

Diagrama de flujo sobre el proceso de desarrollo



C0.3 Reto en clase

Proceso de desarrollo y flujo de trabajo de la metodología a seguir

Instrucciones


- Basado en la información proporcionada, los videos observados y si es necesario realice una breve investigación sobre **el proceso de desarrollo y flujo de trabajo de la metodología a seguir**.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo **Enlace a mi GitHub**
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo .md se debe exportar un archivo .pdf con la nomenclatura **C0.3_NombredelaActividad_NombreAlumno.pdf**, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma **oficial** aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

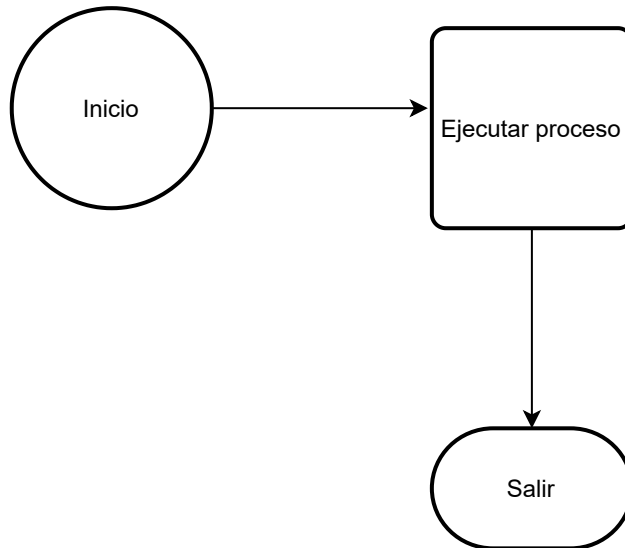
```
| readme.md
| | blog
| | | Cx.1_NombredelaActividad.md
| | | Ax.1_NombredelaActividad.md
| | diagrams
| | docs
| | html
| | img
| | pdf
```



Desarrollo

1. Instale la extensión Draw.io integration dentro de visual studio code y elabore el diagrama solicitado, bajo la metodología que se está planteando

-  [Extension de visual studio code - Draw.io integration](#)
- Ejemplo de un diagrama de flujo utilizando Draw.io:



2. Elabore el diagrama de flujo que contenga la secuencia de bloques que se tendría que seguir durante un desarrollo de software, bajo esta metodología. y explique que actividades que se involucrarían en cada parte del flujo.

-Modelado de negocios

Definir objetivos: se definen objetivo general y específicos del sistema a desarrollar.

Definir riesgos: se definen riesgos o posibles limitaciones al desarrollar el sistema.

Encontrar actores y casos de uso: identificar quien interactua con el sistema y como interactua con el.

-Requerimientos

Definir requerimientos funcional y no funcionales del sistema.

Priorizar casos de uso: determinar cuales son los casos de uso más importantes.

Detallar casos de uso: describir el flujo del caso de uso detalladamente.

-Análisis y diseño

Análisis de la arquitectura: esbozar el modelo de análisis y la arquitectura.

Analizar un caso de uso: Capturar requisitos especiales sobre la realización del caso de uso.

Analizar una clase: Identificar atributos y relaciones de la clase. Diseño de arquitectura: esbozar los modelos de diseño y despliegue

Diseño de clase: definir operaciones, atributos, relaciones, métodos y estados.

Diseño de caso de uso: identificar clases necesarias para llevar a cabo el caso de uso.

Diseño de subsistema: diseño de un subsistema que sea independiente de los demás.

Implementación:

Implementación de la arquitectura.

Integrar el sistema: integrar los componentes del sistema.

Implementar un subsistema: al implementar se verifica que el subsistema cumpla su papel.

Implementar una clase: generar código de la clase e implementar operaciones de la clase como métodos.

-Prueba

Planificar prueba: describir una estrategia para la prueba.

Diseñar la prueba: identificar y estructurar los procedimientos de prueba.

Implementar la prueba.

Realizar pruebas de integración: realizar las pruebas de integración relevantes ejecutando los procedimientos o componentes de prueba correspondientes

Realizar pruebas de sistema.

Evaluar la prueba: se comparan resultados de la prueba con resultados esperados mediante el uso de métricas.

