Plan de proyecto

Para el trabajo de grado Ane-Stent

Stephanie Domínguez Andrade

@.co

Juan Sebastián Espinosa Torres

[espinosa\_j@javeriana.edu.co](mailto:espinosa_j@javeriana.edu.co)

Jose Antonio Quintero Gómez

@.co

David Alonso Villamizar Lizcano

@.co

J

# Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Descripción** | **Fecha** | **Responsable** |
| 1.0 | Versión inicial del documento | 12/11/2017 | Juan Sebastián Espinosa |

# Prefacio

El presente documento muestra el plan de gestión del trabajo de grado, consistiendo en una guía para el desarrollo del trabajo de grado de la modalidad de investigación “Ane-Stent”. Dicho trabajo consiste en un proyecto que probará la factibilidad aplicación de tecnologías para llevar a cabo una simulación.

El contenido de este documento incluye la descripción de la vista general del proyecto, glosario, contexto, administración, monitoreo y control y Procesos de soporte.

Este documento está dirigido al equipo de trabajo que desarrollará este proyecto, la información y parámetros establecidos acá tienen como fin servir a acuerdos establecidos en la metodología con la que se lleva a cabo el proyecto.

# Tabla de contenidos

[1. Historial de cambios 1](#_Toc498283846)

[2. Prefacio 1](#_Toc498283847)

[3. Tabla de contenidos 1](#_Toc498283848)

[4. Lista de figuras 3](#_Toc498283849)

[5. Lista de tablas 3](#_Toc498283850)

[6. Vista general del proyecto 3](#_Toc498283851)

[6.1. Propósito, alcance y objetivos 3](#_Toc498283852)

[6.1.1. Propósito 3](#_Toc498283853)

[6.1.2. Alcance 4](#_Toc498283854)

[6.1.3. Objetivos 4](#_Toc498283855)

[7. Glosario 4](#_Toc498283856)

[8. Contexto del proyecto 4](#_Toc498283857)

[8.1. Lenguajes y Herramientas 5](#_Toc498283858)

[8.2. Plan de aceptación del producto 6](#_Toc498283859)

[Objetivos: 6](#_Toc498283860)

[Responsables: 7](#_Toc498283861)

[Desarrollo: 7](#_Toc498283862)

[8.3. Organización del proyecto y comunicación 7](#_Toc498283863)

[9. Administración del proyecto 8](#_Toc498283864)

[9.1. Métodos y herramientas de estimación 8](#_Toc498283865)

[9.2. Inicio del proyecto 9](#_Toc498283866)

[9.2.1. Entrenamiento del personal 9](#_Toc498283867)

[9.2.2. Infraestructura 10](#_Toc498283868)

[10. Monitoreo y control del proyecto 12](#_Toc498283869)

[10.1. Cierre del Proyecto 12](#_Toc498283870)

[11. Procesos de soporte 12](#_Toc498283871)

[11.1. Análisis y administración de riesgos 12](#_Toc498283872)

[Responsables 12](#_Toc498283873)

[Recursos: 13](#_Toc498283874)

[Descripción y clasificación 13](#_Toc498283875)

[Probabilidad e impacto 13](#_Toc498283876)

[Análisis de riesgo 13](#_Toc498283877)

[Peores riesgos en términos de probabilidad e impacto 13](#_Toc498283878)

[Plan de respuesta a riesgos 13](#_Toc498283879)

[11.2. Administración de configuración y documentación 14](#_Toc498283880)

[11.2.1. Ítems de configuración 14](#_Toc498283881)

[11.2.2. Lineamientos propios 15](#_Toc498283882)

[11.2.3. Proceso 16](#_Toc498283883)

[11.3. Métricas y proceso de medición 18](#_Toc498283884)

[Plantilla de calidad para documentos: 18](#_Toc498283885)

[Toggl como medición de tiempo 19](#_Toc498283886)

[Plantilla de control de calidad de código 19](#_Toc498283887)

[Métricas de diseño 19](#_Toc498283888)

[Métricas de riesgo 19](#_Toc498283889)

[Métrica de requerimiento 19](#_Toc498283890)

[12. Referencias 20](#_Toc498283891)

# Lista de figuras

[Figura 1. Diseño de carpetas. 18](#_Toc498284616)

# Lista de tablas

[Tabla 1. Interfaces externas. 8](#_Toc498283893)

[Tabla 2. Gestión de componentes de estimación. 9](#_Toc498283894)

[Tabla 3. Convenciones. 10](#_Toc498283895)

[Tabla 4. Especificaciones de los computadores personales 11](#_Toc498283896)

[Tabla 5. Ítems de configuración. 15](#_Toc498283897)

[Tabla 6. Ramas de git. 18](#_Toc498283898)

[Tabla 7. Metricas. 20](#_Toc498283899)

# Vista general del proyecto

## Propósito, alcance y objetivos

### Propósito

Según Courbebaisse et al. “Gracias a avances en tecnología, mejoras en visualización y un mercado creciente para dispositivos mínimamente invasivos, actualmente los practicantes de medicina tienen un conjunto de opciones de tratamiento más amplio para tomar una decisión objetiva adaptada de manera cercana a las condiciones específicas del paciente. Sin embargo, hay muchas preguntas sobre la influencia de estos dispositivos en el proceso de curación de aneurismas que permanecen abiertas, requiriendo un mejor y más profundo estudio de estos métodos, combinando aproximaciones numéricas y virtuales con datos específicos de cada paciente.” (Courbebaisse, 2014)

Se espera que el resultado de este proyecto pueda ser aplicado en los vasos sanguíneos con aneurismas para aumentar el conocimiento en el proceso de tratamiento de aneurismas. Así mismo, se puede entrenar a los médicos en proceso de aprendizaje para la realización de la cirugía de postura de stent.

