



Bases de datos relacionales

*Modelamiento de Bases de Datos
Relacionales*





Bases de datos relacionales

Modelamiento de Bases de Datos Relacionales





ESCUELA CONSTRUCCIÓN E INGENIERÍA

Director de Escuela / Marcelo Lucero Yáñez

ELABORACIÓN

Experto disciplinar / Aida Villamar Gallardo

Diseñador instruccional / Camila Orellana Díaz

VALIDACIÓN

Experto disciplinar / Alex Flores Fuentealba

Jefa de diseño instruccional y multimedia / Alejandra San Juan Reyes


DISEÑO DOCUMENTO

Equipo de Diseño Instruccional AIEP



Contenido

Introducción al módulo	6
Aprendizajes esperados	7
La arquitectura de una base de datos	8
¿Qué es un dato?	9
Clasificación de datos	10
Revisa a continuación los principales tipos de datos: numéricos, fecha y aquellos con formato <i>string</i>	10
Tipos de datos más importantes	11
¿Qué es un proceso?	14
Sistema de información (SI)	14
Clasificación de los sistemas de información.....	16
Ciclo de vida de un sistema de información (SDLC).....	20
Modelos para planificar el proceso	23
Calidad dentro del desarrollo de un sistema de información	33
Garantía de la calidad del software	33
Calidad de software	33
Costo de la calidad	34
Modelo de negocios	35
Modelo de negocio Canvas.....	35
Segmentos de clientes	35
Propuesta de valor	36
Canales.....	36
Relación con el cliente	37
Fuente de ingresos.....	37
Recursos clave.....	38
Actividades clave.....	38
Socios clave.....	39
Estructura de costos	40



Ejemplo de aplicación del modelo Canvas	41
Arquitectura de un sistema de base de datos.....	44
Objetivos de los DBMS.....	44
Independencia lógica y física de los datos.....	45
Requerimientos de clientes	48
Definición de requerimiento.....	48
Importancia de los requerimientos:	49
Documentos de requerimientos.....	50
Clasificación de requerimientos	52
Contextualización del análisis del problema (Antecedentes).....	56
Modelos de datos	57
Tipos de modelos de datos	59
Modelo conceptual	59
Modelo lógico	61
Modelo Físico.....	63
Requerimientos iniciales y reales para la mejora de un modelo de datos en una base de datos	64
Análisis de los requerimientos.....	64
Reconocer las necesidades del sistema.....	64
Recopilación de información	64
Conclusión	65
Ideas clave.....	66
Links y material de profundización	69
Bibliografía	71



Introducción al módulo

En este módulo revisarán los principales conceptos y definiciones del proceso de modelamiento de una base de datos. Podrán crear una base de datos desde el comienzo, comprendiendo la importancia de construirla correctamente, con el fin de cubrir todas las necesidades del futuro desarrollo, y por supuesto, del cliente. Es a través de sus requerimientos que se modelan los requisitos funcionales para cada uso que se quiera dar a la base de datos, representados mediante técnicas de UML, generando vistas que son orientadas al usuario.

A medida que se vaya avanzando en el módulo, se irán conociendo los elementos de los Sistemas de Información y las normas que intervienen en los elementos de las bases de datos: los Sistema de Gestión de Bases de Datos (SGBD) para Optimización de un modelo de datos.

Saber interpretar los requerimientos del cliente es uno de los aspectos más importante del proceso de modelamiento de bases de datos. Si no se logra entender las necesidades del cliente, lo más probable es que no se obtenga una solución óptima. En este sentido, sin importar su apariencia y el lenguaje de programación que use, una base de datos mal diseñada tendrá problemas de funcionamiento. Dependiendo del momento en que se detecten las fallas, podría incluso significar la cancelación del proyecto. Claramente esto se puede ver disminuido con un buen proceso de control de calidad, pero siempre implicará más tiempo y, por ende, más costo para el cliente.



Aprendizajes esperados

¿Qué aprenderás en este módulo, por unidad y por semana? Revisa la competencia asociada al módulo y lo que evaluaremos semana a semana para que la cumplas.

Unidad de competencia del módulo:	Al finalizar el módulo los participantes serán capaces de realizar diseños de bases de datos relacionales, de acuerdo con los requerimientos de información especificados por el cliente.	Duración:	72 horas
		Modalidad:	Online
Aprendizaje esperado de la semana	Analizar la arquitectura de una base de datos según requerimientos propuestos por el cliente.		
Criterios de evaluación de la semana	<p>Al finalizar esta semana serás capaz de:</p> <ul style="list-style-type: none">• Identificar arquitectura y conceptos asociados al modelamiento de una base de datos.• Identificar requerimientos iniciales para el modelamiento de una base de datos.• Clasificar tipos de modelos de datos para establecer una base de datos.• Realizar las tareas asignadas respetando normas, protocolos y necesidades en el contexto de su quehacer. (R)		

La arquitectura de una base de datos

Antes de revisar lo concerniente a la arquitectura de una base de datos, revisarás algunos conceptos previos para contextualizar y tener una visión general, para luego pasar a revisar los contenidos en detalle.



¿Qué es una base de datos?

Una **base de datos** es una colección de información organizada de forma que los programas que están en los computadores puedan tomar rápidamente aquellos fragmentos de datos que necesiten para su funcionamiento.

El concepto de **modelado de datos** está asociado a la forma de establecer una estructura y organización de los datos para que puedan ser accedidos fácilmente por las bases de datos, esto por medio de un previo análisis de la situación actual en el entorno en que se desea implementar.


Los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general, se refieren a algoritmos y conceptos matemáticos. No se refiere a algo físico. Define las reglas por las cuales los datos son descritos.

Hay tres características importantes inherentes a los sistemas de bases de datos:

- La separación entre los programas de aplicación y los datos.
- El manejo de múltiples vistas por parte de los usuarios.
- El uso de un catálogo para almacenar el esquema de la base de datos.

En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

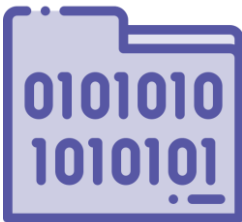
El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

-
- 
- Estructura interna.
 - Estructura conceptual.
 - Estructura externa.

Un ejemplo de arquitectura de datos bien definida son aquellas empresas que, tras conocer los cambios que se iban a aplicar a la RGPD (Reglamento General de Protección de Datos) en España, adaptaron sus bases de datos antes de que la normativa entrara en vigor.

¿Qué es un dato?

Un dato es una **representación simbólica** (numérica, alfabética, algorítmica, etc.) de un atributo¹ de una entidad². Un dato no tiene valor semántico (sentido) en sí mismo, pero al ser procesado puede servir para realizar cálculos o tomar decisiones. Pueden ser comprimidos, encriptados, transmitidos y almacenados.



En su nivel más básico, los datos de un computador corresponden a infinitas combinaciones entre ceros y uno, conocidos como datos binarios.

El concepto **entidad** se refiere a una “cosa” u “objeto” en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades.

Las entidades se describen en una base de datos mediante un conjunto de atributos. Por ejemplo, los atributos número-cuenta y saldo describen una cuenta particular de un banco y pueden ser atributos del conjunto de entidades cuenta.

¹ Atributo: Propiedad de una entidad.

² Entidad: “cosa” u “objeto” en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades. Las entidades se describen en una base de datos mediante un conjunto de atributos. Por ejemplo, los atributos número-cuenta y saldo describen una cuenta particular de un banco y pueden ser atributos del conjunto de entidades cuenta.

Datos v/s información



¿Cuál es la diferencia entre dato e información? Generalmente estos dos conceptos se utilizan indistintamente, pero son muy diferentes. La información corresponde al sentido que se le da a un conjunto de datos. Por ejemplo: Tenemos los números 16, 17, 17, 17, 17, 15, 15, 16. Se convierten en información cuando se sabe que son las edades de un grupo de alumnos de un curso.

Los datos se pueden clasificar según su estructura. Existen datos simples y datos compuestos:

- Los **datos simples** son indivisibles. Si se tratara de dividirlos, ya no serán datos, sino que serán fracciones del dato, la cual no tiene ningún significado sobre el mismo.
- Los **datos compuestos** están formados por datos simples para su estructuración y significado, dentro de un modelo de datos debidamente estructurado para el caso. Esta composición se lleva a cabo desde un inicio, cuando los datos son capturados desde el medio donde se encuentran o, se pueden componer o combinar cuando ya están capturados, estructurándose un nuevo tipo de dato desde el contenedor para una nueva definición en la información.

Revisa a continuación los tipos de datos según estructura:

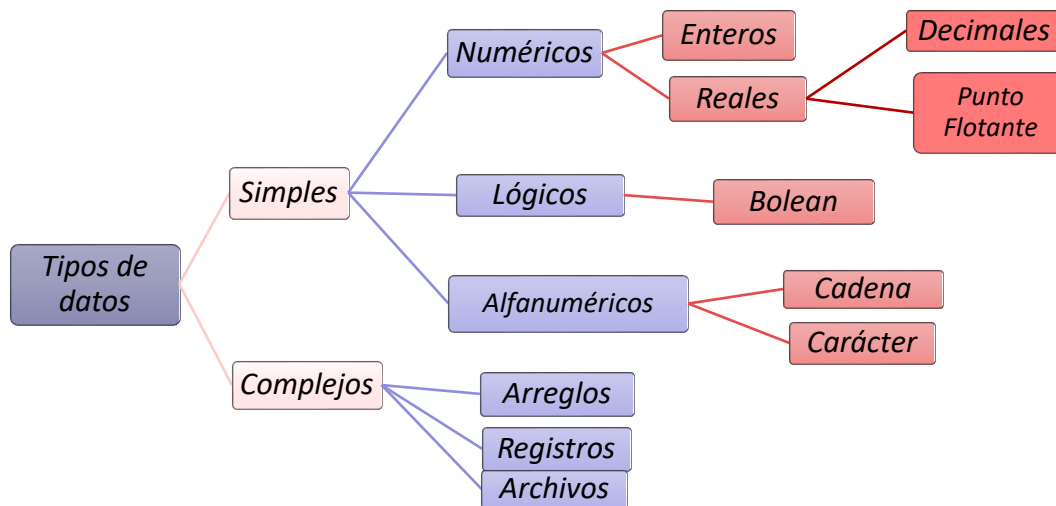


Figura 1. Tipos de Datos. Aida Villamar G., 2020.

Revisa a continuación los principales tipos de datos: numéricos, fecha y aquellos con formato *string*.



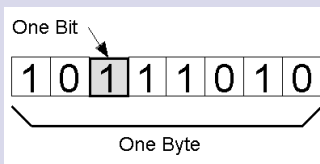
Tipos de datos más importantes

Un tipo de datos corresponde al tipo de valor que puede contener una columna, por ejemplo, si contendrá datos enteros, o tal vez, datos de caracteres, datos de fecha y hora, cadenas binarias, etc.

Es necesario que cada columna, en una tabla de base de datos, cuente con un nombre y posea un tipo de datos.

Antes de comenzar, es necesario que definamos uno de los conceptos que te ayudarán a comprender los tipos de datos:

¿Qué es un byte?



Un byte es un conjunto de 8 bits que recibe el tratamiento de una unidad y que constituye el mínimo elemento de memoria direccionable de una computadora. Por ejemplo, 4 bytes = 32 bits.

Para conocer más sobre este concepto, te invitamos a revisar el siguiente enlace:
[https://www.significados.com/byte/\[ALVG1\]](https://www.significados.com/byte/[ALVG1])

Revisa a continuación, los datos más importantes que utilizarás en este módulo:

Tipo de dato	Descripción	Ejemplo
Int (integer)	Ocupación de 4 bytes con valores entre -2147483648 y 2147483647 o entre 0 y 4294967295.	Se utilizan en campos donde se vayan a almacenar valores enteros, por ejemplo: cantidades.
blob	Una longitud máxima de 65.535 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.	Se utilizan para almacenar objetos cuyo tamaño es o puede llegar a ser muy grande. Ejemplo: <ul style="list-style-type: none">• Archivos de texto muy grandes• Documentos (.DOC, .PDF, etc.)• Hojas de cálculo (.XLS, etc.)• Gráficos y fotografías (.CAD, .CRW, .GIF, .JPG, .PNG,



		<ul style="list-style-type: none">.TIFF, etc.)• Canciones y música (.MID, .MP3, .WAV, .WMA, etc.)• Vídeos (.AVI, .FLV, .MP4, .RMVB, WMV, etc.)• Páginas web (.HTM, .HTML, .CSS, etc.)• Cualquier otro archivo
float(m,d)	Almacena números de coma flotante, donde 'm' es el número de dígitos de la parte entera y 'd' el número de decimales. Al ser de punto flotante, sus cálculos son aproximados. Permite almacenar pequeños números decimales.	Se utiliza para almacenar valores numéricos con decimales. por ejemplo: campos de este tipo para guardar los valores de la U.F. o bien las notas de un curso.
bit	(BOOL, BOOLEAN): En informática y otras disciplinas, unidad mínima de información, que puede tener solo dos valores (cero o uno). Número entero con valor 0 o 1.	Se tiene un campo bit en una tabla en SQL server, en ese caso, por ejemplo: 1 es true y 0 es false.
char	Ocupación fija cuya longitud comprende de 1 a 255 caracteres.	Se usa CHAR cuando los tamaños de las entradas de datos de la columna sean consistentes. Cuando se utiliza CHAR se está diciendo que la memoria será reservada para la cantidad de bytes que se está especificando. Por ejemplo, si se declara un CHAR de longitud 18, y se guarda "Hola" sólo se ocuparán 4 bytes y los otros 14 se llenarán con espacios. Al recibir los datos, se les suprimen los espacios. Pero siempre se ocupará la longitud que se ha definido, se llenen de datos o no. Esto posee un beneficio, y es que CHAR es más rápido que VARCHAR en cuanto a rendimiento, debido a que se sabe de antemano cuál es la longitud de los datos, no así con VARCHAR que tiene una longitud variable.
varchar	Ocupación variable cuya longitud comprende de 1 a 255 caracteres.	Se utiliza VARCHAR cuando los tamaños de las entradas de datos de la columna varían



		<p>considerablemente.</p> <p>Al declarar un tipo de dato como VARCHAR se está diciendo que será como CHAR, pero con una longitud variable.</p> <p>Esto quiere decir que sólo reservará memoria para los datos que se ingresen, así, aunque se declare un VARCHAR de 500 y se ingrese "hola", sólo guardará los 4 caracteres. Esto se utiliza por ejemplo para direcciones, nombre de empresa, nombre de personas...</p>
text	Una longitud máxima de 65.535 caracteres.	<p>Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.</p> <p>Estos campos se utilizan con frecuencia para descripciones, observaciones o notas que no necesitan contener un índice, solo ser guardados.</p>
date	Válido para almacenar una fecha con año, mes y día, su rango oscila entre '1000-01-01' y '9999-12-31'.	Almacena datos de tipo fecha.

