Forma

Descripción generada automáticamente con confianza media  

**Universidad Autónoma de Yucatán**

**Facultad de Matemáticas**

**Licenciatura en Ingeniería de Software**

**Aseguramiento de la calidad de Software**

**Proyecto:** Primer avance

**Integrantes:**

Kuk Baeza, Carlos Adrián

Ruiz Blanco, Joel Iván

Us Cachón, Dilian Armando

García Cetina, Daniel

**Fecha de entrega:** 31/01/2024

Índice

Contenido

[Contenido 2](#_Toc1856355987)

[Introducción 3](#_Toc630824228)

[Organigrama funcional 3](#_Toc2047244707)

[Estructura de procesos 4](#_Toc1845143555)

[Simbología utilizada en los diagramas de flujo 5](#_Toc1680920807)

[Metodología 6](#_Toc138147311)

[PROCEDIMIENTO: Requerimientos 8](#_Toc1157461289)

[OBJETIVO 9](#_Toc1316277357)

[POLÍTICAS DE OPERACIÓN 9](#_Toc1539994733)

[DIAGRAMA DE FLUJO DE PROCESO 9](#_Toc1384507409)

[PROCEDIMIENTO: Diseño de datos 10](#_Toc1511512669)

[OBJETIVO 11](#_Toc1839731885)

[POLÍTICAS DE OPERACIÓN 11](#_Toc1572678591)

[DIAGRAMA DE FLUJO DE PROCESO 11](#_Toc907181000)

[PROCEDIMIENTO: Diseño de procesos 12](#_Toc839319560)

[OBJETIVO 13](#_Toc1982312436)

[POLÍTICAS DE OPERACIÓN 13](#_Toc676541569)

[PROCEDIMIENTO: Codificación del proyecto 13](#_Toc1708556992)

[OBJETIVO 14](#_Toc1428914047)

[POLÍTICAS DE OPERACIÓN 14](#_Toc1418749689)

[DIAGRAMA DE FLUJO DE PROCESO 14](#_Toc1912551505)

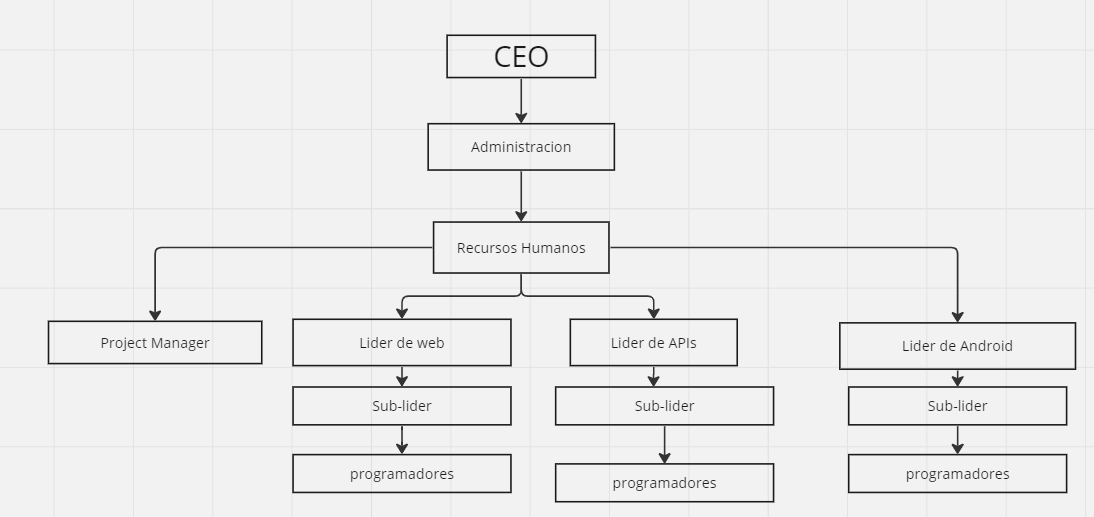
# Contenido

## Introducción

Este manual de procedimientos es la estructura organizacional de la organización “IGR Tec”, el contenido indica los procesos, actividades e interacciones que conforman el desarrollo del producto.

El manual tiene como objetivo, identificar los roles designados de cada departamento y como cada uno realiza las actividades respectivas de su propio departamento. El proceso incluye un nombre, su objetivo, su descripción narrativa y un diagrama de flujo que demuestre gráficamente el comportamiento del departamento.

## Organigrama funcional

**CEO**: En esta organización, se encarga de la conexión con el cliente, obtención inicial de requisitos y funciones administrativas.