La herramienta “Bullet Physics” brinda una oportunidad de investigación para la representación de la colocación de un stent en un vaso sanguíneo con aneurisma, aumentar para el conocimiento en esta área de la medicina y la computación gráfica. Así mismo, se puede entrenar a médico en proceso de aprendizaje para la realización de la cirugía.

### Alcance

Se propone investigar acerca de la aplicabilidad de la tecnología de simulación física “Bullet Physics” en el contexto del proceso de postura de un stent en los vasos sanguíneos.

Dicha simulación, debería mostrar la deformación de un segmento de vaso sanguíneo (e.g. venas y arterias) y los cambios del flujo sanguíneos causados por la colocación del stent.

### Objetivos

#### Objetivo general

Evaluar la factibilidad técnica, de una aproximación para implementar una simulación física, en un segmento de vaso sanguíneo. La simulación, inmersa en el contexto de una cirugía de postura de stent; la aproximación, implementada usando “Bullet Physics”.

#### Objetivos específicos

1. Definir una arquitectura basada en “Bullet Physics”.
2. Elaborar una prueba de concepto de una simulación física de un aneurisma con stent usando “Bullet Physics”.
3. Definir un criterio y usarlo para evaluar la factibilidad de la implementación de la prueba de concepto de la simulación, usando métricas calculadas a partir de mediciones obtenidas durante el desarrollo.
4. Elaborar un informe del proyecto en el que se detallan los problemas y soluciones según lo especificado en la plantilla generada por el grupo.

# Glosario

(Ver Anexo 1 Glosario)

# Contexto del proyecto

## Lenguajes y Herramientas

**El proyecto tiene relaciones de dependencia con módulos desarrollados en trabajos de grado anteriores y con Bullet Physics,** los cuales requieren de C++ o algún mecanismo que permita comunicación con este lenguaje. (Pontificia Universidad Javeriana, 2015) (Coumans, 2017)

El prototipo, una vez terminado, probablemente necesite muchos recursos de cómputo (tanto de CPU como de GPU) y de memoria RAM, por tanto, se necesita un lenguaje que pueda ser compilado AOT que permita aplicar optimizaciones al ejecutable final que mejoren el rendimiento.

La complejidad del prototipo que se quiere desarrollar puede llevar a una gran cantidad de errores de programación, por tanto, puede ser útil contar con un compilador que detecte muchos de esos errores sin necesidad de probar en ejecución. Esto sugiere un lenguaje que pueda ser compilado AOT.

Teniendo en cuenta los puntos anteriores, la experiencia del equipo y los sistemas operativos que se tienen como objetivo (GNU Linux Ubuntu 14 LTS, Windows 10 y MacOS Sierra), se considera que **el lenguaje principal que se va a usar para este proyecto es C++,** con el **GCC (GNU Compiler Collection) para compilar.** Si se necesita usar otro lenguaje para completar el prototipo o para alguna otra tarea relacionada con el trabajo de grado, se podrá hacer luego de obtener la aprobación de al menos dos miembros del equipo. (GNU Tools, 2017) (Stroustrup, 2016)

**El prototipo final debe soportar principalmente GNU Linux Ubuntu 14 LTS y 16 LTS, y de ser posible, Windows 10 y MacOS Sierra.** Por tanto, todos los miembros del equipo deben tener una de estas versiones de Linux instalada en uno de sus computadores, y al menos uno de los otros dos. Esta elección se hace teniendo en cuenta la familiaridad de los miembros del equipo con estos sistemas operativos y la facilidad para conseguir dichas versiones de Linux. (Canonical Ltd., 2017) (Microsoft, 2017) (Apple Inc., 2017)

**Para todas las tareas de ofimática, se usará la suite de Google Docs con Google Drive** en documentos compartidos con todo el equipo incluido el director del trabajo de grado. Todos los cambios realizados por miembros del equipo deben realizarse desde su cuenta personal de Google, con el fin de dejar registro claro de las contribuciones de cada miembro. El director de trabajo de grado tendrá permiso de ver, comentar y sugerir cambios en los documentos. Esta elección se hace teniendo en cuenta la experiencia del equipo con la herramienta y las diferentes características que ofrece que facilitan el trabajo en equipo. (Google LLC, 2014) (Google LLC, 2015)

**La administración de requerimientos, la asignación de tareas y el registro de métricas y se hará usando Google Sheets y para el desarrollo de todos los documentos de texto se usará Google Docs.** En caso de requerir entregas en un formato específico se convertirá el documento al formato necesario. **Se usarán los productos de Microsoft Office para convertir a formatos de la misma.** (Google LLC, 2015)(Microsoft Corporation, 2017)

**Para tareas de diseño arquitectónico y detallado de software se usará UML y BPMN con draw.io y Bizagi Modeler.** De necesitar un elemento de diseño o tipo de diagrama que no aún esté disponible en la librería del servicio, se debe discutir una solución con el equipo. Los documentos de diseño deben incluirse en la carpeta de Google Drive del trabajo de grado. Esta elección se hace teniendo en cuenta la experiencia del equipo con UML, y la facilidad para colaborar y la posibilidad de crear nuevos elementos de diseño que ofrece draw.io. (JGraph Ltd, 2015) (Bizagi, 2014)