Mantén presente la tabla anterior, ya que si bien no la utilizaremos en esta semana, podrás volver a ella cuando necesitemos comenzar a modelar bases de datos.

¿Qué es un proceso?

Un proceso es la ejecución por parte del microprocesador de diversas instrucciones (actividades, acciones y tareas), de acuerdo con lo que indica un programa.

Un proceso puede, de manera informal, pensarse como un programa en ejecución. Se define como "una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, posee un estado actual, y utiliza un conjunto de recursos de sistemas asociados".

Recuerda que...



La información corresponde a conjunto de datos organizados y procesados que constituyen mensajes, instrucciones, operaciones, funciones y cualquier tipo de actividad que tenga lugar en relación con un computador. La información está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje. Autores como Czinkota, en su libro Marketing internacional (2008), declara: "la información consiste en un conjunto de datos que han sido clasificados y ordenados con un propósito determinado".

Sistema de información (SI)

Según Alejandro Peña Ayala, en su libro Ingeniería de Software para crear sistemas de información (2006), "un sistema de información es un conjunto de elementos relacionados cuya finalidad es atender las demandas y necesidades de información de una organización".

Algunos ejemplos de pueden ser:

- **Los sistemas de control de calidad**, en los que se pide una retroalimentación al cliente y se evalúan los resultados estadísticamente para elaborar resultados interpretables por la gerencia.
- **Las bases de datos de una biblioteca**, es donde está contenido el grueso volumen de documentos (libros, revistas, tesis, etc.) de la biblioteca, con la finalidad de ubicar y recuperar cada uno lo más rápida y precisamente posible.

- **Las hojas de cálculo**, en las que se ingresa información en bruto y se la organiza de manera cuantificable para obtener directrices de conducción financiera.

Revisa, a continuación, las actividades básicas de un sistema de información y la interacción de un proceso dentro de un sistema de información:

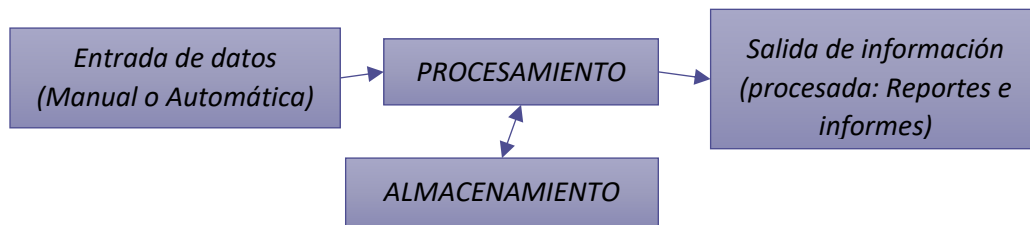


Figura 2. Actividades básicas de un sistema de información. Fuente: *Aída Villamar G. - 2020*

Conoce la descripción de las principales subetapas de la etapa de **procesamiento** dentro de un sistema de información:

- a) **Entrada:** Proceso mediante el cual se captura y prepara datos para su posterior procesamiento. Las entradas pueden ser manuales o automáticas.

Las manuales se realizan por el operador o el usuario, y las automáticas surgen de otros sistemas.

- b) **Almacenamiento:** Proceso mediante el cual el sistema almacena de manera organizada los datos e información para su uso posterior y permitir su fácil recuperación, los datos almacenados se organizan en:

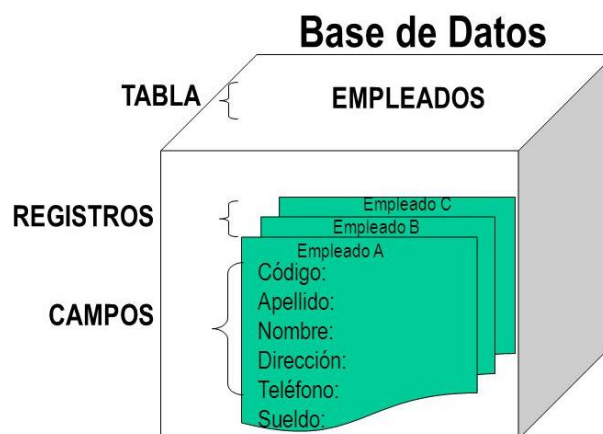



Figura 3. Componentes de una base de datos. Fuente: (images.slideplayer.es, s.f.)

-
- 
- ✓ **Campo:** Agrupación de caracteres que identifican a un sujeto, lugar u objeto. Por ejemplo, en la imagen se aprecian: **código, Apellido, Nombre...**
 - ✓ **Registro:** Conjunto de campos interrelacionados. Por ejemplo, la nómina de empleados podría componerse por el nombre, apellido, dirección y sueldo. En este caso, un registro corresponde a: Código del empleado A, Apellido del empleado A, Nombre del empleado A... Otro registro sería: Código del empleado B, Apellido del empleado B, Nombre del empleado B... y así con cada empleado.
 - ✓ **Tabla:** Conjunto de registros interrelacionados, por ejemplo, la tabla “planilla del mes enero del año 2001” podría estar compuesto por registros de la nómina de todos los trabajadores durante el mes de enero de 2001. En este caso la Tabla es **Empleados**.
 - ✓ **Base de datos:** Conjunto integrado de registros interrelacionados. Por ejemplo, la base de datos de empleados de una organización podría incluir archivos de las planillas de todos los meses, junto con otros archivos relacionados a registros de evaluación de desempeño de cada trabajador, asistencia a capacitaciones, etc.
- c) **Procesamiento:** Es la capacidad de efectuar operaciones con los datos guardados en las unidades de memoria. Durante el procesamiento se evidencia lo siguiente:
- a) Aumenta, manipula y organiza la forma de los datos.
 - b) Analiza y evalúa su contenido.
 - c) Selecciona la información para ser usada en la toma de decisiones, y constituye un componente clave en el sistema de información gerencial.
- d) **Salida de información:** Actividad que permite transmitir información útil y valiosa a los usuarios finales.

Un sistema de información debe tener control del desempeño de la aplicación, es decir, debe generar retroalimentación sobre las actividades de entrada, procesamiento, almacenamiento y salida de información. Esta retroalimentación debe evaluarse para determinar si el sistema cumple con los estándares de desempeño establecidos.

Clasificación de los sistemas de información

Los sistemas de información se clasifican o agrupan en tres tipos: según su propósito, estructura-funcionamiento y organización física. Revísalos a continuación:

- a) **Sistema de información según el propósito:** Pueden ser de tipo transaccional, de sistema de soporte, o sistemas estratégicos.



- *Sistemas transaccionales:* Son aquellos que automatizan procesos operativos en una empresa. Su función principal radica en procesar transacciones como pagos, contratos, papeletas, planillas, etc.
- *Sistemas de soporte:* Son aquellos que se basan en modelar y analizar una gran cantidad de datos, y que permiten entregar varias alternativas para escoger. Son empleados como apoyo en la toma de decisiones.
- *Sistemas estratégicos:* Son aquellos desarrollados en las empresas con la finalidad de lograr ventajas competitivas, por medio del uso de la tecnología de información.

b) **Estructural y de funcionamiento:** Pueden ser de tipo manual, mecanizado, o computacional.

- *Manual:* Cuando una persona utiliza equipos (máquinas de escribir, calculadora, archivos, etc.) y realiza las principales funciones de recopilar, registrar, almacenar, calcular y generar información.
- *Mecanizada:* Cuando alguna maquinaria realiza las principales funciones de procesamiento.
- *Computacional:* Son los sistemas que hacen uso de un computador. Hay interacción entre humano y máquina. Estos sistemas se dividen en:
 - *Batch:* El usuario proporciona los datos requeridos para que se ejecute un proceso y espera que el computador finalice la tarea para obtener los resultados.
 - *En línea:* Hay una interacción directa entre el usuario y el computador durante la ejecución de un proceso.

c) **De acuerdo con la organización física:** Pueden ser centralizados o distribuidos.

- *Centralizados:* Los recursos y los procesos se encuentran centralizados en un computador central específico, por lo que su acceso se realiza por medio de terminales remotos que acceden a ellos. A continuación, se muestra una organización centralizada, donde se puede apreciar un equipo central que posee el sistema operativo y los distintos terminales que acceden a él.



Figura 5. Sistema centralizado. Fuente imagen: (Fandom, s.f.)

- **Distribuido:** Los recursos se encuentran ubicados en distintos lugares físicamente, corresponde a una serie de computadores conectados por una red de comunicaciones, el usuario lo ve como un solo sistema y accede a los recursos remotos de la misma forma en que accede a los recursos locales.

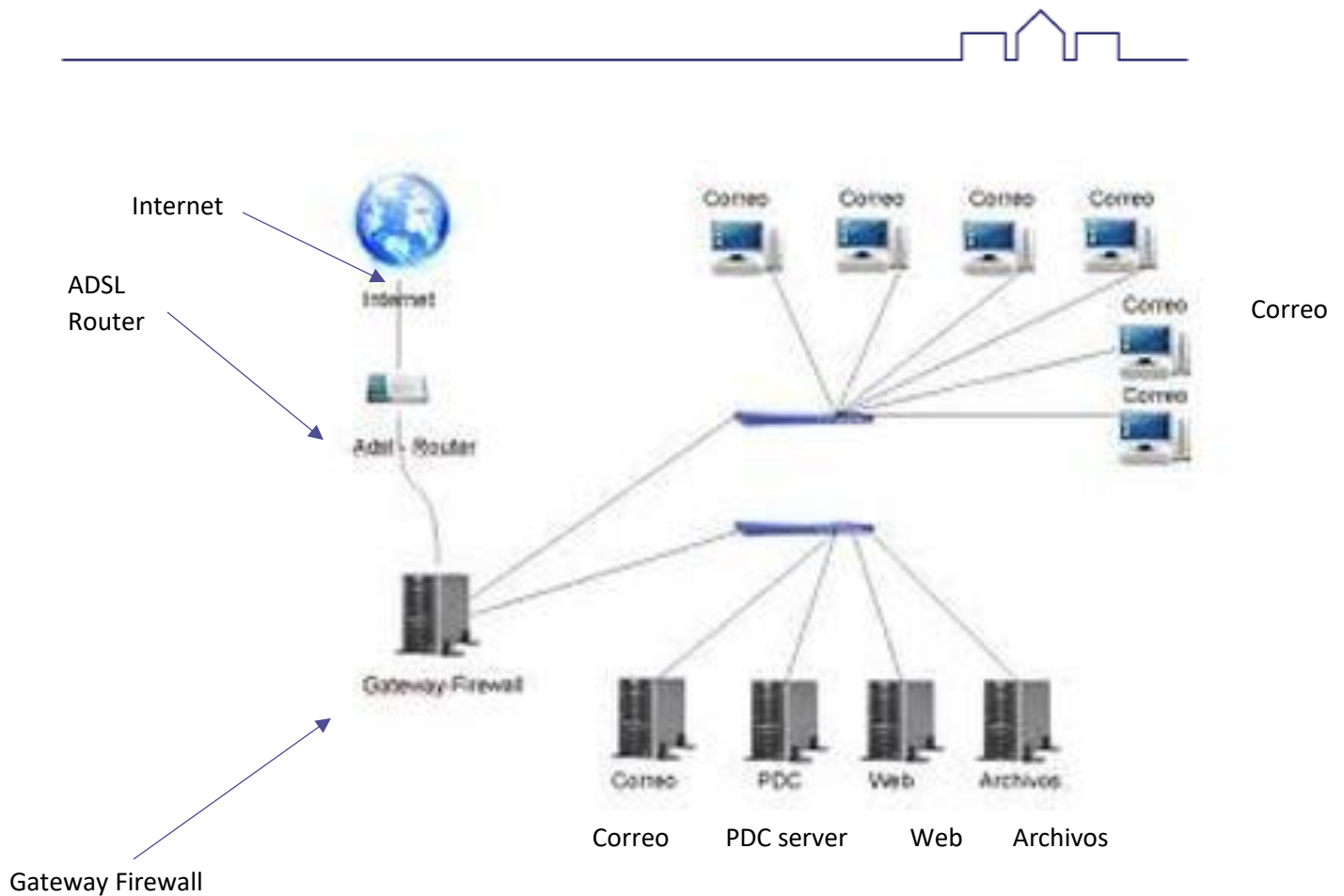


Figura 6. Sistema distribuido Fuente imagen: (weebly, s.f.)