**RH:** Tramites del personal, contrataciones, evaluaciones, retributivos y promoción interna en la organización.

**Project Manager:** Supervisión del avance de los proyectos de los diferentes departamentos de desarrollo, reportes de avances y contratiempos, gestión de tiempos de entrega, reuniones de equipos e informes a los clientes.

**Líder de departamento de desarrollo (Web, API, Android):** Encargado de su área asignada. Revisiones de código, funcionalidades y artefactos producidos en su departamento. Gestión de proyectos enfocado en el departamento y participación activa en la construcción del software. Realizar informes sobre el estado de los proyectos, contacto cercano con los lideres de los otros departamentos y el Project manager.

**Sublíder:** Apoyar al líder del equipo en sus actividades. Participación activa en la construcción del software. Capacitación de personal de nuevo ingreso. Revisiones de artefactos producidos en el área, correcciones de funcionalidades, gestión de los repositorios. En caso de ausencia del líder, toma la responsabilidad.

**Programadores:** Creación de artefactos de software, testing, mantenimiento, informes de errores y contratiempos.

## Simbología utilizada en los diagramas de flujo

|  |  |  |
| --- | --- | --- |
| FIGURA | NOMBRE | DESCRIPCIÓN |
|  | Inicio / Fin | Este símbolo representa la manera gráfica con la que se inicia la descripción de un procedimiento. |
|  | Actividad | Se utiliza cuando la unidad orgánica que participa en el procedimiento realiza una actividad o acción. |
|  | Decisión | Se usa cuando existe una toma de decisión y deba seguirse una u otra alternativa, pudiendo tener dos o más opciones de respuesta. |
|  | Documento | Se usa para referirse a cualquier documento que ingrese o se genere dentro del procedimiento e incluso que salga de él (nombre, clave del formato, informe, etc.) |
|  | Procedimiento | Se emplea cuando dentro del procedimiento que se está describiendo exista o se inicie otro procedimiento |
|  | Conector de actividad | Esta imagen se usa para dar continuidad entre una actividad y otra, enviando con ello el cruce de líneas con punta de flecha, su orden deberá ser numérica y progresivo. |
|  | Conector de página | Esta figura se usa para explicar que existe continuidad entre una página y otra, su orden deberá ser numérico y progresivo. |
|  | Sistema | Se emplea para el nombre de una aplicación y describir la actividad desarrollada dentro de la misma. |
|  | Líneas conectoras | Simbolizan la unión entre las diferentes figuras para dar continuidad y dirección al flujo de actividades. No deben cruzarse entre sí. |

## Metodología

La metodología aplicada para la entrega y la revisión de los procesos es SCRUM. Es un marco de trabajo para el desarrollo ágil de software. Este proceso se divide en varios roles, eventos y artefactos.

**Roles:**

**Product Owner:**

* Representa los intereses del cliente y las partes interesadas.
* Define y prioriza las historias de usuario en el Product Backlog

**Scrum Master:**

* Facilita el equipo SCRUM y elimina los obstáculos.
* Ayuda a aplicar y seguir los principios y prácticas de SCRUM.

**Equipo de desarrollo:**

* Grupo autoorganizado de profesionales que entregan el incremento del producto al final de cada Sprint.

**Eventos en SCRUM:**

**Sprint:**

Un periodo de tiempo fijo (generalmente 2-4 semanas) en el que se realiza trabajo potencialmente entregable.

**Reunión de Planificación del Sprint:**

El equipo selecciona elementos del Product Backlog y decide cómo los implementará durante el Sprint.

**Revisión del Sprint:**

Al final de cada Sprint, el equipo presenta el trabajo realizado al Product Owner y a las partes interesadas.

**Retrospectiva del Sprint:**

El equipo revisa el propio desempeño y busca maneras de mejorar en futuros Sprints.

**Daily Scrum (Scrum Diario):**

Una breve reunión diaria (15 minutos) donde el equipo revisa el progreso y ajusta el plan para alcanzar el objetivo del Sprint.

**Artefactos en SCRUM:**

**Product Backlog:**

Una lista priorizada de todas las características, historias de usuario y tareas pendientes del proyecto.

**Sprint Backlog:**

Un conjunto de elementos del Product Backlog seleccionados para el Sprint y un plan para entregarlos.

**Incremento del Producto:**

El producto funcional al final de cada Sprint, que debe ser potencialmente entregable.