**Como sistema de control de versiones se usará un repositorio de git en GitHub.** Esta elección se hace teniendo en cuenta que git es el sistema de control de versiones que más conoce el equipo y que GitHub lleva registro de métricas que pueden ser de utilidad para el proyecto más adelante. (GitHub, Inc., 2017) (Software Freedom Conservancy, 2012)

**Para las pruebas unitarias automatizadas se usará el framework Catch2.** Esta elección se hace teniendo en cuenta la facilidad de uso del framework. (Catch Org, 2017)

**Los editores de código que se usarán serán Sublime Text 3 y Atom.** Sin embargo, los miembros del equipo tendrán la libertad de escoger el editor de su preferencia. (GitHub Inc., 2015) (Sublime HQ Pty Ltd, 2017)

**Para el renderizado se planea usar un motor de renderizado que aún está por definirse.** Como propuestas se tienen el de Unreal Engine 4 y el de Unity 2017. (Epic Games, 2017) (Unity Technologies, 2017)

**Se usará el linter de C++ de Google para verificar que todo el código producido cumpla con el Google C++ Style Guide.** (Google LLC, 2017)

**Se usará Doxygen para generar documentación a partir de comentarios en el código fuente.** (Heesch, 2015)

**Se usará CMake y Make para compilar el código con sus dependencias.** (Kitware, 2017)(Feldman, 1979)

## Plan de aceptación del producto

El plan de aceptación del producto debe acordar los criterios de aceptación entre el equipo y el director del proyecto.

### Objetivos:

El objetivo principal de este plan es dar a conocer los criterios de aceptación donde el equipo de trabajo y el director del proyecto considere que se han alcanzado los objetivos de cada iteración del ciclo de vida.

### Responsables:

Todos los miembros del grupo deben ser responsables del cumplimento del plan, para asegurar la calidad de los entregables.

### Desarrollo:

La definición de los documentos entregables se encuentra en la propuesta de grado. Cada iteración de desarrollo corresponde a un hito del modelo de ciclo de vida, las fechas en las que se deben realizar están inmersa en el rango de semestre 2018-1 de la pontificia universidad javeriana. Para los entregables estos deberán ser colocados en la página del grupo ([Sección 9.2.2](#_Infraestructura)).

Para considerar cada entregable como válido para la entrega, este debe pasar por el control de calidad y debe ser aprobado según las plantillas de calidad (Ver documento de calidad) ([Ver sección 11.2](#_Administración_de_configuración)).

## Organización del proyecto y comunicación

Un punto fundamental para la creación de este proyecto es conocer a quién o cómo se va a contactar con personas externas. En la tabla X se pueden ver dichas personas, por qué son importantes para el proyecto, qué rol cumplen y cómo será la comunicación.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Descripción | Responsabilidad | Medio de comunicación |
| Leonardo Florez Valencia | Director de trabajo de grado | Está comprometido con realizar seguimiento y de monitorear semanalmente el progreso del proyecto.  Debe evaluar la propuesta con rúbricas establecidas. (Pontificia Universidad Javeriana, 2016) | Datos de contacto:  [flores-l@javeriana.edu.co](mailto:flores-l@javeriana.edu.co)  3208320 ext 5317  Medios de contacto:  En reuniones semanales de control de avance.  Vía correo electrónico. |
| Guillermo Cristancho | Administrador de servidores del departamento de Ingeniería de sistemas - Pontificia Universidad Javeriana | Es la persona encargada de facilitar un servidor para poder crear y subir la página web requerida para el trabajo de grado. | Datos de contacto:  [g.cristancho@javeriana.edu.co](mailto:g.cristancho@javeriana.edu.co)  Medios de contacto:  Vía correo electrónico |
| Alejandro Sierra | Responsable de asignatura Trabajo de grado. | Encargado de coordinar los trabajos de grado. | Datos de contacto:  [alejandro-sierra@javeriana.edu.co](mailto:alejandro-sierra@javeriana.edu.co)  Medios de contacto:  Vía correo electrónico |
| Pontificia Universidad Javeriana | Entidad proveedora de infraestructura | Para el desarrollo del proyecto, la universidad está abierta para la solicitud de salas y para hacer uso de sus servicios informáticos. (Pontificia Universidad Javeriana, 2012) | Datos de contacto:  3208320  [salasbiblioteca@javeriana.edu.co](mailto:salasbiblioteca@javeriana.edu.co)  Medios de contacto:  Presencial.  Vía correo electrónico. |

Tabla 1. Interfaces externas.

Para la comunicación interna, están las opciones de hacerlo directamente o por medios electrónicos, todos los integrantes del grupo tienen los datos de contacto de los demás, tales como número de celular, correo, entre otros. La estructura organizacional es básicamente una jerarquía horizontal, sin embargo, el liderazgo o la capacidad de toma de decisiones existirá y se cambiará semanalmente, adicionalmente, será el encargado de la gestión de la herramienta trello. ([Sección 9.2.2](#_Infraestructura))

# Administración del proyecto

## Métodos y herramientas de estimación

Las estimaciones son subjetivas y poco exactas (Torres, 2016) tienen como objetivo permitir la organización del grupo con anticipación, cumplimento exitoso y con calidad para las entregas (Chemuturi, 2009)