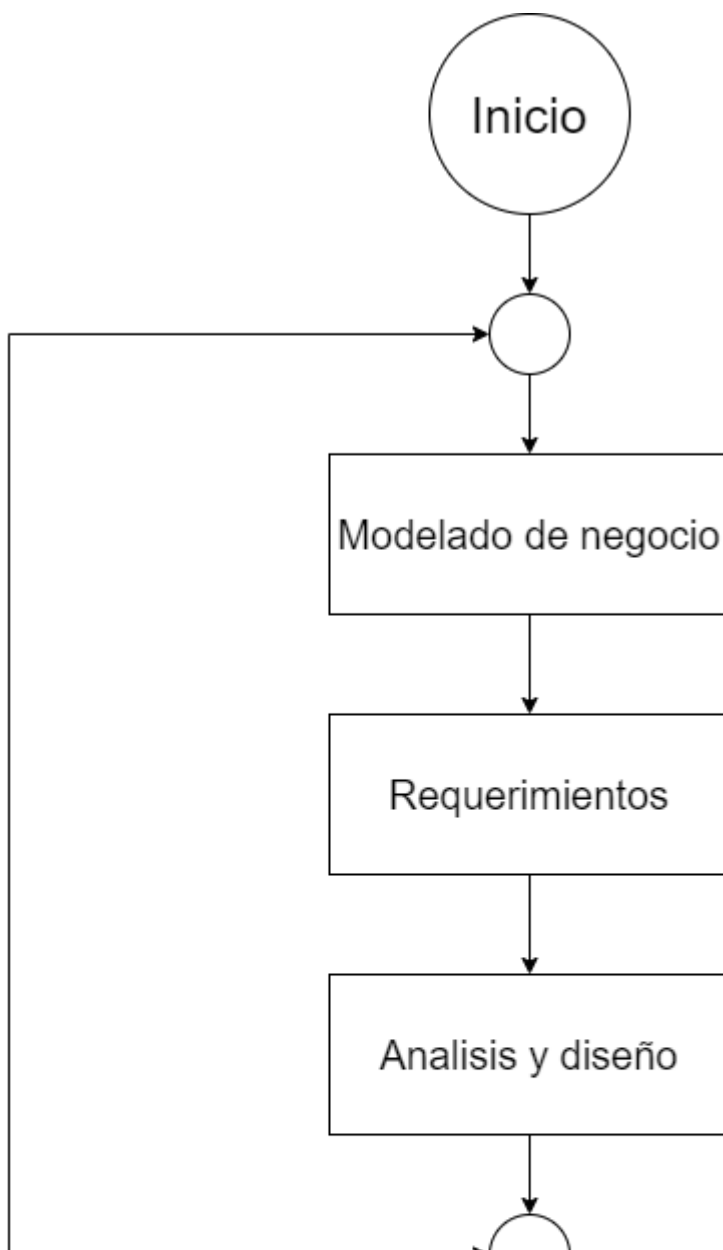
Despliegue

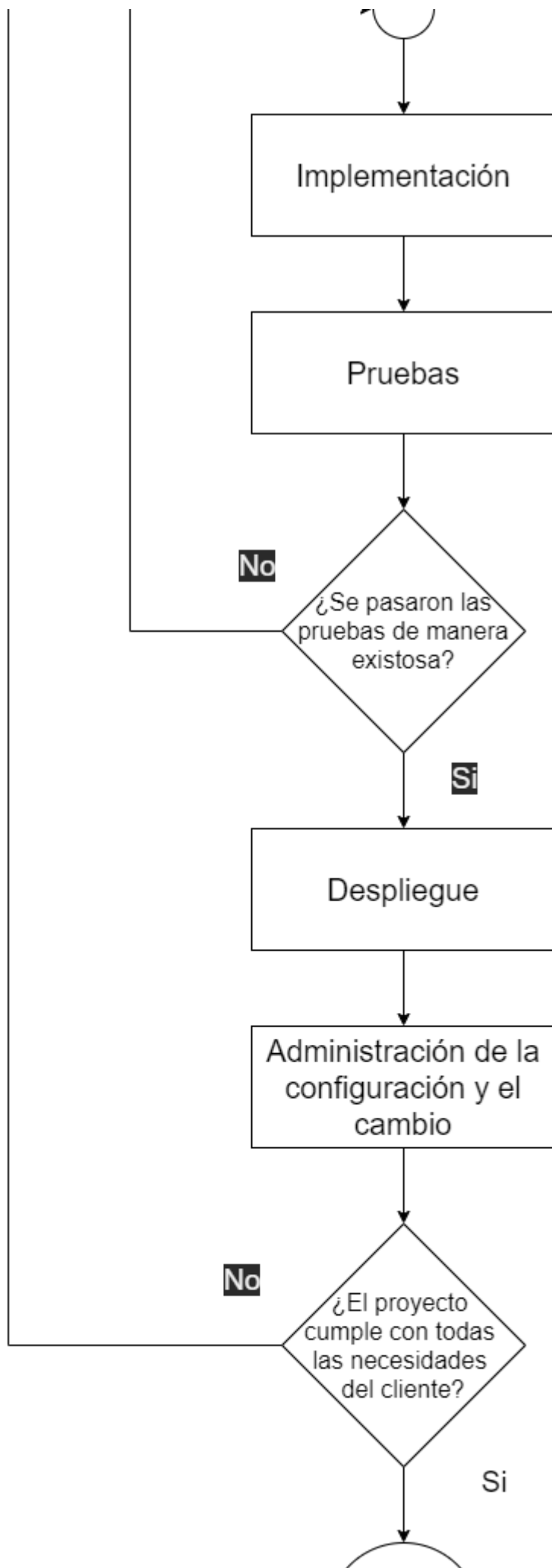
Distribución e instalación del software.

Monitoreo de rendimiento.

-Administración de la configuración y el cambio

Gestión de los cambios del sistema.







Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80



[Mi repositorio de Github](#)