# Índice

[**Índice**](#_awmaq8xh7wud) **1**

[Temario 2017](#_3vid029mlog1) 2

[A. Análisis de sistemas de información](#_a9vgdqmiadll) 3

[A 1. Diagnóstico del problema y valoración de la factibilidad para el desarrollo de sistemas de información](#_fvp4ok3yux5e) 3

[A 2. Modelado de los requerimientos de un sistema de información](#_eecydcwr60zd) 4

[B. Desarrollo e implantación de aplicaciones computacionales](#_ql3fbuozf6v1) 4

[B 1. Diseño de la solución del problema de tecnología de información](#_wam1gpeign89) 4

[B 2. Desarrollo de sistemas](#_7xht2pe36fi5) 4

[B 3. Implantación de sistemas](#_nzx3hydi32nd) 5

[B 4. Aplicación de modelos matemáticos](#_ue2e8gcrk4lp) 5

[C. Gestión de proyectos de tecnologías de información](#_7kb797dl86d5) 5

[C 1. Administración de proyectos de tecnologías de información](#_t3h36dbz9bec) 5

[C 2. Control de calidad de proyectos de tecnologías de información](#_5q11t096a16f) 5

[**GUÍA**](#_ny6sj3cgq2al) **6**

[A 1. Diagnóstico del problema y valoración de la factibilidad para el desarrollo de sistemas de información](#_gxuj3w1lq8dp) 6

[Análisis preliminar de los sistemas de operación de la organización](#_d3jwaac75s8e) 7

[Planeación e Iniciación del proyecto](#_s23bhwlch3hp) 7

[Factibilidad Técnica](#_1bj5qhmmfpi2) 7

[Factibilidad Económica](#_y4mg9rma4hof) 7

[Beneficios Económicos](#_rs6w2lr7tjs6) 7

[Beneficios Tangenciales](#_beoglay2zls2) 8

[Beneficios No Tangenciales](#_u1c76v69q4rp) 8

[Costos](#_i421b3tnrqx8) 8

[Costos tangibles](#_wx54b75i0z0p) 8

[Costos intangibles](#_2aqne1x3naxo) 8

[Costos de una sola vez](#_z3rt8w5qr6gq) 8

[Costos recurrentes](#_nwy7mce7k47v) 8

[Petición del sistema](#_dqocjunyyg0n) 9

[Ejercicios](#_nmb6ltvugewd) 9

[Administración del proyecto](#_oy18n234ay0i) 9

[EJERCICIOS](#_ap3c09uk3d95) 9

[REACTIVOS](#_rz8kw55hrgwz) 10

[REACTIVOS](#_g5nrveja0rdo) 10

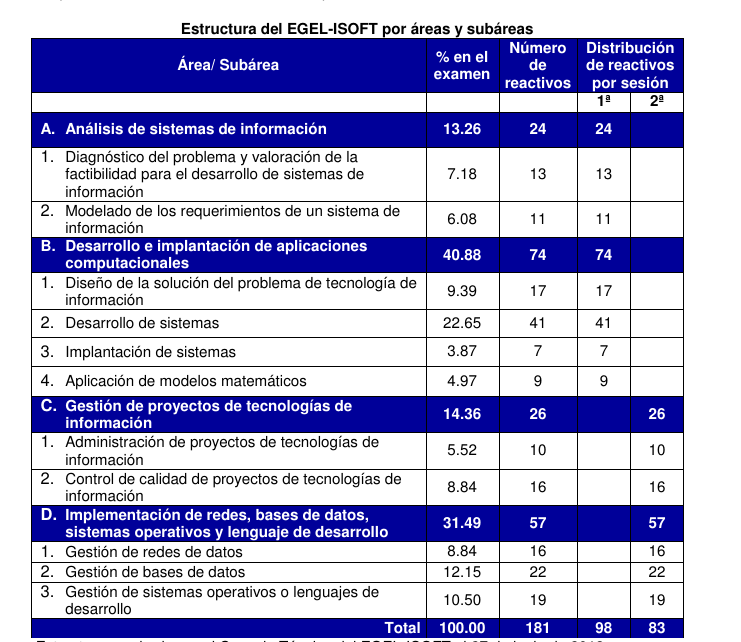
[REACTIVOS](#_n9sjvdq1j9qz) 10

[REACTIVOS](#_xh5sk4dlqxid) 11

[A 2. Modelado de los requerimientos de un sistema de información](#_jioyt523uhm4) 11

## Temario 2017

Contenido de la prueba



## A. Análisis de sistemas de información

En esta área se evalúa la capacidad de diagnosticar un problema en la operación de una

organización, utilizando como estrategias (metodologías) el análisis de factibilidad, así

como el análisis, la validación y la documentación de requerimientos a fin de proponer un

sistema de información computacional.