Se debe realizar diferentes tipos de estimaciones: El tamaño del producto, de esfuerzo y de planificación. En la tabla 3, se observa la gestión de dichos componentes.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Etapa de estimación** | | |
| Tamaño de la simulación | Esfuerzo | Planificación |
| ¿Cómo? | Con la cantidad de requerimientos funcionales y no funcionales. Ya que estos dan una visión de la totalidad de lo que se pretende simular. | Con el esfuerzo requerido por los integrantes del grupo de trabajo (persona-sprint) y con el tamaño de la simulación. | Cuando se tenga las estimaciones de esfuerzo y de tamaño de producto se debe analizar el tiempo que cada requerimiento y subsecuentemente cada tarea de este para cada sprint. |
| ¿Quién? | Todo el grupo de trabajo | Todo el grupo de trabajo | Stephanie Domíguez |
| Herramienta | Excel: Ya que debe tener de manera ordenada, todos los requerimientos para una estimación exitosa. | Excel: Gracias al posible tamaño de la simulación. Esta herramienta contiene los datos ordenados y provee las ecuaciones necesarias para obtener una estimación exitosa | Trello: está herramienta ayuda tener una organización visual sobre los tiempos asignados a cada tarea. |
| ¿Cuándo? | Después de la segunda iteración *“Elaboración”* al realizar el SRS. | Después de la estimación del tamaño de la simulación. | En cada sprint, después de la realización del SRS. |

Tabla 2. Gestión de componentes de estimación.

## Inicio del proyecto

### Entrenamiento del personal

En la tabla 4 se encuentra el entrenamiento de personal que se planea hacer. Donde hay habilidades que ningún miembro del equipo conoce, se planea que unos miembros la aprendan por su cuenta y se la enseñen a los demás.

|  |  |
| --- | --- |
| **Letras** | **Significado** |
| A | Aprende |
| E | Enseña |
| AE | Aprende y Enseña |

Tabla 3. Convenciones.

### Infraestructura

Las herramientas que se pueden llegar a utilizar de acuerdo con las necesidades del proyecto para que este alcance los objetivos serán las siguientes:

#### Computadores

Los computadores son una necesidad básica para el proyecto y serán necesarios en todo momento. Cada integrante del grupo realizará sus labores en sus computadores personales con las siguientes especificaciones:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Proprietario** | **Tipo** | **Modelo** | **Procesador** | **Tipo de sistema** | **Frecuencia** | **RAM** | **Sistemas operativos** | **Tarjeta gráfica** |
| Stephanie Dominguez | Principal | HP envy | Intel Core i7 | 64 bits | 2,00 GHz | 12 GB | Windows 10 y Ubuntu 16.04 LTS | NVIDIA GEForce GTX 850M |
| Juan Sebastian Espinosa | Principal | ASUS X55B | AMD A9-9410 RADEON R5, 5 COMPUTE CORES 2C+3G | 64 bits | 2,00 GHz | 8 GB | Windows 10 y Ubuntu 14.04 LTS | AMD Radeon(TM) R5 Graphics |
| José Antonio Quintero | Principal | Lenovo W510 | Intel Core i7 | 64 bits | 2,00 GHz | 16 GB | Windows 10 y Ubuntu 16.04 LTS | NVIDIA Quadro FX 880M |
| David Alonso Villamizar | Principal | MacBook Pro | Intel Core i7 | 64 bits | 2,50 GHz | 16 GB | OSX Sierra, Windows 10 y Ubuntu 16.04 LTS | NVIDIA GeForce GT 750M 2 GB e Intel Iris Pro 1536 MB |

Tabla 4. Especificaciones de los computadores personales

En caso de que un computador personal presente fallos, robo o cualquier inconveniente, es deber de cada persona informar al equipo de trabajo y reparar o reponer lo antes posible.

#### Servidor:

La Pontificia universidad Javeriana, brindara el espacio para un servidor en pegasus (<https://pegasus.javeriana.edu.co>), donde se debe crear y diseñar una página web, que contenga todos los entregables del proyecto final como el SRS y la propuesta de grado. El encargado del servidor es el administrador de servidores Guillermo Alejandro Cristancho Delgado ([Sección 8.1](#_Lenguajes_y_Herramientas)).

#### Espacios físicos

El grupo de trabajo debe contar con espacios para realizar las reuniones de final del sprint o de desarrollo de las actividades. Estos lugares deben contar con acceso a internet y fuentes de energía. Dentro de la universidad se puede separar el lugar donde se va a trabajarse debe reservar una sala de la biblioteca o de la universidad por medio del correo electrónico, donde se debe especificar la hora de entrada y salida, el número de personas y la sala a solicitar. También se puede hacer uso de sitios fuera de la universidad como la casa de alguno de los integrantes del grupo.

#### Programas

Los programas como Excel y Word, estos pueden ser instalados por Microsoft Office 365 (Microsoft Corporation, 2017), el cual se encuentra de manera gratuita gracias a la universidad y deben ser utilizados por los miembros del equipo.

Bizagi modeler es software libre y se puede descargar desde la página oficial (Bizagi, 2014), debe ser usado para modelar los procesos que el equipo considere.

Atom.io se utilizará como editor de texto, es software libre y puede ser descargado en la página oficial (GitHub Inc., 2015), este es el recomendado para ser usado, los integrantes del equipo pueden elegir el que prefiera.