¡Piensa rápido!



ERP (Enterprise Resource Planning)

Sistema utilizado para el manejo de los recursos de una organización. Combina funciones de varios departamentos en una sola aplicación, emplea una base de datos para entregar información más fácilmente y comunicarse entre ellos. Permite integrar, mejorar y estandarizar los distintos procesos que se realizan.

Según las clasificaciones vistas anteriormente ¿a cuál pertenece?

Respuesta: sistemas de información estratégicos

Ciclo de vida de un sistema de información (SDLC)

Todo sistema de información va pasando por una serie de etapas a lo largo de su vida. Un ciclo de vida considera una serie de fases las que serán revisadas a continuación, dentro de lo que se conoce como metodología para el desarrollo de software.

Metodología para el desarrollo de software (ciclo de vida)

La metodología para el desarrollo de software es un modo sistemático para realizar, gestionar y administrar un proyecto con el objetivo de finalizarlo con altas posibilidades de éxito. Esta sistematización permite dividir un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas nos ayuda a definir sus entradas y salidas y, sobre todo, normaliza el modo en que se administrará el proyecto. Entonces, una metodología para el desarrollo de software corresponde a los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto, hasta que se cumple el objetivo por el cual fue creado.

Desde un punto de vista general, puede considerarse que el ciclo de vida de un software tiene siete etapas claramente diferenciadas, las cuales se detallan a continuación:



Figura 7. Las siete fases del ciclo de desarrollo de sistemas (SDLC). Fuente: (KENDALL, 2011)



1. Identificación de los problemas, oportunidades y objetivos

En esta primera fase del ciclo de vida del desarrollo de sistemas, el analista se encarga de identificar correctamente los problemas, las oportunidades y los objetivos. Esta etapa es imprescindible para el éxito del resto del proyecto. Se evalúan las alternativas de solución propuestas considerando incluso opciones que no consideren un sistema computacional, lo que significaría que el proyecto no continúe. Las personas involucradas en la primera fase son los usuarios, los analistas (rol que deben cumplir inicialmente como estudiantes) y los administradores de sistemas que coordinan el proyecto.

En esta etapa las actividades radican en entrevistar a los encargados de la administración de los usuarios, resumir el conocimiento obtenido, estimar el alcance del proyecto y generar documentación con los resultados obtenidos.

2. Determinación de los requerimientos de información del factor humano:

La segunda fase a la que pasa el analista es definir las necesidades de los usuarios que están involucrados, por medio del uso de varias herramientas, para comprender como interactúan en el contexto laboral con sus sistemas de información actuales. El analista utilizará métodos interactivos como entrevistas, muestreos e investigación de datos duros, juntamente con los cuestionarios y los métodos discretos, como observar el comportamiento de los encargados al tomar las decisiones y sus entornos de oficina, y los métodos integrales como la creación de prototipos.

Las personas involucradas en esta fase son los analistas y los usuarios, por lo general los gerentes y los trabajadores de operaciones.

3. Análisis de las necesidades del sistema

La tercera fase que debe realizar el analista incluye examinar las necesidades del sistema. En esta fase hay herramientas como los diagramas de flujo de datos (DFD), diagramas de secuencia y técnicas especiales que ayudan al analista a determinar los requerimientos (más adelante lo veremos en detalle).

Durante esta etapa el profesional analiza las decisiones estructuradas llevadas a cabo. Las decisiones estructuradas son aquellas para las que se pueden determinar condiciones, alternativas de condición, acciones y reglas de acción. En este punto del ciclo de vida, el analista desarrolla una propuesta de sistemas en la que resume todo lo que ha recopilado acerca de los usuarios, la capacidad de uso y la utilidad de los sistemas actuales. Contempla un análisis de costo-beneficio de las alternativas y en caso de ser solicitado, efectúa recomendaciones. Si la administración acepta una de las recomendaciones, el análisis continúa por esa vía.



4. Diseño del sistema recomendado

En la fase de diseño, el analista se basa en la información recopilada previamente para realizar el diseño lógico del sistema de información. El analista diseña los procedimientos para ayudar a que los usuarios introduzcan los datos de manera exacta, cerciorándose de que los datos que ingresen al sistema sean los correctos. Por otra parte, el analista debe ayudar a que los usuarios completen la entrada de datos efectiva al sistema mediante el uso de las técnicas del buen diseño de formularios y páginas Web o pantallas.

Esta fase también considera el diseño de bases de datos las que contendrán muchos de los datos requeridos por los encargados de tomar las decisiones en la organización, además de diseñar los controles y procedimientos de respaldo para resguardar el sistema y los datos.

5. Desarrollo y documentación del software

En la quinta fase, el analista trabaja con los programadores para desarrollar el software original requerido. Durante ella, el analista desarrolla junto con los usuarios una documentación efectiva para el software, incluyendo manuales de procedimientos, ayuda en línea, sitios Web con preguntas frecuentes (FAQ) y archivos Léame (Read Me) para incluir con el nuevo software. Como los usuarios están involucrados desde el principio, la fase de documentación debe lidiar con las preguntas que hicieron y resolvieron junto con el analista. La documentación indica a los usuarios cómo deben usar el software y qué deben hacer en caso de que ocurran problemas. Los programadores desempeñan un rol clave en esta fase, ya que diseñan, codifican y eliminan los errores sintácticos de los programas de computadora. Para asegurar la calidad, un programador puede llevar a cabo un recorrido por el diseño o por el código para explicar las porciones complejas del programa a un equipo formado por otros programadores.

6. Prueba y mantenimiento del sistema

Antes de utilizar el sistema de información, se debe probar. Es mucho menos costoso detectar los problemas antes de entregar el sistema a los usuarios. Una parte del procedimiento de prueba es llevado a cabo por los programadores solos; la otra la realizan junto con los analistas de sistemas. Primero se completa una serie de pruebas para señalar los problemas con datos de muestra y después se utilizan datos reales del sistema actual. A menudo, los planes de prueba se crean en las primeras etapas del SDLC y se refinan a medida que el proyecto avanza. El mantenimiento del sistema y la documentación de este mantenimiento empieza en esta fase y se lleva a cabo de manera constante durante toda la vida del sistema de información. Gran parte del trabajo rutinario del programador consiste en el mantenimiento, por lo cual las empresas invierten una gran cantidad de dinero en este proceso. Muchos de los procedimientos sistemáticos que emplea el analista durante el SDLC pueden ayudar a asegurar que el mantenimiento siempre se mantenga en el nivel mínimo necesario.

7. Implementación y evaluación del sistema

En esta última fase, el analista ayuda a implementar el sistema de información. En esta etapa hay que capacitar a los usuarios para operar el sistema. Los distribuidores se encargan de una parte de la capacitación, pero la supervisión de la capacitación es responsabilidad del analista de sistemas. Además, el analista necesita planificar un cambio sin problemas del sistema antiguo al nuevo. Este proceso incluye convertir los archivos de los formatos anteriores a los nuevos, o crear una base de datos, instalar equipos y llevar el nuevo sistema a producción.

Modelos para planificar el proceso

Existen distintos modelos para planificar el proceso que seguirá el desarrollo de software. Cada modelo tiene características que le permiten adaptarse mejor a unos tipos de proyecto que a otros.

Es posible separar los modelos en dos grandes grupos principales: las metodologías secuenciales y las metodologías ágiles.

- *Las metodologías secuenciales:* También conocidas como "tradicionales". Consisten en varias etapas bien definidas que se van desarrollando de manera ordenada, una después de la otra. Fueron las primeras metodologías utilizadas en los proyectos de desarrollo de software y hasta hoy en día son ampliamente utilizadas.
- *Las metodologías ágiles:* Se desarrollan de manera iterativa, repitiéndose etapas cortas en las que se realizan pequeñas secciones del proyecto. Dos palabras que caracterizan a un proyecto realizado mediante una metodología ágil son **interactivo** e **incremental**. Posee cinco etapas: exploración, planeación, iteraciones para la liberación de la primera versión, puesta en producción y mantenimiento. Revisémoslas a continuación:

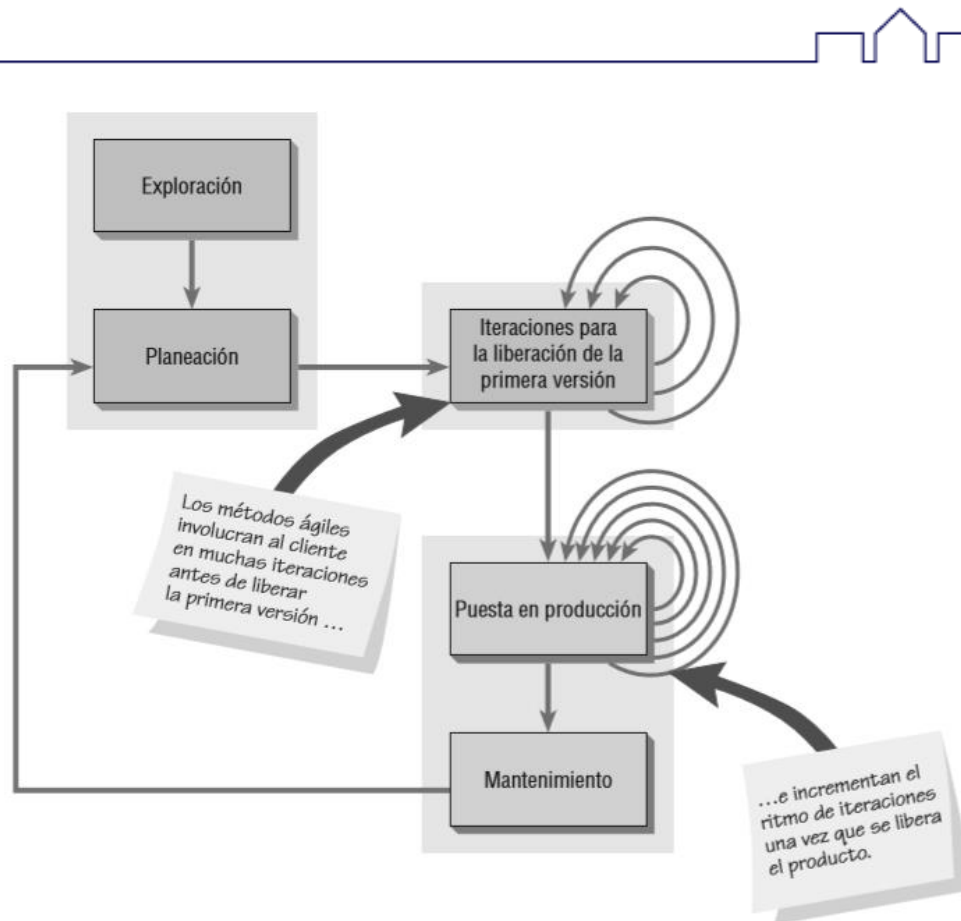


Figura 8. Las cinco etapas del proceso de desarrollo de modelado ágil muestran que las iteraciones frecuentes son esenciales para un desarrollo exitoso del sistema. (KENDALL, 2011)

Las primeras tres flechas color gris que iteran de vuelta a la caja “Iteraciones” representan los cambios incrementales creados por medio de los procesos repetidos de prueba y retroalimentación que en cierto momento conducen a un sistema estable, pero en evolución. El ritmo de iteraciones se acrecienta una vez que se libera el producto. La flecha sale de la etapa de mantenimiento y regresa a la etapa de planeación, de manera que hay un ciclo continuo de retroalimentación que incluye a los clientes y al equipo de desarrollo a medida que se ponen de acuerdo para modificar el sistema en evolución.

Las cinco etapas del proceso de desarrollo de modelado ágil muestran que las iteraciones frecuentes son esenciales para un desarrollo exitoso del sistema.



Es importante destacar que en las metodologías secuenciales el producto estará completamente terminado y será utilizable sólo cuando se terminen todas las etapas que componen el proyecto, en una metodología ágil se van situando las partes, una sobre otra, que van haciendo el proyecto cada vez más completo y funcional.

Algunos modelos secuenciales:

a) Ciclo de vida tradicional o modelo en cascada (Royce, 1970)

Se basa en la idea de producir secuencialmente tipos de productos en diferentes niveles de abstracción. Seguir este modelo en un proyecto significa transformar linealmente, todos los requerimientos iniciales en un producto cada vez más concreto.

Este es el modelo más antiguo y más utilizado. Para utilizarlo es necesario considerar que los proyectos no necesariamente siguen el flujo secuencial, hay iteraciones, por lo que es difícil para el usuario establecer explícitamente todos los requerimientos. El usuario debe ser “paciente”. Si bien es un modelo antiguo, es útil para comenzar.