### A 1. Diagnóstico del problema y valoración de la factibilidad para el desarrollo de sistemas de información

En esta subárea se evalúan los siguientes temas:

* Análisis preliminar de los sistemas de operación de la organización
* Diagnóstico de la situación de los sistemas de operación de la organización
* Identificación de los problemas a resolver con sistemas de información
* Análisis de factibilidad de productos comerciales contra desarrollos a la medida como estrategias de solución del problema
* Propuestas de sistemas de información computacional que solucionen la problemática detectada en la organización

### A 2. Modelado de los requerimientos de un sistema de información

En esta subárea se evalúan los siguientes temas:

* Análisis de los requerimientos de un sistema de información
* Validación de los requerimientos de un sistema de información
* Documentación de los requerimientos de un sistema de información

## B. Desarrollo e implantación de aplicaciones computacionales

En esta área se evalúa la capacidad de elegir metodologías, técnicas y herramientas para

modelado de la solución del problema mediante la selección adecuada del lenguaje de

programación y de la herramienta de implementación, realizando la codificación y

validación de la misma. Además se evalúan las técnicas de implantación del sistema y su

simulación basada en modelos matemáticos.

### B 1. Diseño de la solución del problema de tecnología de información

En esta subárea se evalúan los siguientes temas:

* Desarrollo del modelo inicial de la solución del problema de tecnología de información
* Ajuste de modelos de la solución de tecnología de información
* Diseño de la arquitectura del sistema
* Modelado de datos para el sistema

### B 2. Desarrollo de sistemas

En esta subárea se evalúan los siguientes temas:

* Herramientas de desarrollo del sistema
* Codificación del sistema
* Validación de la solución tecnológica
* Ajuste del producto de software desarrollado

### B 3. Implantación de sistemas

En esta subárea se evalúan los siguientes temas:

* Técnicas de implantación de sistemas
* Elementos para poner en operación un producto de software
* Pruebas de operación y validez de un producto de software

### B 4. Aplicación de modelos matemáticos

En esta subárea se evalúan los siguientes temas:

* Aplicación de las ciencias básicas a la ingeniería de software
* Simulación de sistemas mediante herramientas computacionales

## C. Gestión de proyectos de tecnologías de información

En esta área se evalúan los conocimientos de las metodologías de planeación, gestión y

administración de proyectos de desarrollo e infraestructura, así como los métodos de

control de calidad en tecnologías de la información.

### C 1. Administración de proyectos de tecnologías de información

En esta subárea se evalúan los siguientes temas:

* Administración de los recursos de proyectos de tecnologías de información
* Verificación del cumplimiento de las metas del proyecto de tecnologías de información

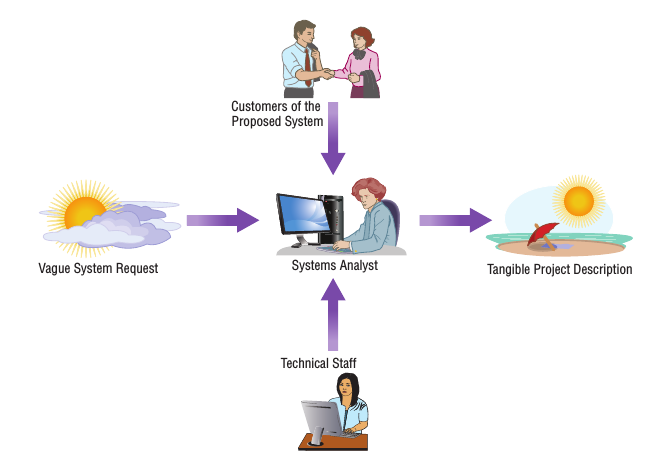
### C 2. Control de calidad de proyectos de tecnologías de información

En esta subárea se evalúan los siguientes temas:

* Selección de los modelos de calidad para tecnologías de información
* Aplicación de metodologías para el modelo de calidad seleccionado
* Establecimiento de las métricas de calidad para proyectos de tecnologías de información
* Verificación del cumplimiento de las métricas de calidad en proyectos de tecnologías de información

# GUÍA

## A 1. Diagnóstico del problema y valoración de la factibilidad para el desarrollo de sistemas de información



En esta subárea se evalúan los siguientes temas:

### Análisis preliminar de los sistemas de operación de la organización

Durante la fase de planeación del sistema es de suma importancia el entender el **por qué** un sistema de información debe de ser construído y **cómo** el equipo de desarrollo deberá construirlo.

Este proceso consta de dos pasos: Planeacion del proyecto e Iniciación del proyecto y Administración del proyecto.