Bullet Physics, es componente esencial para el desarrollo del proyecto por lo que todos los integrantes del grupo deben tener la misma versión e instalarlo, las instrucciones para esto se encuentran en la documentación de la librería. (Coumans, 2017)

([Sección 8.1](#_Lenguajes_y_Herramientas))

#### Herramientas online

Las herramientas como Google Drive (Google LLC, 2014), GitHub (GitHub, Inc., 2017), requieren de conexión a internet y una cuenta. En todas se puede obtener una cuenta gratuita. Trello debe ser manejado por la persona que lidere esa semana ([Sección 8.3](#_Organización_del_proyecto)).

# Monitoreo y control del proyecto

## Cierre del Proyecto

A continuación, se definirá las acciones que se deben realizar al acabar cada iteración del proyecto.

1. Revisión de calidad: Se hará una revisión de cada punto realizado por los integrantes, asignando un evaluador por punto. Dicho evaluador debe ser distinto al que realizó la sesión, por medio de la rúbrica realizada por el equipo, se calificará de 1 a 3 con base en las rúbricas realizadas por el grupo, se harán los comentarios pertinentes para la mejora de dicho punto.
   1. Revisión del documento: Se realizará una revisión en cuestión a los enlaces creados a carpetas con anexo del proyecto. Que este presentadas de manera clara y sean enumeradas en una tabla de ilustraciones, validar que las referencias tengan concordancia con la sección y no sean de páginas no permitidas o blogs de empresas no oficiales.
   2. Análisis de métrica del tiempo: En esta sección se realiza un análisis del tiempo utilizado versus el estimado para la iteración por medio de la herramienta toogle ([Sección 11.3](#_Métricas_y_proceso)).
   3. Análisis del diseño: En esta sección se realizará un análisis de la evaluación temprana ([Sección 11.3](#_Métricas_y_proceso)).
2. Análisis de resultados: Se debe escribir como se trabajó durante la realización del documento, la organización que se ha venido manejando a lo largo de la iteración, hacer un plan de organización para la siguiente, explicar que se ha aprendido durante el desarrollo y otorgar retroalimentación a los integrantes del grupo. Para mejorar el desempeño en cada iteración.

# Procesos de soporte

## Análisis y administración de riesgos

El plan para la gestión de riesgos consiste en identificar, priorizar y crear una respuesta a los mismos con el fin de mitigarlos o reducir sus consecuencias.

### Responsables

* El líder correspondiente a la semana debe estar al tanto de lo que ocurre en el grupo y así actualizar y tomar las medidas que considere pertinentes.
* Del lado de los integrantes del grupo que no están ejerciendo el rol de líder, deben procurar estar pendientes de los cambios, hacer caso a lo que se estipule en el plan e igualmente, reportar en caso de que no se aplique.

### Recursos:

Microsoft excel es la herramienta que se usará para la gestión de riesgos para la identificación, análisis, priorización y planteamiento de planes de mitigación.

Para esto se tiene una plantilla la cual especifica:

### Descripción y clasificación

* + Número del riesgo: identificador.
  + Riesgo: Nombre del riego.
  + Descripción: Se amplía en el contexto en el que ocurre el riesgo.

### Probabilidad e impacto

* + Probabilidad de que ocurra: descripción cualitativa (Insignificante, baja, media, alta).
  + Impacto: Qué tanto puede afectar al producto o desarrollo si el riego.      (baja, media, alta)

### Análisis de riesgo

* + Probabilidad de que ocurra: descripción cualitativa (Insignificante, baja, media, alta).
  + Impacto: Qué tanto puede afectar al producto o desarrollo si el riego. (baja, media, alta)
  + Resultado**:** Calculo final del riego.

### Peores riesgos en términos de probabilidad e impacto

* + Se describen únicamente los riesgos más urgentes, como se puede ver en la plantilla de riesgos.

### Plan de respuesta a riesgos

1. Decisión equivocada.

Transferencia a terceros: Se realizará consultoría para las decisiones que se consideren importantes.

1. Falla de servicios de terceros

Plan de contingencia: Se buscan servicios similares o iguales.

1. Requerimientos subestimados

Plan de contingencia: Los integrantes tienen la oportunidad de reportar que los requerimientos asignados no son como se esperaban.

1. Supuestos del proyecto incorrectos

Aceptación: Se tendrá que adaptar a los cambios que esto provoque.

1. Falta de disponibilidad de personas externas

Plan de contingencia: Buscar personas que puedan suplir la función, al menos temporalmente.

1. Mala elección de tecnologías

Mitigar: Se investigará antes de elegir.

1. Fallas en servicio de transporte

Plan de contingencia: Se buscan más métodos de transporte o se intenta hacer reunión virtual.

(Anexo 2 plantilla riesgos)

(Anexo 3 BPMN riesgos)

## Administración de configuración y documentación

### Ítems de configuración

Los ítems de configuración identificados para el proyecto son, además de los entregables, otros productos intermedios en nuestro proceso de desarrollo y pueden verse en la tabla 5. Cuando se identifiquen más de estos durante el desarrollo del proyecto, podrán agregarse aquí.