# **PROCEDIMIENTO: Requerimientos**

Se recaba información a través de múltiples reuniones con el cliente para definir el sistema a desarrollar y todas las características que se esperan encontrar en el sistema.

La información obtenida en las reuniones debe ser utilizada en la definición de los requisitos para obtener un panorama sobre el proyecto y sus implicaciones

## **OBJETIVO**

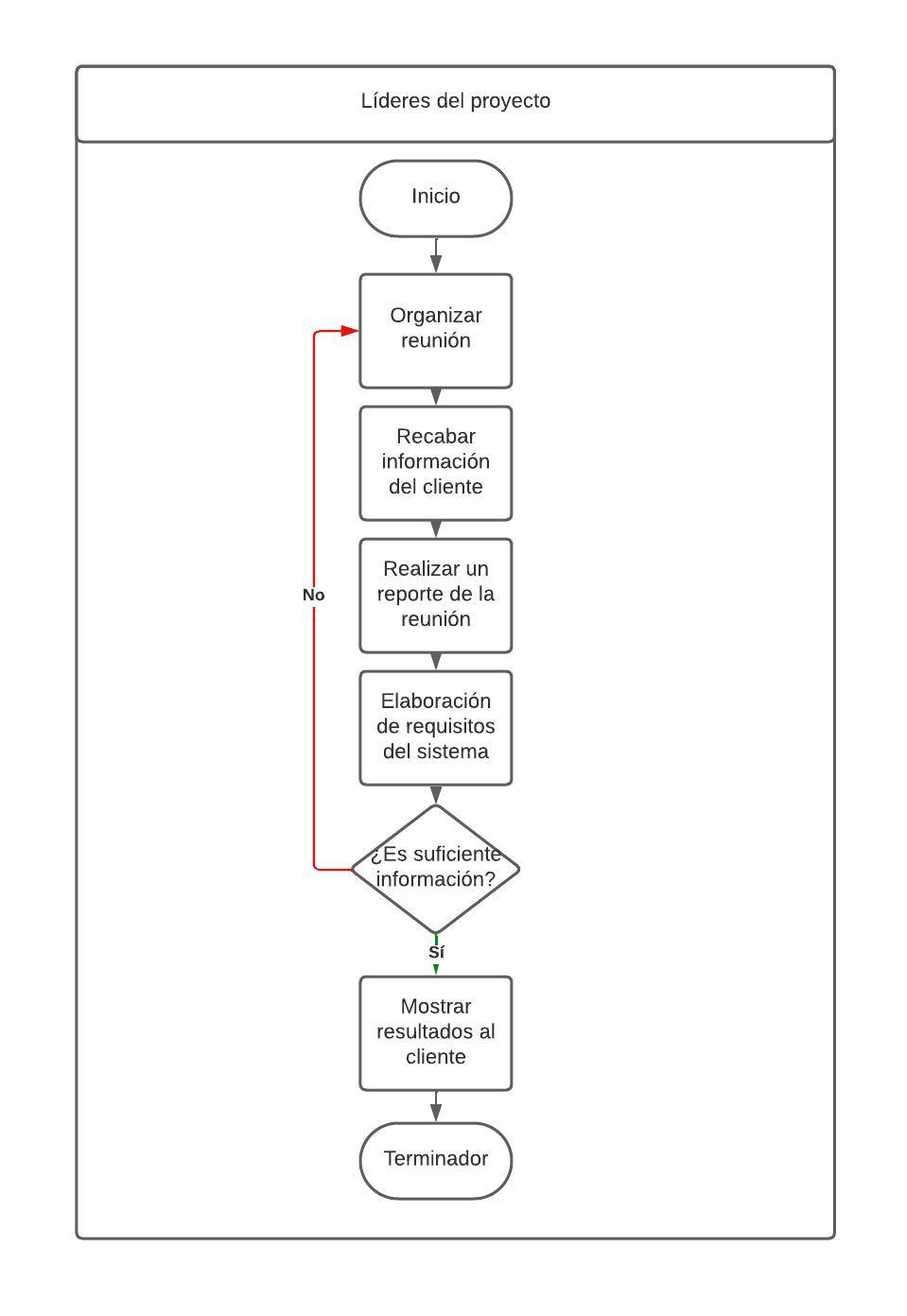
Definición de los requerimientos funcionales y no funcionales del proyecto

## **POLÍTICAS DE OPERACIÓN**

Los equipos de backend, frontend y plataformas participantes deben asistir a las reuniones con los clientes para afinar los requisitos y resolver cualquier duda sobre el objetivo del proyecto.