#### Planeación e Iniciación del proyecto

Esta etapa incluye los siguientes puntos

* Descripción del alcance del proyecto y análisis de factibilidad
* Estimación de recursos
* Desarrollo de un calendario preliminar
* Desarrollo de un plan de comunicación
* Determinación de estándares del proyecto así como procedimientos
* Identificación y estimación de riesgos
* Ajuste de un plan de trabajo del proyecto

En esta etapa se establecen el **COSTO-BENEFICIO** del proyecto. La mayoría de las ideas para un nuevo proyecto son externas al departamento de TI y provienen principalmente del departamento de marketing, del departamento de cuentas, ventas, etc.. En esta etapa se desarrolla la petición de sistema, donde se exponen las necesidades del negocio y cómo el nuevo sistema podría generar un valor de negocio. Junto con esos departamentos el departamento de TI conduce un ANÁLISIS DE FACTIBILIDAD, en el cual se analizan aspectos clave del proyecto propuesto.

* **Factibilidad técnica (¿Podemos hacerlo?**
* **Factibilidad económica (¿Dará valor al negocio?)**
* **Factibilidad organizacional (Si se construye ¿Será utilizado?)**
* **Factibilidad operacional. (Resolverá un problema real de la empresa, es decir contribuirá a la mejora de las operaciones de la empresa?)**

#### Factibilidad Técnica

En este tipo de factibilidad debes de fijarte principalmente si el equipo de trabajo maneja bien las tecnologías del proyecto propuesto, o si es la primera vez que se hará un sistema así en el equipo de trabajo, o existe algún factor que pueda poner en RIESGO la elaboración y buen término del proyecto por parte del equipo de desarrollo.

#### Factibilidad Económica

Implica identificar **beneficios** y **costos mediante análisis costo-beneficio**, para ello normalmente se genera una hoja de cálculo con los costos-beneficios en ella.

##### Beneficios Económicos

Los beneficios económicos tienen dos grandes grupos: **tangibles** y **no tangibles**,

###### Beneficios Tangibles

Los beneficios tangibles son aquellos que pueden medirse en pesos y con precisión. Un beneficios tangible es por ejemplo, reducir los gastos del personal administrativo, bajar el costos de las transacciones o subir los márgenes de ganancias.

###### Beneficios No Tangibles

Un beneficio a la organización o empresa, que no pueda ser medido fácilmente en pesos es un beneficio no tangible, por ejemplo generar competitividad, incrementar la flexibilidad, incrementar la moral de los trabajadores, etc.

##### Costos

Un sistema a su vez puede tener costos tangibles e intangibles, costo de un solo tiempo y costos recurrentes.

###### Costos tangibles

Es un costo asociado a un sistema que puede ser fácilmente medido en pesos, por ejemplo costos de hardware, sueldos a desarrolladores, contrato de servidores, etc.

###### Costos intangibles

Un costo asociado a un proyecto que no puede ser fácilmente medido en pesos, por ejemplo ineficiencia operacional, clientes insatisfecho por mal servicio, etc.

###### Costos de una sola vez

Son costos que se efectúan una sola vez, como por ejemplo cuando se inicia una empresa, iniciación del proyecto, etc.

###### Costos recurrentes

Son costos que constantemente se dan dentro del proyecto y que normalmente están asociados a su mantenimiento, por ejemplo pagos por servicios de nube, pagos por hospedajes web, etc.

###### Costos operacionales

Son aquellos costos tangibles que se requieren para que el sistema funcione, por ejemplo compra de licencias, costos del staff de soporte técnico, cargos por uso de servidores, etc.

###### Costos de desarrollo

Son aquellos costos tangibles relacionados a los salarios de los desarrolladores, entrenamiento de los desarrolladores, gastos de oficinas y renta para el equipo de desarrolladores, etc.

#### Factibilidad Organizacional

Se lleva a cabo para investigar si el nuevo sistema será utilizado por la empresa y que tan bien será aceptado por los usuarios para trabajar en él.

#### Factibilidad Operacional

Se lleva a cabo para saber si la organización a la que se le va a desarrollar el nuevo sistema le será de utilidad para poder resolver algún problema relativo a las operaciones empresariales de dicha organización. Es decir la capacidad del nuevo sistema para resolver problemas del negocio de la empresa.

En la actualidad los nuevos sistemas se generan a partir de la ADMINISTRACIÓN DEL PROCESO DEL NEGOCIO, la cual es una metodología usada por las organizaciones para mejorar continuamente para mejorar los procesos de negocio end-to-end.

La Administración Del Proceso De Negocio sigue un ciclo donde sistemáticamente se crea, mejora y alteran los procesos de negocios. Los analistas de negocios aplican esto para:

1. Definir y mapear los pasos en un proceso de negocios
2. Crear nuevas formas de mejorar el proceso para agregar valor
3. Encontrar formas de eliminar pasos que no agreguen valor
4. **Crear o ajustar flujos de trabajo electrónicos que se ajusten al mapa de mejoras de procesos.**

Este último paso es particularmente relevante, ya que aquí es donde se identifica la necesidad del software del negocio en cuestión. Es la base de muchos s**istemas de tecnologías de la información**.