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Ítem** | **Estándar** |
| Documento | SPMP | 16326-2009 - ISO/IEC/IEEE Systems and Software Engineering--Life Cycle Processes--Project Management  Software Metrics and Software Metrology |
| SRS | 1016-2009 - IEEE Standard for Information Technology--Systems Design--Software Design Descriptions  830-1998 - IEEE Recommended Practice for Software Requirements Specifications |
| SDD | 29119-5-2016 - ISO/IEC/IEEE International Standard - Software and systems engineering -- Software testing. Partes 1-5 |
| Documentación de pruebas | 29119-5-2016 - ISO/IEC/IEEE International Standard - Software and systems engineering -- Software testing. Partes 1-5 |
| Postmortem del proyecto | A defined process for project postmortem review |
| Manual de uso, instalación y configuración. | 26514-2010 - IEEE Standard for Adoption of ISO/IEC 26514:2008 Systems and Software Engineering--Requirements for Designers and Developers of User Documentation |
| Memoria de trabajo de grado |  |
| Código fuente | Código fuente | Google Style Guide |
| Página web de trabajo de grado |  |
| Recursos multimedia del prototipo |  |

Tabla 5. Ítems de configuración.

### Lineamientos propios

Los lineamientos definidos en esta sección deben seguirse para todos los productos entregables o intermedios. Los mecanismos que se seguirán para su verificación se especifican en el plan de control de calidad del proyecto (Ver plan de control de calidad).

#### Documentos

* Todos los tipos de texto deben estar en color negro alineado hacia la izquierda sin justificar.
* Texto del cuerpo Arial de 12 pt.
* Texto del encabezado nivel 1 con formato Arial de 20 pt y centrado.
* Texto del encabezado nivel 2 con formato Arial de 18 pt.
* Texto del encabezado nivel 3 con formato Arial de 16 pt.
* Texto del encabezado nivel 4 con formato Arial de 14 pt.
* Todas las viñetas deben ser redondas rellenas negras.
* Todo el documento debe tener activada la opción “Añadir una línea después de cada párrafo”.
* Tablas con todos los bordes.
* Texto en celdas de las tablas con formato Arial de 12 pt, alineado hacia arriba y hacia la izquierda.
* Texto en celdas de encabezado de las tablas en negrilla, y centrado horizontal y verticalmente.
* Las tablas y figuras deben tener un número único asignado en el orden en que primero aparecen en el documento (los números diferentes para tablas y figuras).
* Las tablas y figuras deben tener pie de figura en formato “Figura <número\_figura>. <descripción\_figura>”, alineado hacia la izquierda.

#### Código fuente

* Todo el código debe cumplir con los lineamientos definidos en el Google Style Guide. Para ello se usarán linters que verifiquen estos lineamientos. (Google LLC, 2017)
* Todo el código debe ser probado, como se especifica en el plan de pruebas del proyecto (Ver plan de pruebas).
* Todas las clases, constantes, funciones y métodos deben tener comentarios que permitan generar documentación automática con Doxygen, de manera que se expliquen retornos, parámetros y forma de uso de cada función y método. (Heesch, 2015)

### Proceso

El proceso de administración de configuración depende del tipo del ítem de configuración que se esté desarrollando.

#### Documentos

Los documentos se trabajarán en una carpeta en Google Drive compartida con todos los miembros del equipo.

La figura 1 describe la jerarquía con la que se dará desarrollo de cada entregable y artefacto de desarrollo de tipo documento.

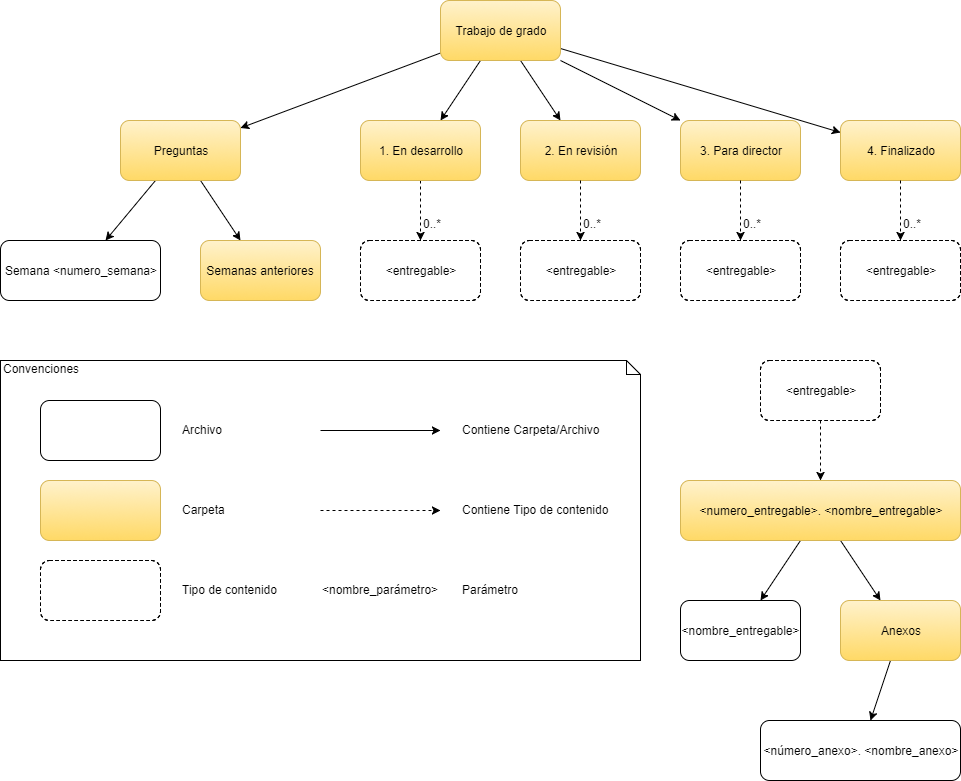


Figura 1. Diseño de carpetas.