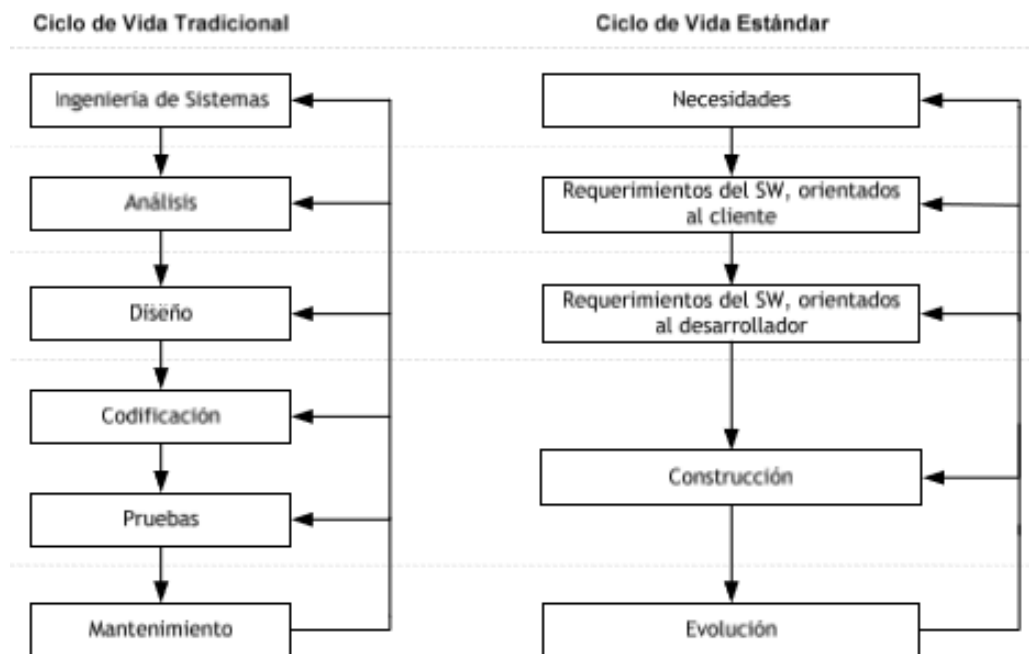


Figura 9. Ciclo de vida tradicional v/s ciclo de vida estándar. Fuente: (Bedini González, 2005)

Ejemplo de proyecto que utiliza esta metodología por sus características:

En una metodología de desarrollo tradicional, antes de empezar a construir el sistema, se intenta predecir y detallar todas las variables del proyecto. Además, el cliente no tiene la capacidad de ver el producto hasta las últimas fases de desarrollo donde el sistema ya está construido.

Todas las etapas del proyecto suelen realizarse secuencialmente. Por lo tanto, si se retrasa una de ellas, se produce un retraso en el proyecto.

La arquitectura o las ingenierías clásicas (industrial, mecánica, eléctrica, puentes y caminos, etc.) necesitan seguir este tipo de ciclos de vida en cascada o predictivos ya que necesitan un diseño previo, exhaustivo e inamovible. Por lo tanto, requieren disponer de los planos antes de empezar la fase de construcción o fabricación. Una vez construidos los cimientos de un edificio, no volverá a rediseñar el plano y no se cambiará lo construido. El software tiene diferencias muy sustanciales con respecto a estos productos físicos. Estas diferencias hacen que el proceso de construcción sea diferente.

b) Modelo en V

Una variante del modelo de cascada se denomina modelo en V. En este modelo, el proceso se realiza de forma secuencial hacia abajo y ejecuta una serie de pruebas por el lado derecho de la V para asegurar la calidad del software. A medida que el equipo de software desciende desde el lado izquierdo de la V, los requerimientos básicos del problema mejoran hacia representaciones técnicas cada vez más específicas del problema y de su solución. Una vez que se ha generado el código, el equipo sube por el costado derecho de la V y en esencia realiza una serie de pruebas que validan todos los modelos creados cuando el equipo fue descendiendo por el lado izquierdo. Este último proporciona una forma de visualizar del modo de aplicación de las acciones de verificación y validación al trabajo de ingeniería inicial.

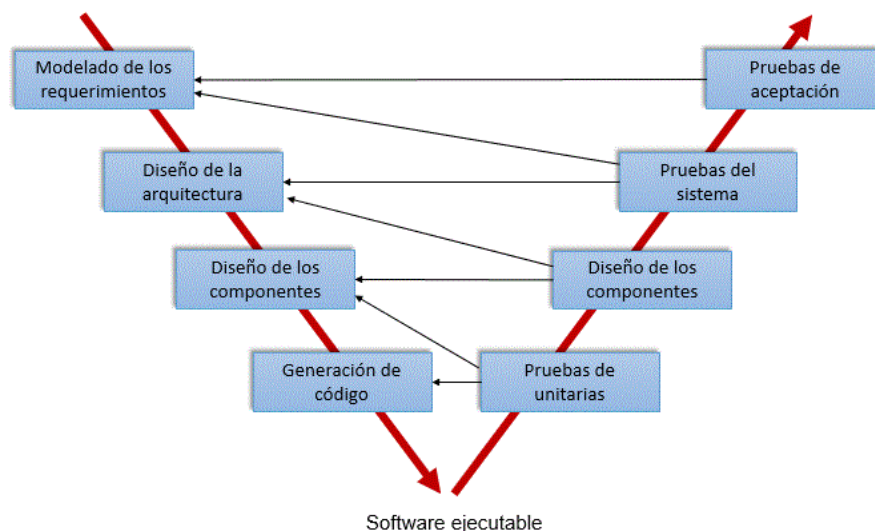


Figura 10. Modelo en V. Fuente: (LCC, s.f.)f.

Ejemplo de proyecto que utiliza esta metodología por sus características:

El Método “V” de evaluación-planeación es a la vez una concepción y un procedimiento para atender problemáticas complejas en devenir, como es el caso de los procesos educativos en las instituciones de educación superior.

La concepción que subyace al Método “V” de evaluación-planeación es que los procesos educativos son altamente complejos por la diversidad de elementos que los integran, que están organizados en múltiples niveles, dimensiones y ámbitos, que interactúan e inciden entre sí de manera endógena y con otra serie de factores exógenos; que están en devenir y son susceptibles tanto de ser transformados como de transformar.

Consecuentemente, los procedimientos que incorpora este tipo de análisis permiten abordar la problemática en sus niveles, ámbitos y elementos, y dar un seguimiento que dé cuenta de los avances y transformaciones que va expresando el sistema.

El Método “V” de Evaluación - Planeación pondera la calidad, eficiencia y superación del objeto evaluado en diferentes fases y etapas de su trayectoria.

Algunos modelos evolutivos:

a) Modelo espiral (definido por primera vez por Barry Boehm en 1986)

Este modelo se basa en un acercamiento orientado a determinar el riesgo en el desarrollo del Software. Los ciclos de desarrollo iterativo se organizan en forma de espiral, donde los ciclos interiores representan un análisis temprano, La construcción de prototipo y los ciclos exteriores representan el ciclo de vida clásico. Esta técnica se combina con análisis de riesgo durante cada ciclo. El modelo se acerca paulatinamente a un sistema completo. Es adecuado para proyectos de gran escala.

Cuenta con 4 actividades principales:

- Planificación: Objetivos, alternativas, restricciones.
- Análisis de riesgo: Análisis de alternativas y riesgos.
- Implementación: Desarrollo de producto.
- Evaluación del cliente

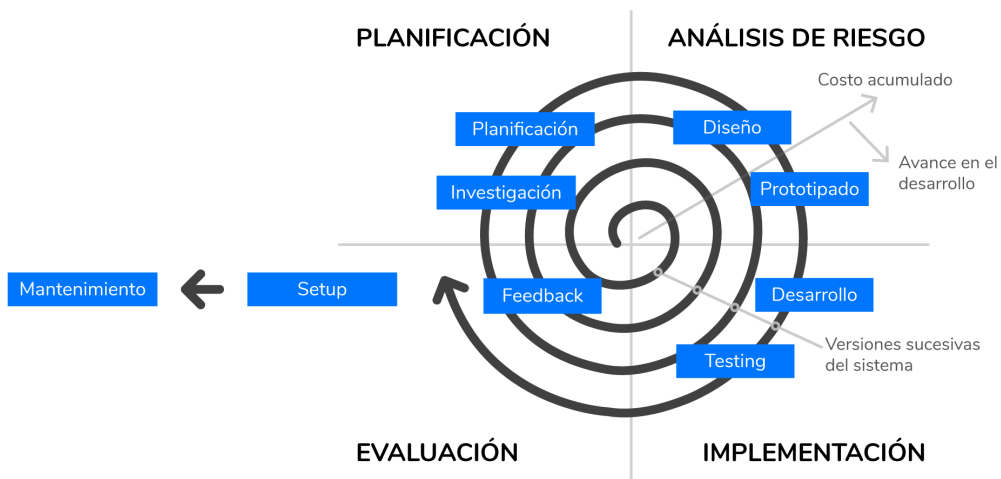
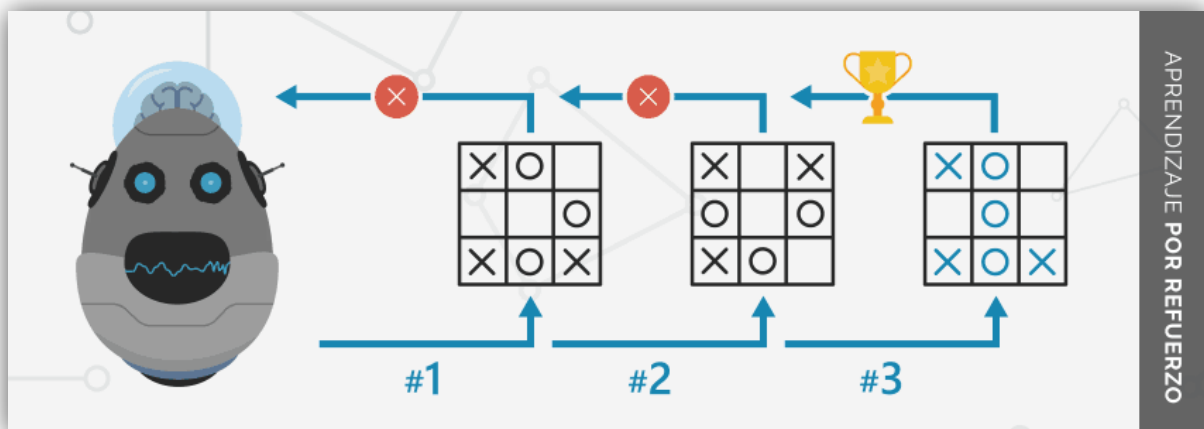


Figura 11. Modelo espiral Fuente: (aspgems, s.f.)

Ejemplo de proyecto que utiliza esta metodología por sus características

Aprendizaje por refuerzo (RL, *Reinforcement Learning*)

El aprendizaje por refuerzo consiste en la iteración constante y basada en “prueba y error” que una máquina es capaz de realizar en tiempo récord ante determinadas condiciones o entorno dado (por ejemplo, las reglas de un juego) y con un objetivo específico llamado “recompensa” (un ejemplo clásico es el de vencer en una partida de ajedrez). De esta forma se pueden obtener resultados, patrones, correlaciones, caminos y conclusiones basadas en experiencia previa generada por la propia máquina. Un ejemplo de este modelo de aprendizaje ha sido la IA ajedrecista AlphaZero de DeepMind.



Fuente: (auraportal.com, s.f.)

b) Metodología Scrum (identificado y definido por Ikujiro Nonaka y Takeuchi a principios de los 80)

Scrum es un modelo para trabajo colaborativo a partir de iteraciones. Es una metodología ágil, por lo que su objetivo será controlar y planificar proyectos con un gran volumen de cambios de última hora, donde el nivel de incertidumbre sea alto. Se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega.

En Scrum, un proyecto se realiza en ciclos de tiempo reducidos y de duración fija, con iteraciones que duran aproximadamente 2 semanas, sin embargo, en algunos equipos consideran 3 e incluso 4 semanas, periodo máximo de retroalimentación de producto real y reflexión. Cada iteración debe entregar un resultado completo., Se va revisando el trabajo validado de la semana anterior debido a que los requerimientos van cambiando a corto plazo, sobre esto se priorizan y planifican las actividades en las que se invertirán los recursos en la próxima iteración.

En cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint³. El trabajo realizado dentro ellos se adapta al problema en cuestión y se define en tiempo real por parte del equipo Scrum, tiempo que generalmente se modifica.

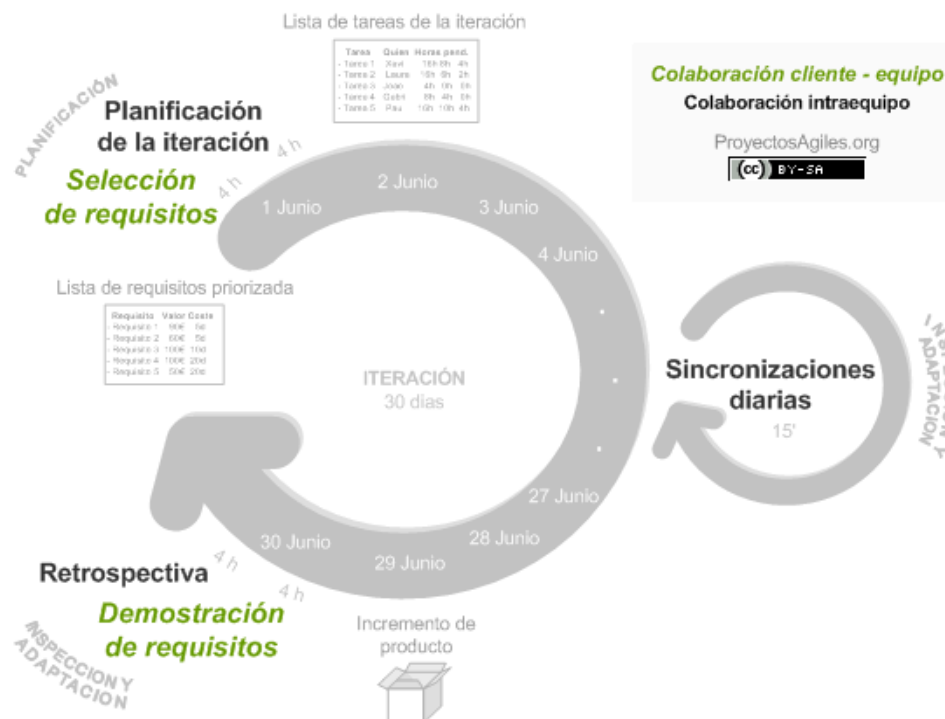



Figura 12. Flujo general del proceso Scrum. Fuente: (scrumentespanol, s.f.)