Los lideres de los respectivos departamentos deben registrar todas las aclaraciones y realizar reportes de las reuniones.

## DIAGRAMA DE FLUJO DE PROCESO



# **PROCEDIMIENTO: Diseño de datos**

Se manda el trabajo de diseño de base de datos al departamento de API’s.

Se determina una base de datos con respecto a la información de los requisitos del sistema.

El diseñador de base de datos mediante el proceso genera sus diseños conceptuales, lógicos y el diseño de una base de datos.

Se despliega la base de datos diseñada hacia los programadores para su uso.

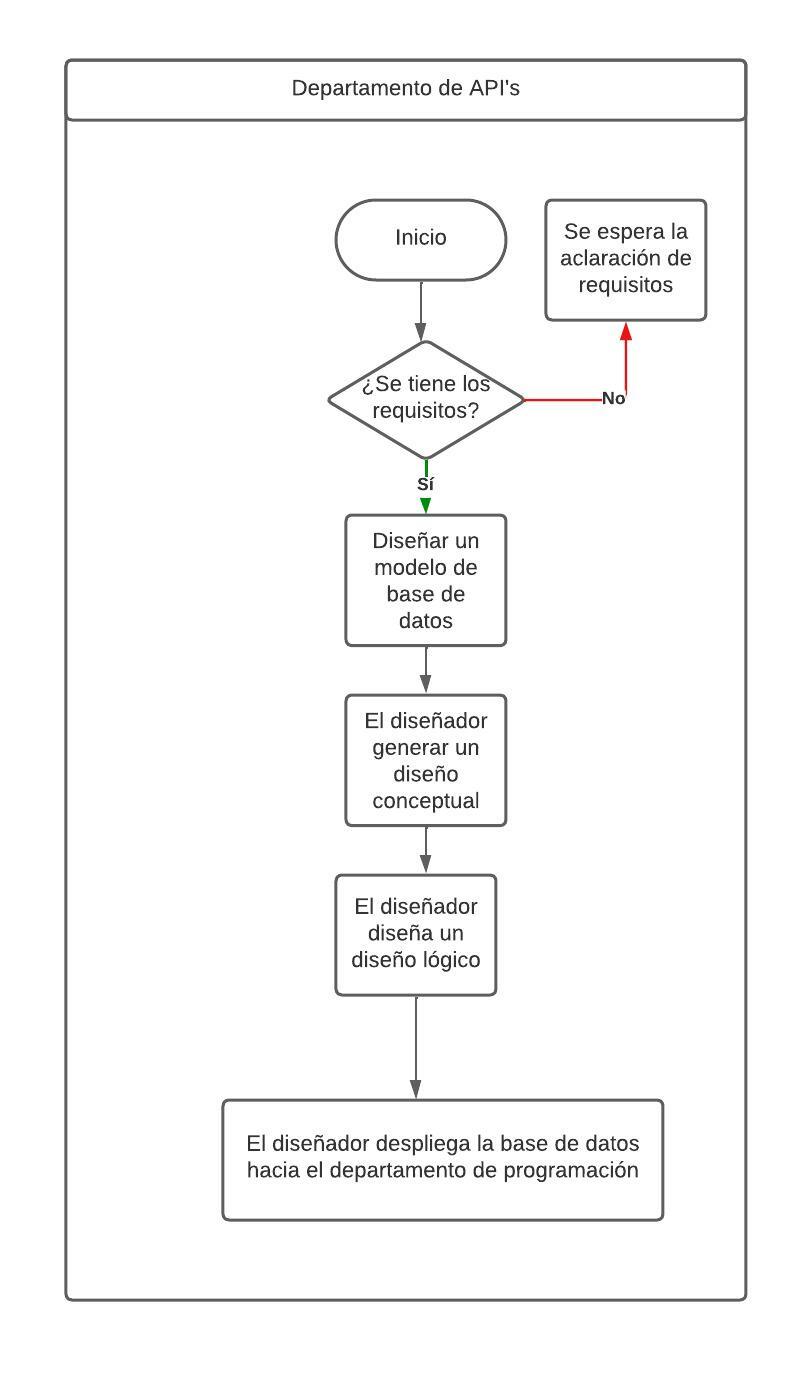
## **OBJETIVO**

Diseño de la base de datos.

## **POLÍTICAS DE OPERACIÓN**

Proceso de revisión: Se establece la práctica de hacer revisiones regulares y validaciones en el diseño de la base de datos para garantizar que sea acorde al sistema y los requisitos.

