hacer. Este tipo de requerimiento especifica algo que el sistema entregado debe ser capaz de realizar.
- Un requerimiento no funcional: de rendimiento, de calidad, etc.; especifica algo sobre el propio sistema, y cómo debe realizar sus funciones. Algunos ejemplos son la disponibilidad, el testeo, el mantenimiento, la facilidad de uso, etc.
- Otros tipos de limitaciones externas, que afectan en una forma indirecta al producto. Estas pueden ir desde la compatibilidad con cierto sistema operativo hasta la adecuación a leyes o regulaciones aplicables al producto.

Posteriormente se revisa que satisfagan las siguientes características, en caso contrario, deben ser reformulados:

- Necesario: Lo que pida un requerimiento debe ser necesario para el producto.
- No ambiguo: El texto debe ser claro, preciso y tener una única interpretación posible.
- Conciso: Debe redactarse en un lenguaje comprensible por los inversores en lugar de uno de tipo técnico y especializado, aunque aun así debe referenciar los aspectos importantes
- Consistente: Ningún requerimiento debe entrar en conflicto con otro requerimiento diferente, ni con parte de otro. Asimismo, el lenguaje empleado entre los distintos requerimientos debe ser consistente también.
- Completo: Los requerimientos deben contener en sí mismos toda la información necesaria, y no remitir a otras fuentes externas que los expliquen con más detalle.

| | | |
|---|--|-----------------------------|
|  | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 18 de 36 |

- Alcanzable: Un requerimiento debe ser un objetivo realista, posible de ser alcanzado con el dinero, el tiempo y los recursos disponibles.
- Verificable: Se debe poder verificar con absoluta certeza, si el requerimiento fue satisfecho o no. Esta verificación puede lograrse mediante inspección, análisis, demostración o testeo. ¹

¹ Especificación de Requisitos según el estándar de IEEE 830

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 19 de 36 |

10. DESARROLLO DE SOFTWARE

10.1 Análisis

Para llevar a cabo un proyecto de software debemos tener claro cuáles son los objetivos del proyecto a cumplir una vez este sea terminado y se encuentre en ambientes de producción. Para ello debemos de realizar un análisis previo al desarrollo del proyecto, donde establezcamos cuales son dicho objetivos, además de establecer el alcance, el impacto del proyecto, cuáles serán las áreas y usuarios beneficiados. Además de establecer una problemática a resolver basado en un análisis de causa efecto.

10.2 Recolección De Historias De Usuario

Se interactúa con el usuario para la recolección de las historias que son la descripción de la funcionalidad que se le quiere dar al aplicativo, se clasifica cada historia por la complejidad de llevarla a desarrollo y la prioridad para el usuario sobre el aplicativo.

10.3 Diseño

El equipo Scrum determina la mejor solución a la necesidad teniendo en cuenta el contexto de la infraestructura actual y la proyección tecnológica de la institución, contemplando normatividad externa e interna, documentos estratégicos, procedimientos y normas de gestión que estandaricen las actividades para el desarrollo a realizar, esta actividad es primordial para determinar un alcance al desarrollo y establecer los usuarios que intervendrán en la solución a implementar.

10.4 Realizar Diagrama Entidad – Relación

Diagrama de los esquemas, bases de datos y tablas, con los respectivos campos y relaciones existentes entre ellos.

10.5 Codificación

Para el propósito de implementación del software se deben respetar los siguientes estándares de desarrollo:

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 20 de 36 |

10.5.1 Nomenclatura De Objetos

- Ningún nombre de objeto llevará tilde o caracteres especiales salvo la línea baja (underline), que se usará para separar palabras o contracciones diferentes.
- Ningún nombre de objeto superará los 20 caracteres.
- Los nombres de tablas irán en MAYÚSCULA. Los nombres de los campos y llaves irán en MAYÚSCULA.
- Los nombres de los objetos irán en singular, ya que el nombre representa la entidad que almacena. Los nombres de las tablas serán en plural, por ser más intuitivo para el programador.
- Los nombres de los objetos serán descriptivos, no crípticos.
- Se usarán abreviaturas cuando sean necesarias y serán suficientemente claras.
- Se usarán abreviaturas estándar siempre que sea posible.
- Los nombres de claves primarias irán precedidos por las siglas PK y los de clave foránea por FK.
- Cada nombre de campo tendrá un prefijo consistente de cuatro letras que identifiquen la tabla en la que se encuentra, seguidas de una línea baja. El nombre de un campo no debería superar los 32 caracteres.
- Tipos de Identificadores: FuncionUno (), CLASE, subrutina
- Variables locales de CLASES: I_nombre_variable.
- Los nombres de las variables que lleven o traigan datos de campos de la Base de Datos tendrán el mismo nombre que el campo siempre que sea posible: unid_nombre contiene la información traída del campo UNID_NOMBRE de la tabla UNIDAD.

10.5.2 Manejo De Los Corchetes

Los corchetes irán justo bajo la instrucción o función que los genere.

```

if (condición1)
{
    If (condición2)
    {
        FuncionUno();
    }
}

```

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 21 de 36 |

10.5.3 Indentación

- Se obtiene consistencia si todas las instrucciones se comienzan en una misma columna.
- En los lenguajes en que se hagan declaraciones sobre las variables a utilizar, la información quedará más claramente representada si los atributos declarados se alinean en forma vertical.
- El uso de líneas en blanco ayuda a mostrar con más claridad las relaciones existentes entre distintos ítems agrupados en las declaraciones
- Las instrucciones son mucho más fáciles de localizar y de cambiar si no se escribe más de una instrucción por línea.
- La visión de control de las estructuras lógicas o de los bloques se clarifica si las indentaciones controladas son idénticas por alguna cantidad constante. Se sugiere una indentación de tres espacios.
- Máximo 4 columnas de indentación.
- El espacio entre indentaciones será de 1 tabulación.

10.5.4 Objetos Web

- Las Variables de Sesión tendrán el sufijo “_ses”: unidad_ses.
- Los nombres de los objetos de las páginas tendrán sufijos de la siguiente forma:
 - ✓ Cajas de texto: _txt
 - ✓ Select: _sel
 - ✓ Grupo de opciones: _opc
 - ✓ Campos de texto ocultos: _ocu
- Los nombres de los forms empezarán siempre con la palabra form.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 22 de 36 |

10.5.5 Comentarios

Se utilizarán comentarios en los siguientes casos.

- En los inicios de las clases y funciones cuyo objetivo no sea evidente: una clase llamada “COMUN _ MULTIPLO” no necesita comentario, mientras una llamada “REPORTE_COMPLETO” sí.
- Como etiquetas en el inicio y fin de los bloques más grandes, de manera que sea fácil identificar el grupo de instrucciones dentro de una condición o ciclo.
- Todos los otros casos en los que el programador considere que el código no puede ser entendido al leerse o que haya algún dato que necesite recordar.

10.5.6 Patrones Para El Desarrollo De Software

- DAO (Data Access Objetc): Cuando la capa de lógica de negocio necesite interactuar con la base de datos, va a hacerlo a través de la API que le ofrece DAO. Generalmente esta API consiste en métodos CRUD (Insertar, Listar, Actualizar y Borrar). En una aplicación, hay tantos DAOs como modelos. Es decir, en una base de datos relacional, por cada tabla, habría un DAO.
- VO (Value Object) API que permite el almacenamiento y manipulación de información, API encargada de transportar la información recopilada en formularios o subrutinas al API DAO la cual se encargara de su mediación con la base de datos.
- JSTL (JavaServer Pages Standard Tag Library): Es un componente de Java EE. Extiende las ya conocidas JavaServer Pages (JSP) proporcionando cuatro bibliotecas de etiquetas (Tag Libraries) con utilidades ampliamente utilizadas en el desarrollo de páginas web dinámicas. Estas bibliotecas de etiquetas extienden de la especificación de JSP (la cual a su vez extiende de la especificación de Servlet). Su API permite además desarrollar bibliotecas propias de etiquetas.

10.5.7 Estructura Del Directorio De Archivos

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 23 de 36 |

Se debe de tener una estructura de archivos para la aplicación que consta de:

- Un almacenamiento fijo para archivos adjuntos en un Servidor FTP.
- Un directorio de imágenes
- Un directorio para hojas de estilos.
- Un directorio para archivos java script o plugins.
- Utilizar los Paquetes de Fuentes para las Clases DAO y VO.