#### Petición del sistema

Es un documento que describe las **razones** para construir un sistema y **el valor** que el sistema provea a la empresa. Este documento normalmente incluye 5 elementos:

1. **Patrocinador del proyecto**: La persona que inicia el proyecto y que sirve como el punto primario de contacto del proyecto del lado del negocio.
2. **Necesidad del negocio**: La razón relacionada al negocio para iniciar el sistema.
3. **Requerimientos del negocio**: Las capacidades de negocio que proveerá el sistema
4. **Valor del negocio**: Los beneficios que el sistema creará para la organización.
5. **Restricciones:** Asuntos que son relevantes para la implementación del sistema y que deben ser aprobados por el comité,

#### EJERCICIOS

1. La aerolínea Australiana Qantas tuvo un fallo de TI. El proyecto eQ un proyecto de 10 años con IBM, se canceló después de 10 años a un costo de $200 millones de dólares. Este tipo de fallas es por.
2. Malos desarrolladores
3. Problemas técnicos
4. **Problemas organizacionales**
5. Falta de staff preparado.

2. Describe las fases del ciclo de vida de un software. R\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Describe la diferencia entre una fase, una actividad y una tarea.

4. Una empresa necesita apoyo para su campaña de marketing, en la cual quiere alcanzar nuevos tipos de clientes. Este es un ejemplo de:

R. Necesidades del negocio

5. Una empresa está preocupada por la caída en las ventas de cierto producto y por el pobre servicio al cliente que posee. Este es un ejemplo de:

R. Necesidades del negocio

6. Una empresa necesita un nuevo plan debido al incremento de ventas de la competencia. Este es un ejemplo de:

R. Necesidades del negocio.

4. La etapa de planeación es parte del proceso?

* Factibilidad técnica
* Factibilidad económica (valor al negocio)
* Factibilidad organizacional.

5. Enumera los análisis necesarios que deben hacerse en la etapa de Planeación.

### Administración del proyecto

En esta etapa se se crea un **PLAN DE TRABAJO**, selecciona el staff del proyecto y las técnicas a implementar para su desarrollo.

* Diagnóstico de la situación de los sistemas de operación de la organización

#### REACTIVOS

* Identificación de los problemas a resolver con sistemas de información

#### REACTIVOS

* Análisis de factibilidad de productos comerciales contra desarrollos a la medida como estrategias de solución del problema

#### REACTIVOS

* Propuestas de sistemas de información computacional que solucionen la problemática detectada en la organización

#### REACTIVOS

### A 2. Modelado de los requerimientos de un sistema de información

En esta subárea se evalúan los siguientes temas:

* Análisis de los requerimientos de un sistema de información
* Validación de los requerimientos de un sistema de información
* Documentación de los requerimientos de un sistema de información

#### Análisis de los requerimientos de un sistema de información

Las actividades dentro de los requerimientos son las siguientes:

1. Elicitar requerimientos
2. Expresar requerimientos
3. Priorizar requerimientos
4. Analizar requerimientos
5. Administrar requerimientos

##### Elicitar requerimientos

Proceso interactivo y de investigación, el cual ocurre cuando se reúne el administrador del proyecto con los clientes y usuarios del futuro sistema. En esta parte se necesita diferenciar el “Yo quiero” del “Yo necesito”. En la primera se tienen características del software que podrían ser necesarias y las segundas son características clave o núcleo.

En esta parte se analiza si dichas caracteristicas son posibles y si las expectativas del cliente son realistas.

##### Expresar requerimientos

Esta actividad involucra darles un límie a los requerimientos. En esta parte se buscar representaciones de dichos requerimientos tales como casos de uso, historias de usuario y los story boards

##### Priorizar requerimientos

En metodologías tales como Scrum es de suma importancia hacer la prioridad de acuerdo a las necesidades del cliente. Se deben hacer para ello las siguientes preguntas

* Que requerimientos deben ser completados primero
* Que requerimientos son críticos para el sistema
* Que necesita hacerse para mejorar el producto

##### Analizar requerimientos

Es la acción de examinar los requerimientos enlistados para asegurarse que son claros, completos y consistentes.

##### Administrar requerimientos

Este es un proceso continuo. Involucra organizar y reorganizar. Involucra también dar seguimiento de las prioridades, análisis y cambios en los requerimientos.