³ Sprint: Es el nombre que va a recibir cada uno de los ciclos o iteraciones dentro de un proyecto Scrum. La cantidad de ellos que se requiere para cada actividad estructural es diferente dependiendo de la complejidad y el tamaño del proyecto.



Ejemplo de proyecto que utiliza esta metodología por sus características:

Proyectos encargados del desarrollo de Videojuegos (VJ).


Todo grupo, para llevar a cabo un buen proceso de desarrollo, debe hacer uso de buenas prácticas, por lo que para el proceso de desarrollo de software de EVIMA se escogió la metodología ágil SCRUM por las características que la contempla, las del grupo de desarrollo y porque es un modelo de referencia que define un conjunto de prácticas y roles, que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Estos son los principales requisitos que se deben llevar a cabo cuando es utilizada esta metodología ágil:

- La cultura de la empresa está basada en trabajo en equipo, delegación, creatividad y mejora continua.
- El compromiso del cliente sigue la dirección de los resultados del proyecto, gestión del ROI y disponibilidad para poder colaborar.
- El compromiso de la Dirección de la organización es para resolver problemas endémicos y realizar cambios organizativos, formando equipos autos gestionados y multidisciplinarios y fomentando una cultura de gestión basada en la colaboración y en la facilitación llevada a cabo por líderes al servicio del equipo.
- Debe existir compromiso conjunto y colaboración de los miembros del equipo.
- La relación entre el proveedor y el cliente está basada en ganar-ganar, colaboración y transparencia.
- Tiene que haber facilidad para realizar cambios en el proyecto.
- El tamaño de cada equipo debe estar entre 5 y 9 personas (con técnicas específicas de planificación y coordinación cuando varios equipos trabajan en el mismo proyecto).
- El equipo ha de estar trabajando en un mismo espacio común para maximizar la comunicación.
- Tiene que existir dedicación del equipo a tiempo completo.
- Es de vital importancia la Estabilidad de los miembros del equipo.

Cada metodología tiene sus roles, y los principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director o jefe de proyecto, el ProductOwner, que representa a los stakeholders (interesados externos o internos), y el Team que incluye a los desarrolladores.

Usar SCRUM como metodología de desarrollo da un gran grupo de beneficios:

-
- 
- Flexibilidad a cambios. Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
 - Reducción del Time to Market. El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
 - Mayor calidad del software. El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
 - Mayor productividad. Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
 - Maximiza el retorno de la inversión (ROI). Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
 - Predicciones de tiempos. A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
 - Reducción de riesgos. El hecho de llevar a cabo las funcionalidades de mayor valor en primer lugar y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.

Por todo lo antes mencionado se puede apreciar que Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Un buen resumen de los conceptos se aprecia en la siguiente imagen:

Predictivos	Iterativos	Incrementales	Ágiles
Los requisitos son definidos por adelantado antes de que comience el desarrollo	Los requisitos pueden ser elaborados a intervalos periódicos durante la entrega	Los requisitos se elaboran con frecuencia durante la entrega	
Entregar planes para el eventual entregable. Posteriormente, entregar solo un único producto final al final de la línea de tiempo del proyecto	La entrega puede ser dividida en subconjuntos del producto global	La entrega ocurre frecuentemente con subconjuntos del producto global valorados por el cliente	
El cambio es restringido tanto como sea posible	El cambio es incorporado a intervalos periódicos	El cambio es incorporado en tiempo real durante la entrega	
Los interesados clave son involucrados en hitos específicos	Los interesados clave son involucrados periódicamente	Los interesados clave son involucrados continuamente	
El riesgo y los costos son controlados mediante una planificación detallada de las consideraciones que mayormente se conocen	El riesgo y los costos son controlados mediante la elaboración progresiva de los planes con nueva información	El riesgo y los costos son controlados a medida que surgen los requisitos y limitaciones	

Características de los ciclos de vida (Fuente PMBOK 6 Ed.)

¡Piensa rápido!



Se solicita el desarrollo de un Software para gestión de las subvenciones agrarias de un país. Para ello han indicado que utilizarán como modelo de desarrollo el *modelo clásico*.

¿Es el mejor modelo para el proyecto? ¿Cuál elegirías tú? Justifica tu respuesta.



Calidad dentro del desarrollo de un sistema de información

El ciclo de vida básico clásico de un sistema consta de 7 etapas, en cada una de ellas es posible considerar elementos que van a permitir evaluar la calidad del software, dentro de ellos se encuentran:

1. Precisar los elementos de un sistema de gestión de calidad.
2. Documentar el sistema de calidad.
3. Soporte del control y garantía de la calidad.
4. Determinar formas de revisión para el sistema de gestión de calidad.
5. Determinar formas de control.
6. Establecer procedimientos para la corrección.

Garantía de la calidad del software


La garantía de la calidad corresponde a un grupo de tareas de auditoría e información que evalúan la efectividad del sistema, es decir, el nivel de certeza de que el sistema no presenta vulnerabilidades ya sea que hayan sido diseñadas de manera intencional o se hayan incorporado accidentalmente en cualquier etapa de su ciclo de vida, además que el sistema funcione como se tiene previsto. En el aseguramiento de la calidad se le presentan al gestor los datos necesarios para que pueda evaluarlos. Es responsabilidad de él estudiarlos y utilizar los recursos necesarios para resolver los conflictos para lograr la calidad.

Calidad de software

La calidad del software se puede observar en una característica o atributo. Se refiere a características que se pueden medir, es decir información que se puede comparar al conocer patrones definidos, como tamaño, color y elasticidad.

En las características medibles se pueden encontrar dos tipos de calidad:

- a) Calidad de diseño: Es la capacidad de realizar un diseño que se ajuste correctamente a las necesidades del cliente y a las cualidades de la calidad.
- b) Calidad de concordancia: Corresponde al grado de cumplimiento de las especificaciones y aspectos indicados durante la "Calidad de diseño", el grado de cumplimiento es directamente proporcional a la calidad de concordancia, es decir, entre mayor sea el grado de cumplimiento, la calidad de concordancia será mucho mayor porque se estará cumpliendo con lo especificado.



El control de calidad involucra una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del proceso de desarrollo de un sistema, por medio del cual se garantiza que el producto final cumpla con los requisitos que se han establecido inicialmente.

Costo de la calidad

Incluye todos los costos que se generan o que se requieren en el desarrollo de las actividades relacionadas con la calidad, los costos de la calidad se dividen en:

- a) Costos asociados con prevención.
- b) Evaluación y fallas.



Modelo de negocios

Un modelo de negocio es una herramienta anterior al plan de negocio que define claramente qué se va a ofrecer al mercado, cómo se hará, quién será el público objetivo, cómo se va a vender y cuál será la forma para generar ingresos.

Los elementos básicos que debe contener un modelo de negocio son:

- Cuáles son y cómo se va a relacionar con los clientes.
- Cuál es la propuesta de valor⁴.
- Cómo va a desarrollar el producto o servicio.
- Cuánto va a costar esto (inversión inicial).

El modelo de negocio abarca un amplio conjunto de factores, como definir las características de los productos que se venderán, anticipar la manera de llegar a los clientes, las formas de promoción publicitaria que se utilizarán, el público objetivo al que irá dirigido el producto, etc.

En definitiva, es plasmar en un documento cómo se va a crear, desarrollar y capturar valor. Una pequeña visión de lo que puede llegar a ser y los diferentes aspectos sobre los que se va a construir la empresa. Sería como los pilares de un edificio, siendo este el negocio y los pilares el propio modelo.

Modelo de negocio Canvas

Es una herramienta que tiene distintos apartados que se encargan de cubrir todos los aspectos básicos de un negocio, desde los segmentos de clientes hasta los socios claves y la estructura de costes. En general, sigue la definición de modelo de negocio y busca reflejar en un solo lugar cómo se crea, entrega y captura ese valor de la puesta en marcha (*start up*).

Segmentos de clientes

Segmentación de mercado o grupo de personas a los que se va a vender el producto o servicio. Es posible clasificar a los clientes por las necesidades, canales, relaciones u ofertas. Algunos ejemplos de segmentos serían el mercado de masas (muy amplios), los nichos de mercado (muy

⁴ Es el factor que hace que un cliente se decida por una u otra empresa y lo que busca es solucionar un problema o satisfacer una necesidad del cliente de manera innovadora.



específicos), los diversificados (distintos públicos muy distintos) o los multi-segmentos (que dependen de varios clientes a la vez).

Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Para quién estamos creando valor?• ¿Quiénes son nuestros clientes más importantes?	<ul style="list-style-type: none">• Necesidades de cada segmento de clientes.• Diferentes canales de distribución.• Diferentes tipos de relación.• Diferentes márgenes de beneficios.

Propuesta de valor

Corresponden a las características y beneficios que se encargan de crear valor para cada uno de esos segmentos. En esta parte se debe explicar qué es lo que se ofrece a los clientes y por qué van a comprarlo. Algunas características de esta propuesta podrían ser la novedad, el rendimiento, la personalización, el diseño o el precio.

Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Qué valor entregamos al cliente?• ¿Cuál de los problemas de nuestro cliente solucionamos?• ¿Qué necesidad estamos satisfaciendo?	Novedad, diseño, precio, marca, mejoras en productos o servicios y una propuesta de valor diferente para cada segmento.

Canales

Medios a través de los cuales se comunica la propuesta de valor al cliente. Pueden ser canales propios (de los socios) o externos (directos o indirectos) y están divididos en 5 fases (notoriedad, evaluación, compra, entrega y postventa).



Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Por qué canales prefieren mis clientes ser contactados?• ¿Cómo estamos contactando con ellos ahora?• ¿Cuál es el canal que mejor funciona?• ¿Cuál es canal más eficiente en coste?• ¿Cómo lo integramos en la rutina de los clientes?	<ul style="list-style-type: none">• Comunicación, distribución y canales de ventas.• Ayudar al cliente a que tome conciencia de nuestra propuesta de valor.• Ayudar al cliente a evaluar la propuesta de valor.• Facilitar la compra al cliente.• Servicio post venta.

Relación con el cliente

Tipo de relación entre la empresa emergente (*start up*) y el cliente. Puede ser asistencia personal, *self-service* o automatizado (mezcla de ambas).

Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Qué tipos de relación queremos establecer y mantener con nuestro cliente?• ¿Cuánto cuesta esta relación?• ¿Cómo se integra en nuestro modelo negocios?	<ul style="list-style-type: none">• Relación personal o relaciones humanas.• Relación personal con dedicación.• Self-service, el cliente se sirve solo.• Servicios automatizados, relación con grabaciones, maquinas, S.O.• Communities, los clientes se relacionan entre ellos.• Concreciones.

Fuente de ingresos

¿De dónde va a llegar el dinero? ¿Cómo se va a generar el beneficio? Algunos modelos de fuente de ingreso podrían ser la venta directa en un único pago, el pago por uso o la suscripción.



Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Por cuál propuesta de valor están dispuestos a pagar nuestros clientes?• ¿Por qué están pagando actualmente?• ¿Cómo están pagando actualmente?• ¿Cómo preferirían pagar?• ¿Cuál es el porcentaje de ingresos de cada línea de ingreso respecto a los ingresos totales?	<ul style="list-style-type: none">• Ingresos por cada segmento de cliente.• Pagos de una vez.• Pagos recurrentes, por servicio.• Pagos de servicios post venta.

Recursos clave

Corresponden a los recursos más importantes para que todo lo anterior funcione. Pueden ser físicos (vehículos, edificios, ...), intelectuales (patentes, copyrights, ...), humanos (expertos clave, empleados muy valorados, ...) o financieros (efectivo, crédito, ...).

Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Qué recursos clave requiere nuestra propuesta de valor, nuestros canales de distribución, la relación con el cliente y las fuentes de ingreso?	<ul style="list-style-type: none">• Recursos propios, alquilados o comprados a otros.• Recursos físicos, intelectuales, humanos, financieros.• Marcas, patentes, etc.

Actividades clave

Si hay recursos clave, también tiene que haber actividades clave ¿Cuáles son las actividades sin las que, en su ausencia, el negocio fracasaría? ¿Son de producción, de solución a problemas individuales o de una plataforma a través de la que funciona toda la empresa emergente (*start up*)?



Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Qué actividades clave intervienen en nuestra propuesta de valor?• ¿Qué actividades clave requieren nuestros canales de distribución?• ¿Qué actividades clave requiere nuestra relación con los clientes?• ¿Qué actividades clave se requieren para nuestra fuente de ingresos?	<ul style="list-style-type: none">• Producir• Atender clientes• Solucionar problemas• Plataforma web, networking

Socios clave

Son aquellos colaboradores clave para que el negocio inicie y funcione ¿Y por qué se buscan estos socios clave? Para optimizar los recursos (contratar proveedores), reducir riesgos con alianzas estratégicas y adquirir recursos y actividades que no existen en la propia empresa emergente (*start up*).

Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Quiénes son nuestros socios?• ¿Quiénes son nuestros proveedores?• ¿Cuáles son los recursos que estamos adquiriendo de los socios?• ¿Cuáles son las actividades que hacen los socios?	<ul style="list-style-type: none">• Descripción del network de proveedores y socios, que hacen que el modelo de negocios funcione y se optimice, se reduzcan riesgos y se adquieran recursos clave.• Optimizar economías de escala⁵.• Reducir riesgos e incertidumbres.• Adquirir un particular activo o recurso.

⁵ Se refiere al poder que tiene una empresa cuando alcanza un nivel óptimo de producción para ir produciendo más a menor costo, esto quiere decir que a medida que la producción en una empresa aumenta, sus costos por unidad producida disminuyen. Entre más produce, menos le cuesta producir cada unidad.

El clásico desglose de los gastos que va a tener el modelo de negocio. Se incluyen los costos fijos, variables, las economías de escala para reducir costos y todo lo relacionado con el gasto.

Preguntas orientadoras	Permiten identificar, por ejemplo:
<ul style="list-style-type: none">• ¿Cuáles son los costos más importantes de nuestro modelo de negocio?• ¿Cuáles son los recursos más caros?• ¿Cuáles son las actividades más caras?	<ul style="list-style-type: none">• La descripción de todos los costos en los que incurre la empresa para desarrollar su modelo: (cost driven⁶, automatizado, simplificado).• Value driven⁷ o valor para el cliente.• Costos fijos y variables.• Economías de escalas.

Las descripciones de los conceptos de modelo de negocio permitirán comprender las reglas en un sistema de información. Evitará al profesional que efectúe el proceso de toma de requerimientos omitir etapas del proceso que tenga una organización en su estructura comercial. Le permitirá, por su parte, efectuar un análisis correcto y real de los actores involucrados, y describir los alcances potenciales del sistema de información desarrollar.

⁶ Son aquellos motivos que causan los costos producidos en todas y cada una de las actividades. Este modelo de negocio se centra en minimizar los costos.

⁷ Con un menor grado de preocupación por el costo, este modelo de negocio se centra en la creación de valor para sus productos y servicios.



Ejemplo de aplicación del modelo Canvas

Lee atentamente el ejemplo para ver paso a paso la aplicación del modelo Canvas.

Monté una Cafetería cuya atracción son las tazas con Impresiones 3D. Conoce mi experiencia.

Propuesta de valor. Paso 1

Se trata de brindar una experiencia personalizada en el diseño e impresión 3D de la taza, en la cual el consumidor va a tomar el café. Incluso, el cliente podrá llevársela a su casa.

Segmento de clientes. Paso 2

En mi cafetería, los clientes son particulares. Algunos desean personalizar su experiencia en la degustación del café o hacer un regalo diferente a familiares y amigos.

Hay clientes que son empresas que quieren realizar reuniones de negocio y, como valor agregado, quieren obsequiar la taza a los asistentes de la sesión.

Un tercer segmento de clientes serían personas que deseen celebrar los cumpleaños a sus hijos.

Al final de la fiesta, en vez de regalar caramelos (azúcar y más azúcar), obsequiarán las tazas que los niños han escogido previamente al llegar a la fiesta.

Canales de distribución y comunicación. Paso 3

Mi cafetería estará ubicada a pie de calle para que los clientes tengan facilidad en el acceso. Tendré mi propia página web, para que los clientes que deseen diseñar y adquirir su taza, previamente, lo realicen en línea.

Cuando lleguen a mi local, solo pidan su café y no tengan que esperar la impresión. Tendré cuentas en las redes sociales.

También podrán hacerlo en la tienda, si quieren hacer su experiencia «en vivo» (ver la impresora mientras realiza su trabajo, por ejemplo). Las compañías que se encarguen de organizar fiestas infantiles serán un buen canal de comunicación. Buscaré promoción a través de los medios audiovisuales. Entregaré ofertas de servicios en los Departamentos de Compras y Relaciones Públicas de las empresas. Invitaré a otros emprendedores a hacer negocios en mi cafetería para que se luzcan con los clientes o influencers regalando una taza.

Relación con los clientes. Paso 4

Ofertas especiales los días festivos (Madre, Padre, Navidad, del Amor, Cumpleaños etc.). El cliente se lleva la taza de recuerdo. Posibilidad de celebrar cumpleaños de manera original. Cerrar negocios de manera diferente.



Fuentes de ingreso. Paso 5

En mi local podrán cancelar con cualquier forma de pago. bancaria en línea. El cuánto no puedo decirlo ahora, porque tendría que calcular mis costos y ganancias.

Cuando es un cliente particular, así será el pago. Cuando es un cliente corporativo, pasaré mi factura post-evento. Cuando es celebración, se realizará un contrato de servicio.

Recursos claves. Paso 6

Requiero de personal que atienda la cafetería. Local. Vitrinas, mobiliario, caja registradora, utensilios, máquinas de café. Insumos para hacer café y otros alimentos que acompañen al aromático. Máquinas 3D para la impresión de las tazas. Insumos para la impresión (materia prima). Plataforma tecnológica para página web y pago en línea. Mantenimiento de los equipos y mobiliarios. Seguridad.

Actividades claves. Paso 7

Atención al cliente. Asesoría en el diseño de las tazas. Servicio de cafetería y catering para fiestas y reuniones de negocio.

Socios claves. Paso 8

Compañías que ofrezcan servicios de catering para empresas. Empresas que organicen fiestas de cumpleaños. Centros de impresión 3D (por si la demanda supera la capacidad de producción). Servicio de transporte rápido (motos o bicicletas) para entrega de tazas que se hayan elaborado en otros centros de impresión.

Estructura de costos. Paso 9

Adquisición de todos los activos. Pago de personal. Pago de servicios públicos e impuestos. Adquisición de materia prima.

Una vez finalizado el proceso de 9 pasos, el modelo Canvas de la Cafetería es el siguiente:



Modelo canvas ejercicio cafetería. Fuente: (www.emprender-facil.com, s.f.)

Arquitectura de un sistema de base de datos

Mencionamos anteriormente que una arquitectura de tres niveles se enfoca en separar los programas de aplicación de la base de datos física.

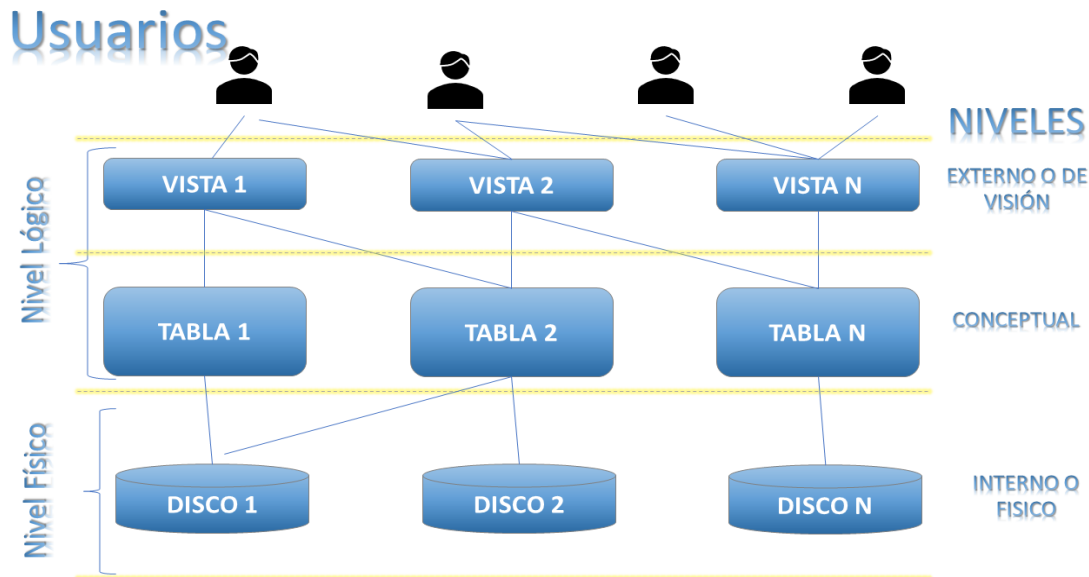


Figura 13. Arquitectura. Elaboración propia, Aida Villamar, 2020.

Por medio de la imagen se pueden apreciar los tres niveles: físico que incluye los discos (donde se almacena la información), conceptual (las tablas dentro de las bases de datos, la estructura en sí) y externo (vistas, que corresponde a la forma en que son presentados los datos al usuario, por ejemplo, las pantallas), sobre las vistas en la imagen se encuentran los usuarios, que son quienes interactúan directamente con el sistema.

Objetivos de los DBMS⁸

Un sistema gestor de base de datos corresponde a programas que permiten almacenar, modificar y extraer la información en una base de datos. Los usuarios pueden acceder a la información utilizando herramientas concretas de consulta y de creación de informes, o bien por medio de aplicaciones.

Los objetivos de los SGBD son:

⁸ Data Base Management System en español SGBD Sistema gestor de base de datos.



- Evitar redundancia de datos: Hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares.
- Evitar inconsistencia de datos: Consiste en que no todas las copias redundantes contienen la misma información. Sólo se produce cuando existe redundancia de datos
- Permitir en todo momento el acceso a los datos.
- Ante el acceso concurrente⁹, evitar anomalías.
- Seguridad, restricción ante accesos no autorizados.
- Integridad de datos: Se refiere la correctitud y completitud de la información en una base de datos
- Backup (respaldos): Una copia de seguridad, copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida.

Independencia lógica y física de los datos

Para comenzar, se verá la independencia lógica y física de los datos, por ejemplo, si se realiza alguna modificación de la estructura de la base de datos (ej.: ampliar o reducir la base de datos), las vistas que poseen las aplicaciones (esquemas externos o programas de aplicación) no tendrían que sufrir alteraciones ni verse afectado eso es Independencia Lógica y si se cambia un disco duro o el sistema operativo, el esquema lógico debe permanecer invariable, se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento esto es Independencia Física.

La **independencia física** de datos corresponde a la **capacidad de modificar el esquema físico** sin provocar que deban volverse a escribir los programas de aplicación, es decir, no debe verse afectado el esquema conceptual.

En la práctica esto significa que, aunque se agreguen o cambien discos u otro hardware, o se modifica el sistema operativo u otros cambios relacionados con la física de la base de datos, el esquema conceptual permanece inalterable.

⁹ Acceso concurrente: Un sistema que permita a varias estaciones de trabajo modificar en forma simultánea una misma base de datos, debe tomar precauciones para evitar operaciones concurrentes sobre un mismo registro. Bloquea el registro usado en ese momento como medida de seguridad.

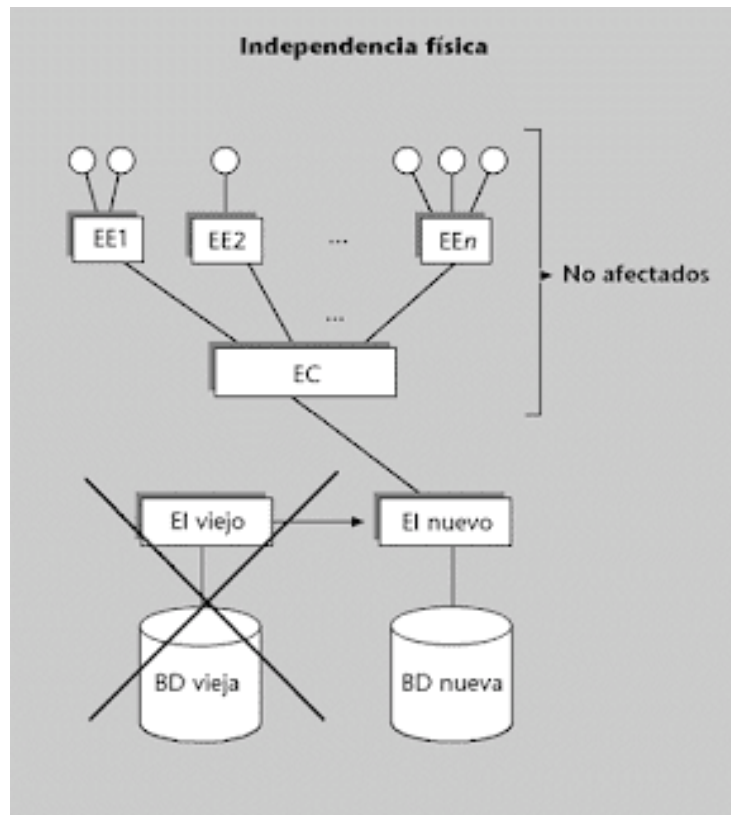


Figura 14. Independencia física. Fuente:-(Diseño de bases de datos, s.f.)

La **independencia lógica** de datos, por su parte, es la **capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación**.

Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos.

Son más difíciles de lograr que las independencias físicas, ya que los programas de aplicación son fuertemente dependientes de la estructura lógica de los datos a los que acceden.