La jerarquía describe 5 directorios principales descritos de la siguiente manera:

* **Preguntas:** En este directorio se consolidan las dudas planteadas por el equipo de desarrollo en cada semana, al final de la semana, el documento con las preguntas, así como sus respuestas, se archiva en el subdirectorio de Semanas anteriores para futura consulta.
* **1. En Desarrollo:** En este directorio, se encontrará cada entregable de tipo documento que se encuentra en desarrollo, es decir, que se encuentra siendo editado.
* **2. En Revisión:** En este directorio, se encontrará cada entregable de tipo documento que está en espera de ser evaluado en el proceso de aseguramiento de calidad.
* **3. Para director:** En este directorio, se encontrará cada entregable de tipo documento que cumple con los lineamientos de calidad establecidos por el equipo de trabajo y se encuentra a la espera para ser revisado por el director de trabajo de grado.
* **4. Finalizado:** En este directorio, se encuentra la versión final de cada entregable de tipo documento. A partir de este directorio, se realiza la entrega final.

El proceso a seguir está descrito en el modelo de proceso de negocio anexado.

(Anexo 4 Proceso de configuración - Documentos)

#### Código fuente

El código fuente se mantendrá en un repositorio de git privado en GitHub. Se tendrán las ramas descritas por la tabla 6:

|  |  |
| --- | --- |
| **Nombre rama** | **Descripción** |
| master | Es la rama principal. Contiene todo el código fuente probado y funcional. |
| testing | Es la rama de pruebas. Contiene todo el código cuya funcionalidad se considera completa, pasó las pruebas unitarias automatizadas y que espera a ser probado para unirse al master. |
| dev | Es la rama de pruebas. Contiene todo el código cuya funcionalidad aún no está siendo completada. |

Tabla 6. Ramas de git.

El proceso a seguir está descrito en el modelo de proceso de negocio anexado.

(Anexo 5 Proceso de configuración - Codigo)

## Métricas y proceso de medición

El propósito de esta sección es informar cómo se mide la calidad de los entregables, el tiempo invertido en cada actividad y el control de diseño, código y riesgos (Ver propuesta de proyecto). Todas las plantillas tienen una métrica y sus resultados dan un resultado cuantitativo.

### Plantilla de calidad para documentos:

Esta métrica fue creada y adaptada del curso de Ingeniería de software. Se toma como criterio de calidad del documento la ausencia de errores de ortografía y redacción, estos deben ser coherentes, sin contradicciones ni ambigüedades. Las plantillas con las correspondientes métricas serán utilizadas por cada sección del documento.

(Anexo 6 Plantilla Calidad Documentos)

### Toggl como medición de tiempo

La métrica de tiempo que se usará es el tiempo planeado para realizar todas las actividades versus el tiempo que se necesitó realizar las actividades. Para ello es importante aclarar que este maneja cuanto se demora la persona en hacer una tarea, esta herramienta permite medir el tiempo que se emplea para cada tarea de manera individual.

### Plantilla de control de calidad de código

Se usará la herramienta GitHub para el versionamiento de código a realizar los sprint. La métrica establecida por el equipo evaluará los errores de código, sintaxis y lógica, se medirá el número de líneas de código y complejidad del desarrollo. La plantilla utilizada para este caso es control de calidad de código explicado en el plan de control de calidad, esta fue elegida porque explica de manera eficiente y concisa el código realizado por el equipo.

(Ver anexo 7 plantilla de calidad control de código)

### Métricas de diseño

Para medir el diseño de alto y bajo nivel, se hará una evaluación temprana (antes de implementar y no necesariamente después de acabar toda la arquitectura) por cada hito de diseño.

Para la evaluación se tendrá en cuenta una calificación sobre 3 puntos. Uno se obtendrá en caso de que el diseño propuesto cumpla con los objetivos de calidad propuestos, otro punto se obtendrá en caso de que el diseño planteado efectivamente pueda llevarse a cabo con los recursos que se poseen y el otro si el diseño está teóricamente bien. (Dávila, Germán, Crutas, & García, 2006)

### Métricas de riesgo

Para asegurar el seguimiento de los riesgos, realizados en la sección de riesgos ([Ver sección 11.1.](#_Análisis_y_administración)) se usa la siguiente métrica: cuantas veces se presentó el riesgo durante las iteraciones y cuantas se pasaron por alto. Se tiene un formato para poder realizar el proceso de medición, la plantilla contiene los siguientes datos: ID del riesgo, veces presentada en la iteración, respuesta al riesgo y persona encargada de la respuesta al riesgo.

(Anexo 2 plantilla riesgos)

### Métrica de requerimiento

Las métricas serán desarrolladas en la sección administrado, que será realizada en el SRS.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Análisis** | **Métrica** | | | | |
| **Documento** | **Código** | **Tiempo** | **Diseño** | **Riesgos** |
| **¿Quién?** | Juan Sebastian Espinosa | Stephanie Dominguez | El lider de la semana | David Villamizar | Jose Antonio Quintero |
| **¿Cómo?** | Con la plantilla de calidad propuesta | Con la plantilla de calidad propuesta | La herramienta Toggl | Con la plantilla de calidad propuesta | Con la plantilla de calidad propuesta |
| **¿Cuándo?** | Al comienzo de cada sprint | Al final del sprint | Al comienzo de cada sprint | Al comienzo de cada sprint | Al final de cada sprint |

Tabla 7. Metricas.