### 

### 

### 

### 

### 

### B 1. Diseño de la solución del problema de tecnología de información

Antes del desarrollo de la solución del problema es importante que identifiques las actividades llevadas a cabo en la ingeniería de Software:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FASE ADMINISTRATIVA DEL PROYECTO** |  | **FASE DE ESPECIFICACIÓN** |  | **FASE DE DISEÑO E IMPLANTACIÓN** |  | **FASE DE VERIFICACIÓN Y VALIDACIÓN** |
|  |  |  |  |  |  |  |
| **Crear un proceso** |  | **Identificar ideas o necesidades** |  | **Diseñar la arquitectura** |  | **Desarrollar procedimientos de test** |
| **Ajustar estándares** |  | **Elicitar requerimientos** |  | **Diseñar las bases de datos** |  | **Crear Test** |
| **Administración de riesgos** |  | **Expresar requerimientos** |  | **Diseñar las interfaces** |  | **Reportar los resultados de evaluaciones** |
| **Ejecutar estimaciones** |  | **Priorizar requerimientos** |  | **Crear código ejecutable** |  | **Auditoría y revisión** |
| **Asignar recursos** |  | **Analizar requerimientos** |  | **Integrar funcionalidad** |  | **Demostraciones a clientes** |
| **Hacer mediciones** |  | **Administrar requerimientos** |  | **Documentar** |  | **Conducir retrospectivas** |
| **Mejorar procesos** |  | **Formular enfoques potenciales** |  |  |  |  |

#### Desarrollo del modelo inicial de la solución del problema de tecnología de información.

Para entender el desarrollo de un proyecto de software es importante que comprendamos que establecer un **proceso** es el punto inicial de todo proyecto de software. **Un proceso** está organizado en distintas **fases** y cada fase contiene distintas **actividades** y **tareas**, relacionadas entre sí.

El **ciclo de vida de un software** se refiere al proceso que cubre todo el espectro, desde la incepción del proyecto, hasta su posible retiro del mercado.

Del bloque de actividades de software, cada una de ellas puede contener varias tareas, por ejemplo, de la actividad crear Test tenemos las tareas, escribir test, diseñar test, ejecutar test.

Las tareas pueden tener dependencias entre ellas, por ejemplo, una tarea no puede iniciarse, si antes no se termina una de la cual es dependiente.

Otros ejemplos de tareas son: Escribir código fuente, escribir documentación, instalar una biblioteca.

#### Ajuste de modelos de la solución de tecnología de información

Adoptar un proceso en particular será clave en la estructuración del trabajo y el éxito de un proyecto de software. Diseñar un proceso puede iniciarse desde ceros, o puede uno basarse en uno de los varios **Modelos de Proceso** que ya se han diseñado y usado con éxito.

Los procesos de software se dividen en los siguientes rubros

1. **Modelos de Proceso Lineales:** Las fases suceden secuencialmente una tras otra
2. **Modelos de Proceso Iterativos:** Las fases son repetidas en ciclos
3. **Modelos de Proceso Paralelos:** Las actividades pueden ocurrir concurrentemente.

##### Actividades de Administración del proyecto.

Las actividades de esta fase son:

|  |
| --- |
| **Crear un proceso** |
| **Ajustar estándares** |
| **Administración de riesgos** |
| **Ejecutar estimaciones** |
| **Asignar recursos** |
| **Hacer mediciones** |
| **Mejorar procesos** |

**Crear un proceso** es sumamente importante porque ajusta la forma y el modelo en que se trabajará el proyecto.

**Ajustar estándares** trata de estandarizar convenciones en la escritura del código, niveles de documentación, estrategias de testing y aspectos como cuando un trabajo es considerado como terminado.

**Manejo de riesgos**: Consiste en administrar los riesgos: Técnicos, administrativos, de negocios, de tiempos, y de seguridad. El producto de trabajo de esta actividad es un **Plan de riesgos.**

**Ejecutar estimaciones**: Esta actividad implica asignar tiempos a cuánto tarda en ejecutarse una tarea.

**Hacer mediciones**: Trata de establecer métricas para hacer un seguimiento y verificación de la calidad de un producto o proceso.

##### Actividades de Fase de Especificación

Es el punto de inicio del proyecto de software. Las actividades de esta fase se enfocan en los requerimientos. En un proceso lineal éstas ocurren una sola vez al inicio del proyecto, pero en un proceso iterativo, estas son revisadas continuamente para verificar si hay cambios y adaptarse a los cambios que quiera el cliente.

Las actividades de esta fase son:

|  |
| --- |
| **Identificar ideas o necesidades** |
| **Elicitar requerimientos** |
| **Expresar requerimientos** |
| **Priorizar requerimientos** |
| **Analizar requerimientos** |
| **Administrar requerimientos** |
| **Formular enfoques potenciales** |

1. Identificar ideas o necesidades: Entender y explorar las ideas del cliente.
2. Elicitar requerimientos: Definir los requerimientos ( 5 técnicas:Entrevista, JAD, Cuestionarios, Análisis de documentos, Observación)
3. Expresar requerimientos: Escribirlos en una forma estructurada
4. Priorizar requerimientos: Asignarles un rango de importancia
5. Analizar requerimientos: Checar su claridad, consistencia y que estén completos
6. Administrar requerimientos: Organizarlos, reutilizarlos y referenciarlos
7. Formular enfoques potenciales: Permite al equipo tener una tener una estrategia o alternativas si algo no funciona.