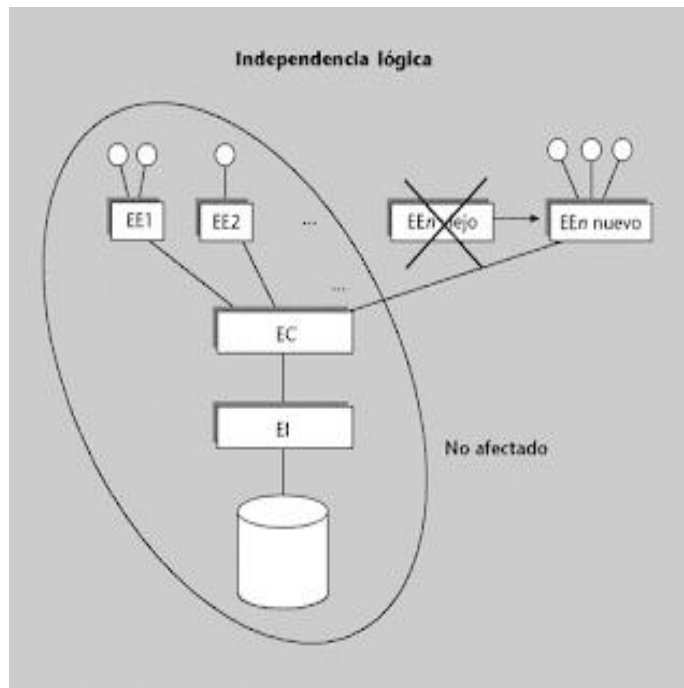


Figura 15. Independencia lógica. Fuente: (Diseño de bases de datos, s.f.)

Se tiene independencia lógica cuando los usuarios no se ven afectados por los cambios realizados en el nivel lógico

Ahora que hemos revisado algunos conceptos generales, veremos todo lo relacionado con la captura requerimientos de clientes reales.

Requerimientos de clientes

Cada uno de los modelos del proceso de desarrollo de software contiene actividades que apuntan a la toma de requerimientos. Por lo tanto, entender el propósito y la función del sistema se inicia con un cuidadoso examen de los requerimientos.

Definición de requerimiento

Cuando un cliente requiere que se desarrolle un sistema posee algunas nociones de lo que debe hacer. Por esto, los sistemas basados en software poseen un propósito, generalmente expresado con algo que el sistema debe hacer.

Los requerimientos son declaraciones que identifican una característica, un atributo o descripción que puede representar una capacidad o condición que necesita cumplir un sistema (o un software) para que tenga valor y utilidad para el usuario, todo esto, con la finalidad de satisfacer el propósito del sistema. Es decir, los requerimientos son lo que los clientes/usuarios esperan que haga el sistema.

Los analistas deben comprender el problema de los usuarios considerando su cultura y lenguaje, para posteriormente construir el sistema que resuelve sus necesidades. En sí, el objetivo del análisis de requerimientos es resolver el problema.



Requerimiento v/s diseño

- Los requerimientos definen el **qué (el problema)** del sistema.
- El diseño define el **cómo (la solución)**.

Durante el análisis de requerimientos no se consideran descripciones específicas de la implementación como requerimientos, a menos que el cliente lo pida. Por ejemplo, bases de datos específicas, lenguajes de programación, etc.

Los requerimientos, por lo tanto, deben centrarse en el cliente/usuario y el problema.



Importancia de los requerimientos:

Muchos proyectos de Software fracasan porque no se realiza un estudio previo de los requisitos del usuario, como tampoco una definición completa del alcance del proyecto.

Si no es realizado el modelado del negocio antes de desarrollar el software, significa que el analista no se ha involucrado en el problema. Aunque tenga claro que el sistema debe desarrollarse para dar soporte a los procesos de la organización, si no se involucra en la problemática, se arriesga a que los requisitos identificados no correspondan a las necesidades para lo que se debe crear.

Otro problema conocido corresponde al no involucrar al usuario de manera activa en el desarrollo del producto. **Una buena práctica sería modelar junto al cliente.**

Requerimientos incompletos y el cambio constante de los requerimientos establecidos, son otros factores que llevan los sistemas al fracaso.

Por lo tanto, es muy importante que no exista carencia de requerimientos bien definidos evitando así esta lista de problemas:

- ✗ Se realizan estimaciones que no son realistas.
- ✗ Se emplean de manera incorrecta herramientas de planeación.
- ✗ No es posible realizar revisiones periódicas del progreso en base a las especificaciones.
- ✗ La arquitectura, el diseño y el desarrollo del software estarán carentes de una base firme.
- ✗ Las pruebas se basarán en supuestos, no en lo que el usuario requiere.
- ✗ No es posible controlar el crecimiento de los requerimientos.

Entonces, se puede decir que los requerimientos son importantes debido a que son el hilo conductor de todo desarrollo de software.

Obtener requerimientos de calidad indica que el trabajo realizado culminará con éxito, esto se debe a dos factores:

1. El uso adecuado de las técnicas de captura de requerimientos con los clientes.
2. Las experiencias de los analistas del proyecto.

Esto ocurre porque la experiencia de trabajo en el rol de analista de sistema le permite al equipo de Analistas del Proyecto definir qué técnicas utilizarán a la hora de la entrevista con el cliente,



debido a que los clientes no entienden el lenguaje informático, es por eso que se debe considerar qué lenguaje se va a utilizar a la hora de entrevistarse con el cliente.

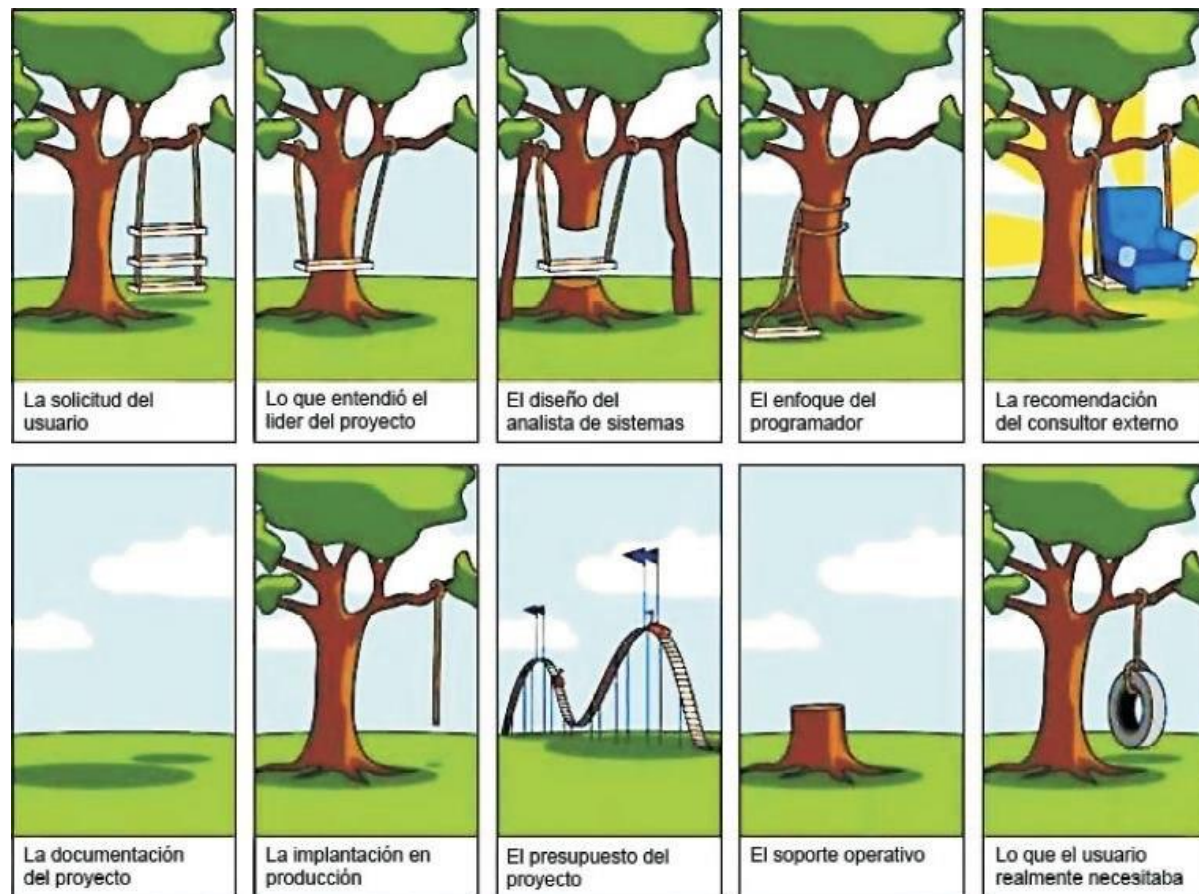


Figura 16. Los Requerimientos y su importancia en el desarrollo del Software. Fuente: (itsoftware, s.f.)

Documentos de requerimientos

Existen dos documentos que emanan del análisis de requerimientos:

-
- *Definición de requerimientos:* Documento que debe redactarse en términos que el cliente pueda comprender. Es decir, este documento es un listado que incluye todo aquello que el cliente espera que realice el sistema propuesto. El Documento es escrito en conjunto por el cliente y el desarrollador.
 - *Especificación de requerimientos:* Documento que reitera la definición de los requerimientos en los términos técnicos adecuados para el desarrollador del diseño de un sistema. Es la contrapartida técnica al documento de definición de requerimientos y es escrito por los analistas de requerimientos.

A veces solo un documento puede servir para ambos propósitos, lo que lleva a un entendimiento común entre clientes, analistas de requerimientos y diseñadores. Pero generalmente se requieren ambos documentos.

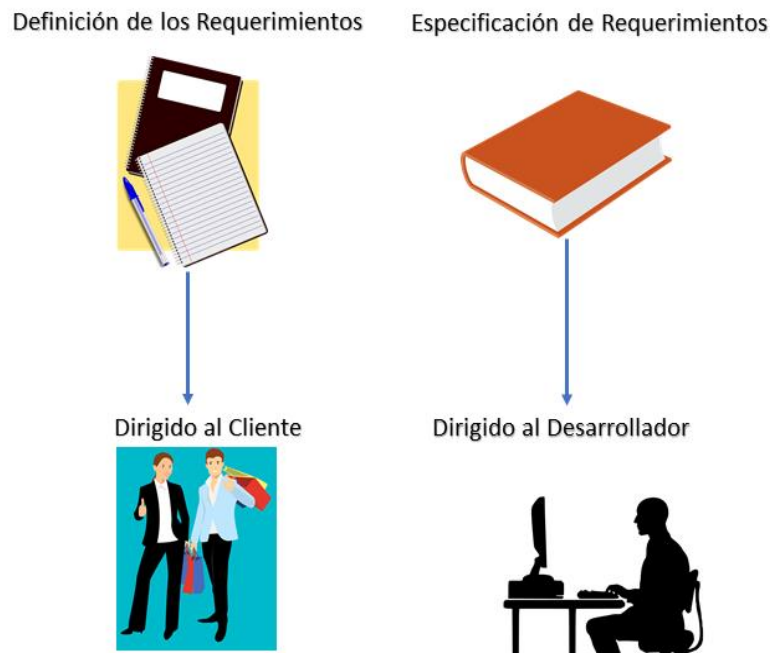


Figura 17. Tipos de documentos de requerimientos. Fuente: Elaboración propia, Aida Villamar, 2020.

Es muy importante que, al utilizar estos documentos, exista una relación directa entre cada requerimiento del documento de definición y los documentos en la especificación. Esto para que el enfoque del cliente esté unido a la de los desarrolladores (esto se consigue gracias a la gestión de configuración).



Normalmente, la definición de los requerimientos está redactada en lenguaje natural, mientras que la especificación de los requerimientos se redacta de una forma más técnica, por ejemplo, puede definir un requerimiento que se hizo en lenguaje natural, como una serie de ecuaciones, un diagrama de flujo de datos, casos de uso, etc.

Clasificación de requerimientos

Los requerimientos se clasifican según su objetivo o a quién van dirigidos. Revísalos a continuación:

Según objetivo	Según a quién van dirigidos
<ul style="list-style-type: none">• Requerimientos funcionales.• Requerimientos no funcionales.• Requerimientos del dominio.	<ul style="list-style-type: none">• Requerimientos del usuario.• Requerimientos del sistema.

Revisaremos en detalle esta clasificación.

Según tipo de requerimiento:

a) Requerimientos funcionales


Describen la funcionalidad o los servicios que se espera que el sistema proveerá. Dependen del tipo de software, del sistema que se desarrolle y de los posibles usuarios.

Cuando se expresan como requerimientos del usuario, se definen de forma general. Cuando se expresan como requerimiento del sistema, describen con detalle la función de este, sus entradas y salidas, excepciones, etc.

b) Requerimientos no funcionales

Son los requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de este, como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

Muchos requerimientos no funcionales se refieren al sistema como un todo más que a rasgos particulares del mismo.



A menudo son más críticos que los funcionales. Mientras que un incumplimiento de un requerimiento funcional degrada el sistema, el de un requerimiento no funcional del sistema lo inutiliza.

Los requerimientos no funcionales se clasifican según su implicancia:

- Del producto: Especifican comportamiento del producto. Por ejemplo, de desempeño en la rapidez de ejecución del sistema, cuanta memoria se requiere; los de fiabilidad que fijan la tasa de fallas para el sistema sea aceptable, los de portabilidad¹⁰ y de usabilidad.
- Organizacionales: Proviene de las políticas y procedimientos que existen en la organización del cliente y del desarrollador. Por ejemplo, los estándares que se deben utilizar en los procesos, requerimientos de implementación como los lenguajes de programación o el método de diseño a utilizar.
- Externos: Cubre todos los requerimientos que se derivan de los factores externos al sistema y de su proceso de desarrollo. Por ejemplo, requerimientos de interoperabilidad, requerimientos legales, requerimientos éticos.