10.5.8 Lenguajes De Programación

- Lenguaje de Servidor: JSP.
- Objetos Web: HTML, java script, CSS.
- Clases: java.

10.6 Pruebas

| PRUEBAS DE BASES DE DATOS |
|---|
| <p>PRUEBAS DE INTEGRIDAD DE DATOS</p> <p>OBJETIVO: Verificar la calidad de los datos, mediante inspección en cada una de las bases de datos de todos los sistemas, para evitar incongruencia en los datos</p> <p>METODOLOGÍA</p> <ol style="list-style-type: none"> 1. Identificar las herramientas y recursos, el entorno físico que incluye hardware, software y configuraciones de red, incluyendo Herramientas para backup y restore de la base de datos. 2. Contar con un listado de Scripts de carga de información y utilidades para ejecutarlos. 3. Definir los datos de las pruebas. 4. Invocar cada método de acceso y procesos, alimentándolos con datos válidos e inválidos o consultando información. 5. Inspeccionar la base de datos para asegurar que los datos han sido cargados como se pretendía, todos los eventos ocurrieron correctamente, o que la información fue recabada correctamente 6. Aplicar el diseño de la prueba. Desarrollar las pruebas de integridad de acuerdo con el diseño del plan. |

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 24 de 36 |

7. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
8. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario.

PRUEBAS DE DESEMPEÑO

OBJETIVO: Analizar y evaluar las características del software relacionadas con el desempeño como el tiempo de respuesta, throughput y capacidad.

Tiempo de respuesta: Es el intervalo de tiempo que transcurre entre la solicitud de un usuario al sistema y la respuesta de es este último

Thoroughput: Es la tasa a la cual el sistema puede atender las peticiones

Capacidad: máxima cantidad de trabajo útil que se puede realizar por unidad de tiempo.

METODOLOGÍA

1. Identificar las herramientas y recursos, el entorno físico que incluye hardware, software y configuraciones de red. Tener conocimiento sobre esto desde el inicio permite identificar problemas en las pruebas en fases tempranas del proyecto.
2. Identificar los criterios de aceptación de rendimiento, determinado el tiempo de respuesta, el rendimiento y la utilización de los recursos
 - Tiempo de respuesta le corresponde al usuario
 - El rendimiento al negocio
 - Y la utilización de recursos al sistema
3. Planificar y diseñar las pruebas, determinado la variabilidad de los usuarios y definiendo los datos de las pruebas.
4. Aplicar el diseño de la prueba. Desarrollar las pruebas de rendimiento de acuerdo con el diseño del plan.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 25 de 36 |

5. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
6. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario.

PRUEBAS DE STRESS

OBJETIVO: Investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos.

METODOLOGÍA

1. Utilizar scripts que puedan ser ejecutados en una máquina, deben ser repetidos con múltiples clientes (virtuales o actuales).
2. Para probar recursos limitados, las pruebas se deben correr en un servidor con configuración reducida (o limitada).
3. Para las pruebas de stress restantes, deben utilizarse múltiples clientes, ya sea corriendo los mismos scripts o scripts complementarios para producir el peor caso de volumen de transacciones.
4. Aplicar el diseño de la prueba. Desarrollar las pruebas de stress de acuerdo con el diseño del plan.
5. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
6. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario.

PRUEBAS DE VOLUMEN

OBJETIVO: someter al sistema a grandes volúmenes de datos para determinar si el mismo puede manejar el volumen de datos especificado en sus requisitos.

METODOLOGÍA

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 26 de 36 |

1. Utilizar scripts que puedan ser ejecutados en una máquina, deben ser repetidos con múltiples clientes (virtuales o actuales).
2. Deben usarse múltiples clientes, ya sea corriendo las mismas pruebas o pruebas complementarias para producir el peor caso de volumen
3. Se utiliza un tamaño máximo de Base de datos. (actual, escalado o con datos representativos) y múltiples clientes para correr consultas simultáneamente para períodos extendidos.
4. Aplicar el diseño de la prueba. Desarrollar las pruebas de volumen de acuerdo con el diseño del plan.
5. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
6. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario.