##### 

##### 

##### Actividades de Fase de Diseño e implantación

Las actividades de esta fase son:

|  |
| --- |
| **Diseñar la arquitectura** |
| **Diseñar las bases de datos** |
| **Diseñar las inetrfaces** |
| **Crear código ejecutable** |
| **Integrar funcionalidad** |
| **Documentar** |

##### Actividades de la fase de verificación y validación

Las actividades de esta fase son:

|  |
| --- |
| **Desarrollar procedimientos de test** |
| **Crear Test** |
| **Reportar los resultados de evaluaciones** |
| **Auditoria y revisión** |
| **Demostraciones a clientes** |
| **Conducir retrospectivas** |

La **verificación** consiste en una revisión técnica de si el software hace lo que debería hacer, de acuerdo a los requerimientos.

La **validación** una de cuyas actividades es Demostraciones a Clientes, checar si el producto se ajusta a las necesidades del cliente.

La actividad de **Conducir retrospectivas** es usada para detectar áreas de mejora y ajustar el proyecto en la siguiente iteración.

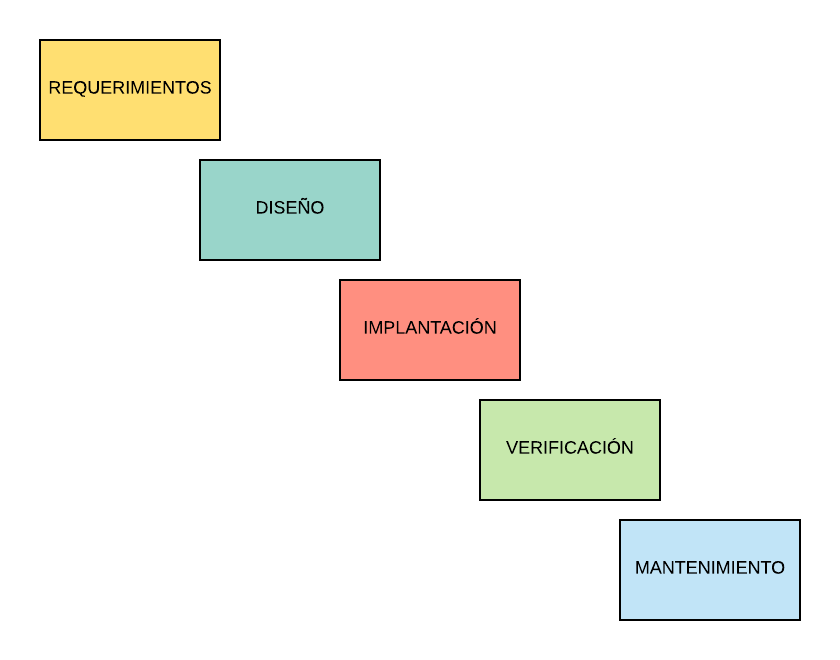
###### Modelos lineales

Son tres:

1. Cascada
2. Modelo V
3. Diente de sierra

###### Cascada

Cada fase produce un producto de trabajo de entrada para la siguiente. En el siguiente diagrama se muestra este **modelo de proceso de software**

****

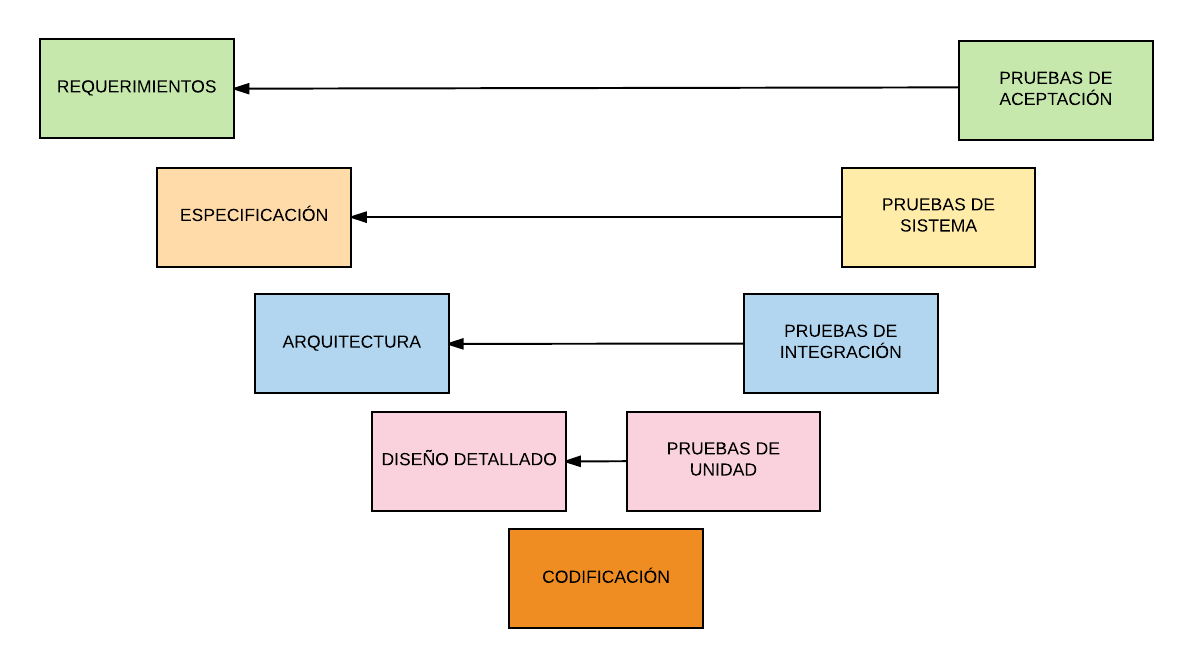
Basados en este diagrama, para llegar a la fase de diseño se debe tener como producto previo el documento de requerimientos, formalmente aprobado y así, sucesivamente hasta terminar con el proceso.