## DIAGRAMA DE FLUJO DE PROCESO



# **PROCEDIMIENTO: Diseño de arquitectura**

Respecto a esta organización, el diseño de la arquitectura consiste en reunionés para definir el panorama del proyecto y localizar elementos que puedan reutilizarse de proyectos anteriores comparando las funcionalidades y la estructura del software. También se manejaban arquitecturas estándares como MCV, pues diferentes de sus proyectos eran elaborados en esta arquitectura y en un framework orientado a mvc.

## **OBJETIVO**

Utilizar los conocimientos de programación para planear el diseño general del software, de modo que puedan agregarse detalles más adelante, lo que permite a los equipos de software delimitar el panorama general y elaborar un prototipo.

## **POLÍTICAS DE OPERACIÓN**

No se pueden definir políticas de operación ya que el proceso no estaba formalizado.

# **PROCEDIMIENTO: Diseño de interfaces**

La organización crea diseños basándose en peticiones del cliente. Se cuenta con un diseñador el cual realiza diseños generales para la empresa. Cuando no se cuenta con diseño, los desarrolladores front-end se encargan de las decisiones del diseño de las interfaces utilizando la retroalimentación y peticiones del cliente.

## **OBJETIVO**

Otorgar a los espacios de forma, función, utilidad y ergonomía, de forma que al usuario le resulte sencillo e intuitivo interaccionar con el ordenador.

## **POLÍTICAS DE OPERACIÓN**

No se pueden definir políticas de operación ya que el proceso no estaba formalizado.

# **PROCEDIMIENTO: Diseño de procesos**

Se realiza la proyección del tiempo estimado requerido para el proyecto y se empieza con la planeación inicial de los módulos.

Organización de los equipos involucrados para coordinar el trabajo colaborativo, asignar tareas iniciales y estimar fechas de entregas de avances. Se establecen tiempos y fechas, sujetos a modificaciones por inconvenientes en el desarrollo o fallas en las estimaciones de tiempo.

También deben considerarse las tecnologías y su viabilidad y alternativas, si las primeras opciones presentan problemas.

## **OBJETIVO**

Definir los procesos por los que se estima que el proyecto debe pasar para poder ser realizado en tiempo y forma.

## **POLÍTICAS DE OPERACIÓN**

Las planeaciones deben realizarse antes de iniciar la construcción del software y debe considerar los posibles contratiempos y los otros factores que puedan afectar al desarrollo.

La planeación deberá mejorarse junto al crecimiento del proyecto.

Deben realizarse revisiones frecuentes del estado de los procesos y su avance

# **PROCEDIMIENTO: Codificación del proyecto**

El Project Manager realiza una estimación del desarrollo de software para las entregas programadas con los avances al cliente.

Se verifica que el equipo de producción de software o desarrolladores estén listos para recibir las tareas designadas por el Project Manager.

Se realiza el proceso de codificación según las tareas, se realizan diseños temporales si hay que acceder a la base de datos. Se integran las interfaces si son válidas para implementación del producto.

El Project manager realiza revisiones cada cierto tiempo para verificar que se cumpla los plazos de entrega.

Los desarrolladores realizan pruebas y reparan errores producidos en el desarrollo.

Si la entrega está lista, se entrega el avance del proyecto en una reunión con el cliente. En caso contrario, se posterga la reunión.

## **OBJETIVO**

Se realiza el proceso de codificación mediante el uso de la metodología SCRUM en la organización.