PRUEBAS DE DISEÑO

PRUEBA DE INTERFAZ DE USUARIO

OBJETIVO: Verificar que La navegación a través de las aplicaciones reflejen las funciones pedidas por el usuario en la captura de los requerimientos, incluyendo la navegación de ventana a ventana, de campo a campo, y la utilización de métodos de acceso y además que este diseñado conforme a los estándares de la universidad

METODOLOGÍA

1. Identificar las herramientas y recursos, el entorno físico que incluye hardware, software y configuraciones de red. Tener conocimiento sobre esto desde el inicio permite identificar problemas en las pruebas en fases tempranas del proyecto.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 27 de 36 |

2. Tener conocimiento con respecto a los requerimientos del usuario y el conocer el diseño estandarizado.
3. Realizar una verificación de ortografía
4. Verificar que la para la creación de pantallas (input boxes, combo boxes, list boxes y check buttons, etc) no cambiar radicalmente sus atributos de despliegue (colores, formas) y estándares tradicionales, para lograr que el usuario sepa intuitivamente como usarlo
5. Verificar que los botones de las funciones básicas siempre se muestren del mismo modo
6. Aplicar el diseño de la prueba. Desarrollar las pruebas de interfaz de usuario de acuerdo con el diseño del plan.
7. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
8. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario.

PRUEBAS DE NAVEGABILIDAD

PRUEBAS DE FUNCIONALIDAD

OBJETIVO: Asegurar la correcta funcionalidad de la aplicación, incluyendo la navegación, la entrada de datos, el procesamiento, la consulta de información, la actualización y la búsqueda.

METODOLOGÍA

1. Identificar las herramientas y recursos, el entorno físico que incluye hardware, software y configuraciones de red. Tener conocimiento sobre esto desde el inicio permite identificar problemas en las pruebas en fases tempranas del proyecto.
2. Realizar una verificación de formularios validando el ingreso de datos para asegurar que sean bien ingresados y guardados.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 28 de 36 |

3. Validación de campos obligatorios, que sean ingresados los campos requeridos de manera obligatoria y también que estén marcados de alguna manera para dar a conocer a los usuarios la obligatoriedad de ingresar la información en estos campos adicional a esto se debe explicar esta condición claramente
4. Se debe hacer una sintaxis de ingreso, validando que en los campos sean ingresados con datos validos
5. Confirmar en la tabla de destino que efectivamente se estén enviando los datos de la manera en que se ha previsto.
6. Comprobar que los diferentes formularios funciones en diferentes versiones de navegadores web, sistemas operativos y de tipos de conexiona internet.
7. Confrontar que los botones interactivos que permiten imprimir, enviar email, regresar a la página anterior, inicio, etc. estén realizando correctamente la acción indicada.
8. En los sistemas de búsqueda se debe validar que efectivamente permitan encontrar documentos existentes en el sitio.
9. Aplicar el diseño de la prueba. Desarrollar las pruebas de funcionalidad de acuerdo con el diseño del plan.
10. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
11. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario

PRUEBAS DE SEGURIDAD Y CONTROL DE ACCESO

OBJETIVO: Verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido y que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

METODOLOGÍA:

1. Identificar cada tipo de usuario, las funciones y los datos que tiene autorizados

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 29 de 36 |

2. Crear pruebas para cada tipo de usuario y verificar cada permiso, creando transacciones para cada tipo de usuario
3. Modificar tipo de usuarios y volver a ejecutar las pruebas y verificar en cada caso si los datos o funciones adicionales quedan correctamente permitidos o denegados
4. Aplicar el diseño de la prueba. Desarrollar las pruebas de seguridad y control de acuerdo con el diseño del plan.
5. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
6. Analizar los resultados; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario

PRUEBAS CON EL USUARIO

PRUEBAS DE ACEPTACIÓN

OBJETIVO: Determinar si el usuario final acepta o rechaza el aplicativo web desarrollado