Ventajas: Enfatiza el término de una etapa sobre la otra y puede adoptarse para desarrollar partes más específicas dentro de un proyecto muy grande.

Desventajas: No se adapta al cambio. No se ajusta a las metodologías ágiles. El modelo de cascada enfatiza conocer completamente los requerimientos al inicio del proyecto, sin permitir cambio alguno durante el desarrollo del mismo. En este modelo los clientes no ven el producto sino hasta que esté completamente terminado.

###### Modelo V

El modelo V se ilustra en el siguiente diagrama:

[](https://www.lucidchart.com/documents/edit/0a07b015-b3da-4936-aed5-107868c742f7/0?callback=close&name=docs&callback_type=back&v=583&s=595.4399999999999)

Este es otro modelo Lineal y fué creado como respuesta al problema que surgía en el modelo de cascada, en el cual no podíamos ir hacia atrás, en este modelo se tienen dos ramales, el de la izquierda y el de la derecha. El de la izquierda corresponde a las etapas de análisis y diseño y la rama de la derecha corresponde a actividades de testing e integración. Cada etapa de la derecha es verificada contra la correspondiente en el lado izquierdo.

Ventaja: Es fácil de entender. Permite verificar el producto a múltiples niveles.

Desventaja: No se ajusta a los cambios. El cliente no ve el producto hasta que está completamente terminado.

Aquí se pone especial énfasis en la etapa de testing: Estados y tipos.

1. Estado:Pruebas de unidad

Tipos: Black-box: Trata los programas como caja negra. White-box: Verifica detenidamente el código no solamente la salida.

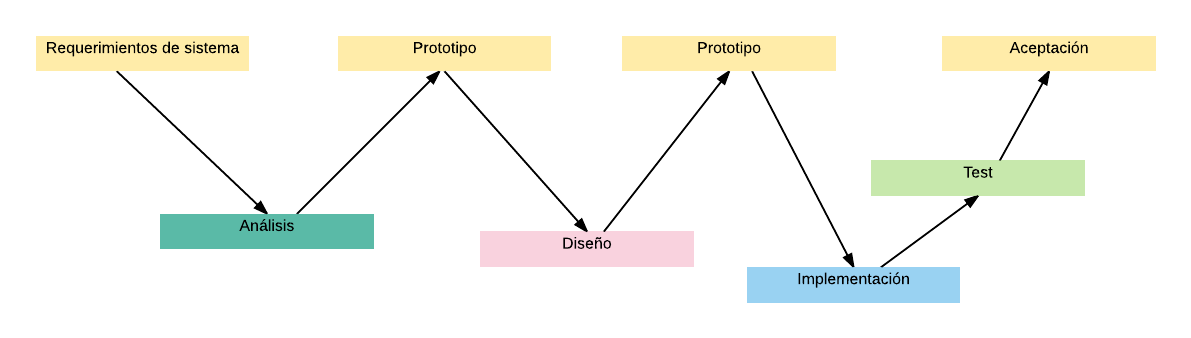
2. Estado: Pruebas de integración: Pruebas de interfaz de usuario: Verifica que cada menú funcione correctamente. Prueba de flujo de datos. Prueba cada etapa del flujo de datos paso a paso. Prueba de Sistema de interfaz de usuario: Prueba que el sistema se comunique bien con un segundo o tercer sistema.

3. Estado: Pruebas de sistema: Tipos: Prueba de requerimientos. Son aquellas que reúnen las pruebas de unidad, de integración y prueban al sistema en su totalidad. Pruebas de usabilidad: Prueban la interfaz de usuario. Pruebas de seguridad, Pruebas de desempeño.