# Referencias

Apple Inc. (25 de Septiembre de 2017). *Upgrade to macOS High Sierra*. Obtenido de Official Apple Support: https://support.apple.com/macos/high-sierra

Bizagi. (6 de Agosto de 2014). *Bizagi BPM Modeler - BPMN Software - Free Download*. Obtenido de Bizagi Time to Digital: https://www.bizagi.com/en/products/bpm-suite/modeler

Canonical Ltd. (8 de Junio de 2017). *The leading operating system for PCs, IoT devices, servers and the cloud | Ubuntu*. Obtenido de ubuntu: https://www.ubuntu.com/

Catch Org. (3 de Noviembre de 2017). *Catch2*. Obtenido de GitHub: https://github.com/catchorg/Catch2

Chemuturi, M. (2009). *Software Estimation Best Practices, Tools & Techniques.* J. Ross.

Coumans, E. (9 de Noviembre de 2017). *Bullet Physics SDK*. Obtenido de GitHub: https://github.com/bulletphysics/bullet3

Courbebaisse, G. (25 de Junio de 2014). *A new model for the construction of virtual fully resolved flow-diverters | Thrombus-VPH*. Obtenido de Thrombus-VPH: http://thrombus-vph.eu/a-new-model-for-the-construction-of-virtual-fully-resolved-flow-diverters

Dávila, M., Germán, M., Crutas, D., & García, A. (9 de Junio de 2006). *Evaluación de Arquitecturas de Software.* Obtenido de Facultad de Ingeniería - Universidad de la República - Uruguay: https://www.fing.edu.uy/inco/cursos/gestsoft/Presentaciones/Evaluacion%20de%20Arquitecturas%20-%20G10/Evaluacion%20de%20Arquitecturas.doc

Epic Games. (23 de Febrero de 2017). *Game Engine Technology by Unreal*. Obtenido de Game Engine Technology by Unreal: https://www.unrealengine.com/en-US/what-is-unreal-engine-4

Feldman, S. I. (1979). *Make --- A Program for Maintaining Computer Programs.* Obtenido de CiteSeerX: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=41166288016E83F6B2B830F701B7A755?doi=10.1.1.39.7058&rep=rep1&type=pdf

GitHub Inc. (25 de Junio de 2015). *Atom*. Obtenido de Atom: https://atom.io/

GitHub, Inc. (1 de Marzo de 2017). *GitHub*. Obtenido de GitHub: https://github.com/

GNU Tools. (28 de Febrero de 2017). *GCC, the GNU Compiler Collection*. Obtenido de GCC, the GNU Compiler Collection: https://gcc.gnu.org/

Google LLC. (17 de Marzo de 2014). *Google Drive: almacenamiento en la nube, copias de seguridad de fotos, documentos y mucho más*. Obtenido de Goole Drive: https://www.google.com/drive/

Google LLC. (2 de Septiembre de 2015). *Documentos de Google: crea y edita documentos online de forma gratuita.* Obtenido de Google Docs: https://www.google.com/docs/about/

Google LLC. (10 de Noviembre de 2017). *google/styleguide: Style guides for Google-originated open-source projects*. Obtenido de GitHub: https://github.com/google/styleguide

Heesch, D. (3 de Julio de 2015). *Doxygen Manual*. Obtenido de Doxygen Manual: Getting started: https://www.stack.nl/~dimitri/doxygen/manual/starting.html

JGraph Ltd. (26 de Noviemebre de 2015). *draw.io*. Obtenido de draw.io: https://www.draw.io/

Kitware. (10 de November de 2017). *CMake*. Obtenido de Cmake: https://cmake.org/

Microsoft. (17 de Octubre de 2017). *Windows 10 keeps getting better*. Obtenido de Microsoft Corporation: https://www.microsoft.com/en-us/windows/features

Microsoft Corporation. (17 de Octubre de 2017). *Microsoft Office | Productivity Tools for Home & Office*. Obtenido de Microsoft Office: https://products.office.com/en/home

Pontificia Universidad Javeriana. (1 de Noviembre de 2012). *ACUERDO N° 567 (Actualización del Reglamento de Estudiantes)*. Obtenido de Pontificia Universidad Javeriana: http://www.javeriana.edu.co/documents/10179/48161/REglamentoEstudiantes.pdf/81d51d42-7361-4735-b8e6-fd3789d62912

Pontificia Universidad Javeriana. (30 de Enero de 2015). *Trabajos de grado*. Obtenido de Trabajos de grado. Pontificia Universidad Javeriana: http://pegasus.javeriana.edu.co/portal

Pontificia Universidad Javeriana. (4 de Diciembre de 2016). *Trabajo de Grado*. Obtenido de Pontificia Universidad Javeriana: http://ingenieria.javeriana.edu.co/estudiantes/pregrados/ingenieria-sistemas/trabajo-grado

Software Freedom Conservancy. (4 de Mayo de 2012). *Git*. Obtenido de Git: https://git-scm.com/

Stroustrup, B. (21 de Diciembre de 2016). *The C++ Programming Language*. Obtenido de The C++ Programming Language: http://www.stroustrup.com/C++.html

Sublime HQ Pty Ltd. (13 de Septiembre de 2017). *Sublime Text - A sophisticated text editor for code, markup and prose*. Obtenido de Sublime Text: https://www.sublimetext.com/

Torres, M. E. (2016). Notas de clase ingeniería de software. Bogotá: Pontifica Universidad Javerian.

Unity Technologies. (27 de Junio de 2017). *Unity - Game Engine*. Obtenido de Unity - Game Engine: https://unity3d.com/es/