METODOLOGÍA

1. Haber ejecutado las pruebas mencionadas anteriormente, corregido fallas encontradas y haber realizado cambios sugeridos
2. Al ejecutar las pruebas de aceptación el usuario final comprueba si la aplicación sigue con los estándares establecidos
3. El usuario realiza una navegación por la aplicación verificando los estándares de GUI del cliente
4. Valida objetos gráficos contra los estándares establecidos
5. Aplicar el diseño de la prueba. Desarrollar las pruebas de aceptación de acuerdo con el diseño del plan.
6. Monitorizar las pruebas. Validar las pruebas, los datos de las pruebas, y recoger los resultados
7. Analizar los resultados y realizar un informe; Volver a priorizar el resto de las pruebas y volver a ejecutarlas de ser necesario

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 30 de 36 |

10.7 Implementación

10.7.1 Especificaciones Entregables Del Proyecto

- Soportes del desarrollo del proyecto: Esto incluirá historias de usuario, copia de formato de solicitud, control de cambios, instrumentos Scrum.
- Documento de Requerimientos: Un documento con la especificación de requerimientos clasificados por funcionales y no funcionales.
- Diagramas: Documento con el diagrama de clases, entidad relación, interfaz.
- Manual de usuario: Documento con especificaciones de uso flujo de información y navegabilidad de la aplicación.
- Manual técnico: En este documento se especificará las condiciones técnicas necesarias para el buen funcionamiento de la aplicación, deberá incluir diagrama entidad relación, diccionario de datos, versiones de exploradores (Internet Explorer, Mozilla, Opera), Servidor de base de datos, Servidor web, especificaciones técnicas de implementación.
- Paquete código fuente: archivos con código, clases, archivos de conexión a bases de datos necesarios para el funcionamiento de la aplicación.

11. ADMINISTRACIÓN SISTEMAS DE INFORMACIÓN

Dentro del mantenimiento y sostenimiento de los sistemas de información se deben tener en cuenta las principales actividades relevantes en cada sistema de información para su buen funcionamiento y uso por los diferentes usuarios. Para esto se destinan las diferentes pruebas de afinamiento por actualización de versión, implantación de nuevas funcionalidades y/o aplicaciones.

11.1 Pruebas De Afinamiento:

El propósito de esta actividad es identificar los cambios relevantes en las funcionalidades que se generan con las actualizaciones de versión y poder realizar las pruebas de afinamiento necesarias para verificar que las aplicaciones funcionen correcta y eficientemente.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 31 de 36 |

Tipos de pruebas:

- Pruebas de accesibilidad
- Pruebas de navegabilidad
- Pruebas de módulos
- Pruebas de integración
- Tiempos de respuesta en la ejecución de los procesos

Finalizadas las pruebas, es posible que se deban realizar ajustes en la programación por la presencia de errores los cuales deben ser reportados al desarrollador, al administrador de la base de datos o al proveedor externo de acuerdo al origen de la aplicación y el error encontrado.

11.2 Implantación:

11.2.1 Desarrollo Interno:

Se realizan pruebas de navegabilidad y funcionamiento con el fin de determinar que la aplicación cumpla con las necesidades del proceso y posteriormente hacer la entrega y explicación al usuario final para la puesta en marcha en el sitio productivo.

11.2.2 Aplicación externa (nuevas funcionalidades):

Al tener instaladas las aplicaciones externas, se recibe la capacitación correspondiente por parte del proveedor para conocer su manejo y funcionamiento, adicionalmente se recibe la documentación de guía funcional de la aplicación implementada de forma digital.

Una vez recibida la explicación por parte del proveedor, se verifica el funcionamiento realizando pruebas con los usuarios finales para la entrega y puesta en marcha en el sitio productivo.

La Dirección de Sistemas y Tecnología cuenta con dos tipos de soporte por parte del proveedor (asincrónico y en sitio) que permiten apoyar la solución de inconvenientes que puedan presentarse con el uso de los productos de software adquiridos.

11.3 Parametrización:

Esta actividad consiste en acondicionar los sistemas de información para el adecuado funcionamiento de los procesos, teniendo en cuenta la activación de reglas, registro de información propia de la institución, habilitación de fechas, apertura de vigencias, actualización de valores de acuerdo a normatividad vigente, entre otros, teniendo en

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 32 de 36 |

cuenta calendario académico, reglamento estudiantil, estatuto docente, estatuto administrativo y demás parámetros legales y normas que afecten el curso normal de los procesos de la Universidad.