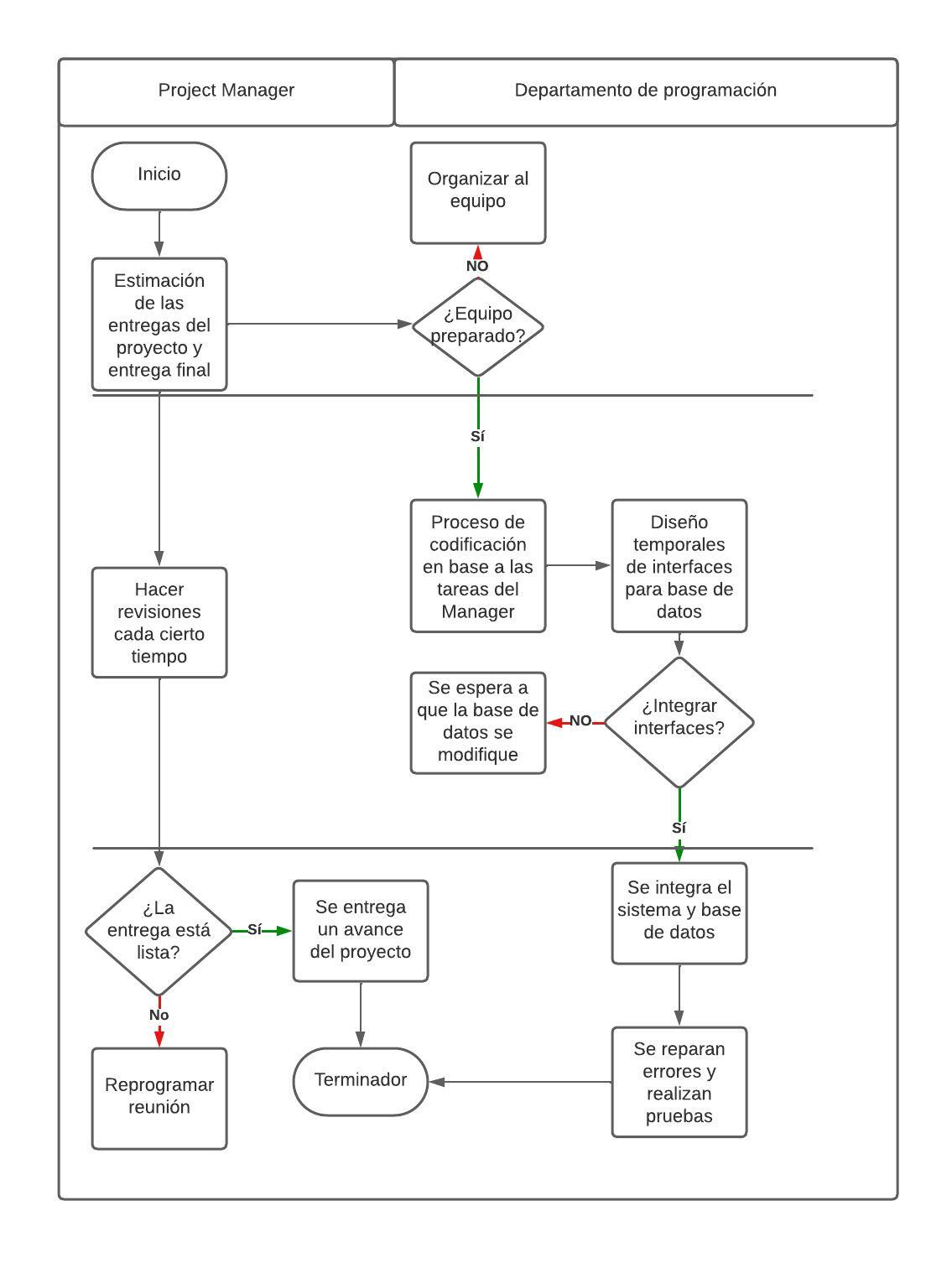
## **POLÍTICAS DE OPERACIÓN**

Solo realizar cambios autorizados en el repositorio.

Revisiones previas a cambios para detectar errores.

Mantener los repositorios locales actualizados para evitar incongruencias.

## DIAGRAMA DE FLUJO DE PROCESO



**PROCEDIMIENTO: Pruebas**

Se realizan pruebas de los artefactos de software por parte de los mismos desarrolladores y lideres de equipos.

## **OBJETIVO**

Se realiza el proceso de codificación mediante el uso de la metodología SCRUM en la organización.

## **POLÍTICAS DE OPERACIÓN**

Solo realizar cambios autorizados en el repositorio.

Revisiones previas a cambios para detectar errores.

Realizar debug al código cada que se construya una nueva funcionalidad.

Los artefactos de software deberán ser revisados por los lideres, o sub-lideres en su defecto, de cada departamento, para ser aprobados.

**Áreas de oportunidad**

La organización carece de un proceso formalizado para el “**Diseño de interfaces**”, este proceso no se incluyó en el documento ya que no existía una forma definida. En la mayoría de los casos se usaban plantillas obtenidas en páginas de internet con ese propósito.

Contar con un proceso más formalizado en esta área, agilizaría el proceso al equipo de desarrollo. Podrán concentrarse en el desarrollo más que en el diseño visual.

**Codificación del proyecto:** Este es un proceso que carecía de verdadera organización. No se manejaba un flujo de trabajo estandarizado ni se utilizaban técnicas de gestión en el repositorio. Tampoco había estándares de construcción del software dentro de la empresa.