4. Estado: Pruebas de Aceptación:son aquellas ejecutadas por el cliente terminal. Tipos: Alfa testing. Beta testing.

###### Modelo diente de sierra

Es también un modelo lineal, A continuación se muestra su diagrama:

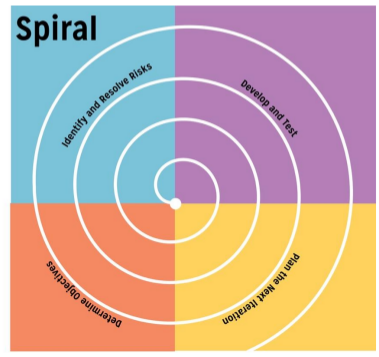
[](https://www.lucidchart.com/documents/edit/cd8ac37d-de36-424e-b0fe-fa23510ba6ce/0?callback=close&name=docs&callback_type=back&v=306&s=595.4399999999999)

A diferencia de los de cascada y V, el cliente si es involucrado para revisar los prototipos intermedios, pudiendo así ver el producto antes , a diferencia de los otros dos lineales. A pesar de que aquí el cliente sí puede verificar algunos prototipos, sigue siendo un modelo de proceso lineal, y los cambios incorporados son aún muy limitados.

##### Modelos de proceso Iterativos

Difieren de los lineales en que éstos están diseñados para revisitar o repetir estados en el proceso. A este proceso se le denomina i**teración.**

###### Modelo en Espiral



Este modelo iterativo se caracteriza por repetir o revisitar fases previas. Cada circuito completo es una iteración. En la que el cliente puede retroalimentar al equipo de desarrollo. Este modelo tiene 4 fases:

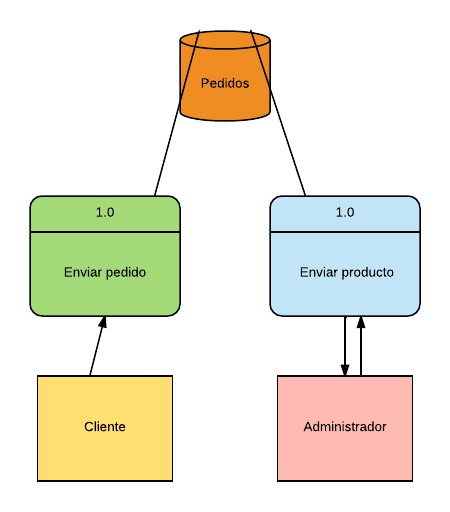
1. Determinar objetivos
2. Identificar y resolver riesgos
3. Desarrollar y testear
4. Planear la siguiente iteración

Cada iteración resulta en un prototipo. Esto resulta en un **producto** que el equipo puede revisar con el cliente. Las primeras iteraciones, que son las más internas, llevan a ideas y conceptos y las más externas llevan a prototipos funcionales

#### Modelado de datos para el sistema

Los diagramas de Flujo de Datos se usan para modelar un proceso de negocios y los pasos en ese proceso. Estos diagramas están constituidos por:

1. **Procesos**: Es una actividad o función definida del software. Ejemplo. “Determinar costos de cantidad pedida”.
2. **Flujos de datos**: Es un elemento de datos: Ejemplo.- “Cantidad disponible”
3. **Almacenes de datos**: Son el punto inicial del modelo de datos. Todos los almacenes de datos al menos deben tener mínimo un flujo de datos
4. **Entidades externas**: Representa una persona, empresa u organización que interacciona: Ejemplos -”Cliente”, “Empresa abastecedora”



Usualmente muestra las 3 formas normales de un sistema utilizando el paradigma relacional.

1. Primer forma normal. Identificar la primary key y campos repetidos
2. Segunda forma norma: Identificar y separar campos que no dependen de la primary key
3. Tercer forma normal: Separar dependencias transitivas

##### Tipos de relaciones de datos:

1. Direccionalidad

3. Cardinalidad

2. Ordinalidad

#### Diseño de la Arquitectura del Sistema

El análisis trata de **Que** se va a construir y el análisis de **cómo** se va a construir. En este apartado es importantes diferenciar los modelos y patrones de diseño utilizados más comúnmente, como es el caso de **MVC**.

#### Tipos de arquitecturas comunes

* Arquitectura cliente servidor(2-ramas, n-ramas)
* Datacentrica
* Flujo de datos (pipes y filters)
* Jerarquica
* Orientada a servicios(distribuída)

### B 2. Desarrollo de sistemas

En esta subárea se evalúan los siguientes temas:

* Herramientas de desarrollo del sistema
* Codificación del sistema
* Validación de la solución tecnológica
* Ajuste del producto de software desarrollado