11.4 Soporte:

La función principal del servicio de soporte a los sistemas de información es apoyar la solución de inconvenientes que puedan presentarse con el uso de las aplicaciones desarrollados interna y externamente.

Esta actividad se realiza a través de medios como mesa de ayuda, correo electrónico institucional, chat (Skype), teléfono y acompañamiento en sitio (oficina del usuario y oficina Dirección de Sistemas y Tecnología).

Por medio de la aplicación mesa de ayuda se gestionan las solicitudes en línea por la categoría de servicio APLICACIONES/PLATAFORMA, subcategoría SOPORTE A APLICACIONES, las cuales son atendidas por medio de las siguientes actividades:

- El Director de Sistemas y Tecnología recibe el requerimiento por parte de los gestores responsables y/o jefes de área de la institución y asigna un funcionario responsable de acuerdo a la solicitud.
- El funcionario analiza el requerimiento y verifica la aplicación teniendo en cuenta el tipo de solicitud.
- Dependiendo el tipo de solicitud, se orienta al usuario sobre el manejo adecuado de la aplicación, de lo contrario, el funcionario encargado de atender la solicitud realiza la actividad correspondiente ya sea parametrización, generación de reportes y bitácoras.
- Si la solicitud lo requiere, se emite una respuesta previa y se continua con la verificación y ejecución de la solución.
- Finalizada la revisión y ejecutada la solución, se emite respuesta y se cierra el requerimiento.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 33 de 36 |

12. TIPOS DE SOLUCIÓN INFORMÁTICA

12.1.1 Actualización

Modificación al código existente que no implica la creación de nuevos archivos (.JSP) o manipulación de campos o tablas de las Bases de Datos.

12.1.2 Módulo

Parte de un aplicativo que cumple al menos una de las tareas necesarias para satisfacer la necesidad del usuario, alimentándose de la salida de otro módulo o de los datos de entrada del sistema y proporcionando una salida que alimentará a otro módulo o que será la salida final del sistema.

12.1.3 Aplicación

Programa informático diseñado para facilitar al usuario la realización de un trabajo determinado. Puede ir integrada con otras aplicaciones o aislada, pero, en cualquier caso, satisface por sí misma una necesidad del usuario.

12.1.4 Sitio Web

Conjunto de documentos (**páginas web**) organizados jerárquicamente. Cada documento (página web) contiene texto y/o gráficos que aparecen como información digital en la pantalla de un ordenador. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos.

13. BASES DE DATOS

El gestor de base de datos para el desarrollo de software a ser utilizado a nivel institucional debe ser Oracle 11g ya que la universidad cuenta con la infraestructura y el licenciamiento necesario para su uso. Y debe respetar las indicaciones dadas en punto 10.5.1 NOMENCLATURA DE OBJETOS.

14. DATOS ABIERTOS

Para la definición de los datos abiertos de cada Sistema de Información institucional se destinará un apartado dentro de los instrumentos Scrum para que en la documentación de cada desarrollo se determine por el Administrador de la aplicación

| | | |
|---|--|-----------------------------|
|  | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 34 de 36 |

(Área beneficiada y/o Solicitante de la necesidad), que van a destinar como Datos Abiertos y quien seria los usuarios de estos.

15. SEGURIDAD DE LA APLICACIÓN

15.1 Identificación

Un sistema de información posee las características de identificación cuando .es capaz de reconocer usuarios individuales. La identificación es el primer paso en conseguir el acceso para el material seguro, y sirve como soporte para las subsiguientes autenticación y autorización. La autenticación e identificación son esenciales para estandarizar el nivel de acceso o autorización que se le concede a un individuo. Por lo general el nivel de identificación es dado mediante un nombre de usuario o ID; su manejo confidencial, así como su permanente actualización son claves a la hora de la implementación.

15.2 Autenticación

La autenticación ocurre cuando un control da pruebas de que el usuario posee la identidad que suministra. Existen algunos hardware en criptografía que facilitan este mecanismo.

15.3 Autorización

Después de que la identidad del usuario es autenticada, Un proceso llamado autorización da la seguridad de que un usuario ha sido específica y explícitamente autorizado para acceder, actualizar o eliminar los contenidos de un archivo, un ejemplo de este control es la activación o uso de listas de control de acceso y grupos de autorización en un ambiente de trabajo en red. Es una característica bien importante cuando se utilizan varias terminales donde las personas pueden adicionar o cambiar datos a la información contenida en bases de datos.

Un programa de seguridad exitoso en información, combina estos y otros elementos. El arte de reducir y administrar el riesgo requiere comunicación y cooperación entre todos los niveles de la organización. En otras palabras, el aseguramiento de los activos de información puede ser alcanzado solo a través de la administración cuidadosa y la concientización de todos los funcionarios, lo cual genera cultura en torno al tema de la seguridad

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 35 de 36 |

16. LINEAMIENTOS DE LEY

16.1 Normatividad

- Norma ISO 27001.
- Ley 1581 de 2012.
- Decreto 1377 de 2013.
- Resolución No. 000050 de 2018.
- Decreto 1074 de 2015.
- Decreto 088 de 2017

16.2 Acuerdo de Confidencialidad

Toda la información suministrada en cualquier modalidad de trabajo, así como el uso del hardware y software puesto a disposición por la oficina de sistemas y tecnología a sus colaboradores directos e indirectos deberá ser tratada bajo los lineamientos de la “POLÍTICA DE PROTECCIÓN DE DATOS PERSONALES” descrita en la resolución No. 000050 de 2018 y del “SISTEMA DE SEGURIDAD DE LA INFORMACIÓN DE LA UNIVERSIDAD DE CUNDINAMARCA” descrito en la resolución 088 de 2017.

17. BIBLIOGRAFÍA Y WEBGRAFÍA

NOTA: Estas políticas deberán seguir un proceso de actualización periódica sujeto a los cambios organizacionales relevantes: crecimiento de la planta de personal, cambio en la infraestructura computacional, desarrollo de nuevos servicios, entre otros.

| | | |
|--|--|-----------------------------|
| | MACROPROCESO DE APOYO | CÓDIGO: ASIM006 |
| | PROCESO GESTIÓN SISTEMAS Y TECNOLOGÍA | VERSIÓN: 2 |
| | MANUAL DE POLÍTICAS DE DESARROLLO DE SOFTWARE INSTITUCIONAL | VIGENCIA: 2020-10-19 |
| | | PAGINA: 36 de 36 |

| CONTROL DE CAMBIOS | | | | | | |
|--|----------------------------|-----------------------------------|-----------------------------|---|-----------|-----------|
| VERSIÓN | FECHA DE APROBACIÓN | | | DESCRIPCIÓN DEL CAMBIO | | |
| | AAAA | MM | DD | | | |
| 1 | 2018 | 08 | 11 | Emisión del documento | | |
| 2 | 2020 | 10 | 19 | Modificaciones al documento para cumplir con la Resolución 000158 del 29 de octubre de 2019, "POR LA CUAL SE ADOPTA LA IMAGEN CORPORATIVA Y SE DICTAN LAS POLÍTICAS DE IDENTIDAD CORPORATIVA DE LA UNIVERSIDAD DE CUNDINAMARCA" | | |
| ELABORÓ | | | | | | |
| NOMBRES Y APELLIDOS | | | CARGO | | | |
| Angélica María Padilla Narváez | | | Profesional IV | | | |
| Diana Milena Rey Gutiérrez | | | Profesional IV | | | |
| Leidi Caterin Peña García | | | Profesional I | | | |
| REVISÓ | | | | | | |
| NOMBRES Y APELLIDOS | | | CARGO | | | |
| Jorge Alfredo Mayorga Cárdenas | | | Profesional Director Área I | | | |
| APROBÓ (GESTOR RESPONSABLE DEL PROCESO) | | | | | | |
| NOMBRES Y APELLIDOS | | CARGO | | FECHA | | |
| | | | | AAAA | MM | DD |
| Edilson Martínez Clavijo | | Director de Sistemas y Tecnología | | 2020 | 10 | 19 |