Un problema común con los requerimientos no funcionales es que a veces son difíciles de verificar.

Idealmente los requerimientos no funcionales se deben expresar de manera cuantitativa utilizando métricas que puedan ser probadas de forma objetiva. En la práctica, es difícil. El costo es muy alto.

c) Requerimientos del dominio

Derivan del dominio del sistema más que de las necesidades específicas del usuario.


Son importantes ya que a menudo reflejan los fundamentos del dominio de la aplicación. Si estos no se satisfacen no es posible que el sistema trabaje satisfactoriamente.

Se expresan utilizando un lenguaje específico del dominio de la aplicación que a menudo es difícil de comprender. Por ejemplo, operación para calcular desaceleración del tren, para un sistema de control de trenes.

Es importante indicar que por medio de los requerimientos se:

- Permite explicar al desarrollador cómo ha entendido lo que el cliente espera del sistema.

¹⁰ Facilidad con que un sistema software puede ser migrado entre diferentes plataformas de hardware y software.

-
- 
- Señala a los diseñadores las funcionalidades y características que tendrá el sistema resultante.
 - Indica al equipo de pruebas que verificaciones se deben llevar a cabo para probar al cliente que el sistema entregado se ajusta a lo que había solicitado.

Los requerimientos deben poseer una calidad alta para permitir una buena comprensión tanto de clientes como desarrolladores.

Con este fin, debe verificarse que los requerimientos cuenten con las características, que se obtienen de las siguientes preguntas:

- ¿Son correctos los requerimientos?, desarrollador y cliente deben revisarlos para asegurarse de que no poseen errores.
- ¿Son consistentes los requerimientos? es decir, ¿Son los requerimientos planteados no conflictivos ni ambiguos? Dos requerimientos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
- ¿Son completos los requerimientos? el conjunto de requerimientos es completo si todos los estados posibles, cambios de estado, entradas, productos, restricciones están descritos en alguno de los requerimientos.
- ¿Son realistas los requerimientos? ¿Puede el sistema realmente hacer lo que el cliente está pidiendo que haga? todos los requerimientos se deben revisar para confirmar que son factibles.
- ¿Cada requerimiento describe algo que es necesario para el cliente? los requerimientos deben ser revisados para mantener sólo aquellos que afectan directamente en la resolución del problema del cliente.
- ¿Son verificables los requerimientos? se deben preparar pruebas que demuestren que se han cumplido los requerimientos.
- ¿Son rastreables los requerimientos? ¿Es posible rastrear cada función del sistema hasta el conjunto de requerimientos que la establece? ¿Resulta fácil encontrar el conjunto de requerimientos que coinciden a un aspecto específico del sistema?

Fuentes de Requerimientos

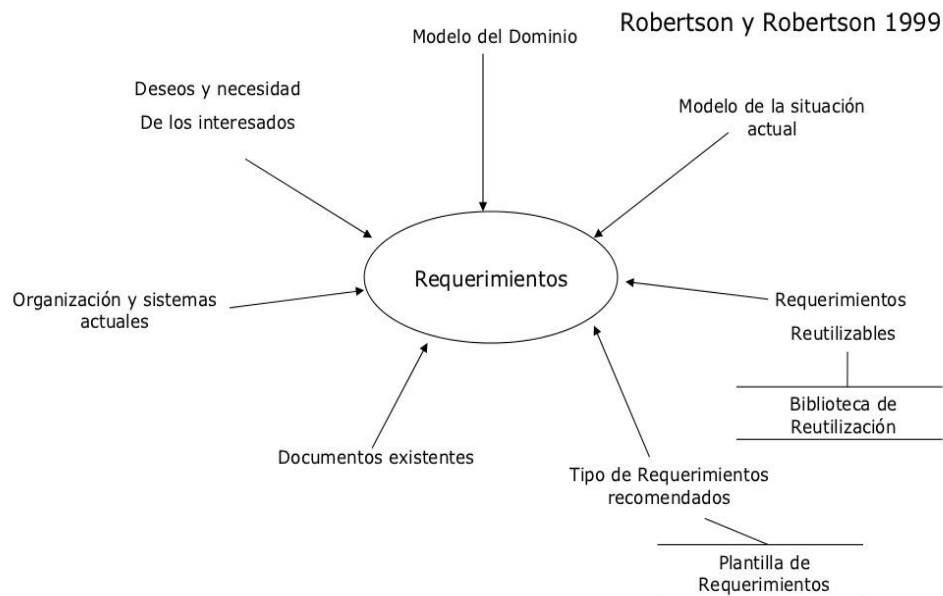


Figura 18. Fuentes de Requerimientos. Fuente: (Slideshare - Analisis De Requerimientos)

Tipo de requerimientos según a quién van dirigidos:

a) Requerimientos del usuario

Describen, utilizando un lenguaje sencillo que se complementa con diagramas para que sea comprensible por los usuarios sin conocimiento técnico, cuáles son las funcionalidades que esperan del sistema y las restricciones con las cuales este debe operar.

b) Requerimientos del sistema

Los requerimientos del sistema son descripciones extendidas de las funciones, servicios y restricciones que se utilizan como punto de partida para diseñar el nuevo sistema. Este documento tiene que definir con precisión lo que se implementará. Es posible incluirlo dentro del contrato entre el comprador del sistema y los desarrolladores del software.



Contextualización del análisis del problema (Antecedentes)

Es importante señalar cuál es la problemática que se solucionará con la ejecución del proyecto. Los antecedentes o contextualización del problema es producto de la observación directa de la problemática y, por lo tanto, se debe realizar un breve diagnóstico, descripción, análisis y argumento del problema. Puede darse el caso de que se incluya información relevante, cualitativa y cuantitativa, de la problemática por solucionar, producto de otras investigaciones.

Generalmente el problema investigado no se presenta inmediatamente. Es normal señalar algunas situaciones o sucesos que determinan un contexto o panorama general, dentro del que aparece el problema como una situación anormal o que llama la atención ya que si es resuelto podría solucionarla o aportar a la mejora.

Para contextualizar el problema se deben contestar las siguientes preguntas:

- ¿Cómo aparece el problema que se pretende solucionar?
- ¿Por qué se origina?
- ¿Quién o qué lo origina?
- ¿Cuándo se origina?
- ¿Cuáles son las causas y efectos que produce el problema?
- ¿Dónde se origina?
- ¿Qué elementos o circunstancias lo originan?



Modelos de datos

Como se indicó anteriormente, los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos. No se refiere a algo físico. Define las reglas por las cuales los datos son descritos.

Un modelo de datos permite describir de las bases de datos:

- Las estructuras de datos.
- El tipo de los datos que contiene.
- La forma en que se relacionan.

El éxito de una base de datos resulta de la combinación de dos factores:

1. La eficacia de las herramientas proporcionadas por el Sistema de Gestión de Base de Datos (SGBD).
2. El correcto diseño de la estructura de la base de datos.

Por muy potentes y adecuadas que sean dichas herramientas, el diseño es el punto clave para determinar la validez de una base de datos.

El diseño de una base de datos consiste en extraer todos los datos relevantes de un problema. Por ejemplo, saber qué datos están implicados en el proceso de facturación de una empresa que vende artículos de informática o, qué datos son necesarios para llevar el control de pruebas diagnósticas en un centro de radiológico.

Para extraer estos datos se debe realizar un análisis en profundidad del problema, para averiguar qué datos son esenciales para la base de datos y descartar los que no sean necesarios. Una vez extraídos los datos esenciales se comienzan a construir los modelos adecuados. Es decir, construir, mediante una herramienta de diseño de base de datos, un esquema que exprese con total exactitud todos los datos que el problema requiere almacenar.

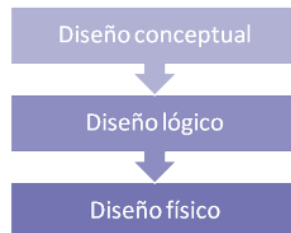
Es algo equivalente al dibujo de un plano previo a la construcción de un edificio. Además, previo al diseño es necesario realizar una primera fase denominada de análisis.

Antes de pasar a diseñar una base de datos hay que tener claro qué es lo que se quiere hacer. Para ello, tradicionalmente los informáticos se reúnen con los futuros usuarios del sistema para recopilar la información que necesitan para saber que desean dichos usuarios.

Generalmente se realiza una reunión inicial y, partir de ella, se elabora una batería de preguntas para entrevistar a los usuarios finales en una segunda reunión y obtener de ella una información detallada de lo que se espera de la base de datos.

De estas entrevistas se extrae el documento más importante del análisis: el documento de Especificación de Requisitos Software o E.R.S. A partir de dicho E.R.S, se extrae toda la información necesaria para modelar los datos.

El diseño de una base de datos es el conjunto de actividades que permite la creación de una base de datos. Esta operación se realiza en tres fases: *diseño conceptual, lógico y físico*.



Cada nivel de concreción se caracteriza por la realización de un esquema representativo de la base de datos.

Figura 19. Fases del diseño de una base de datos. Fuente: (e-ducativa, s.f.)


- **Los modelos conceptuales.**

El modelo que se utiliza en esta primera fase del diseño tiene un gran poder expresivo para poder comunicarse con el usuario que no es experto en informática y se denomina Modelo Conceptual. El modelo conceptual que utilizaremos es el Modelo Entidad/Relación. El objetivo de esta fase del diseño consiste en representar la información obtenida del usuario final y concretarla en el E.R.S. (documento de especificación de requerimientos).

Subestimar esta etapa u obviarla por completo se traduce en bases de datos inadecuadas o ineficientes con todos los problemas que conlleva.

- **Los modelos lógicos.**

En los modelos lógicos, las descripciones de los datos tienen una correspondencia sencilla con la estructura física de la base de datos. En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se



transforma el esquema conceptual en un esquema lógico. El motivo de realizar estas dos etapas es la dificultad de abstraer la estructura de una base de datos que presente cierta complejidad.

- **Los modelos físicos.**

El modelo de datos físicos representa cómo se construirá el modelo en la base de datos. Un modelo de base de datos física muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.

Estos modelos se verán en detalle a continuación.

Tipos de modelos de datos

Los modelos de datos se clasifican de acuerdo con el nivel de abstracción que presentan:

Modelo conceptual

Son los orientados a la descripción de estructuras de datos y restricciones de integridad. Se usan fundamentalmente durante la etapa de Análisis de un problema dado y están orientados a representar los elementos que intervienen en ese problema y sus relaciones.

Hacen de ellos una referencia sobre un análisis que considera no pensar en lo físico ni en los procesos y si, en algo más abstracto como la estructura de los datos y cómo se pueden relacionar con otros de su misma especie.

Los modelos conceptuales se utilizan para representar la realidad a un alto nivel de abstracción, es decir, describen los datos de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar estos datos. Mediante los modelos conceptuales se puede construir una descripción de la realidad fácil de entender.

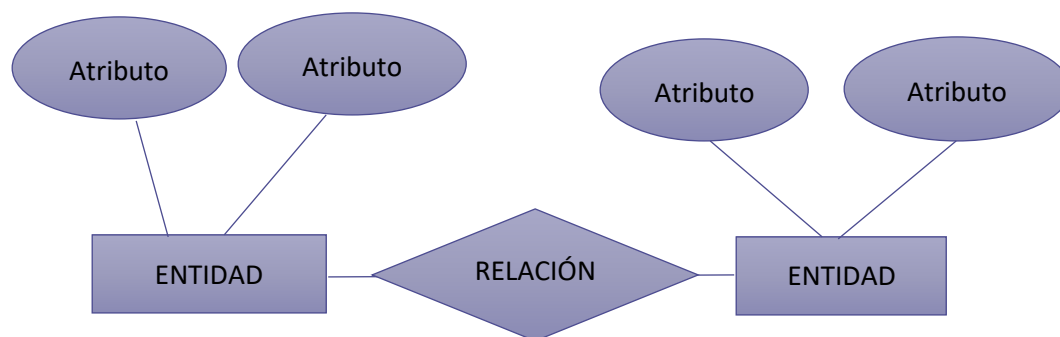


Figura 20. Modelo entidad relación (solo conceptos). Fuente: Elaboración propia, Aida Villamar, 2020.

Como se puede visualizar las notaciones utilizadas, estas entregan el set de símbolos necesarios para poder representar un requerimiento. Cabe mencionar que esta herramienta gráfica permite:

Representar el mundo (mini-mundo¹¹) como:

- Una colección de entidades.
- Relaciones entre entidades.

Para aclarar su alcance se verá el siguiente ejemplo:

Se diseñará un modelo de datos utilizando la notación anterior el cual permitirá indicar la relación existente entre las entidades **cliente**, **producto** y **proveedor**.

Desarrollo

Se debe asignar propiedades a cada entidad para poder detallar sus características. Es importante que podamos distinguirla y asociarla a un elemento común que podamos visualizar fácilmente por medio de sus atributos.

Se debe obtener los verbos que comunicarán las entidades, dichos verbos establecerán las relaciones.

Una vez cubierto los dos puntos anteriores se genera la representación gráficamente utilizando la notación indicada.

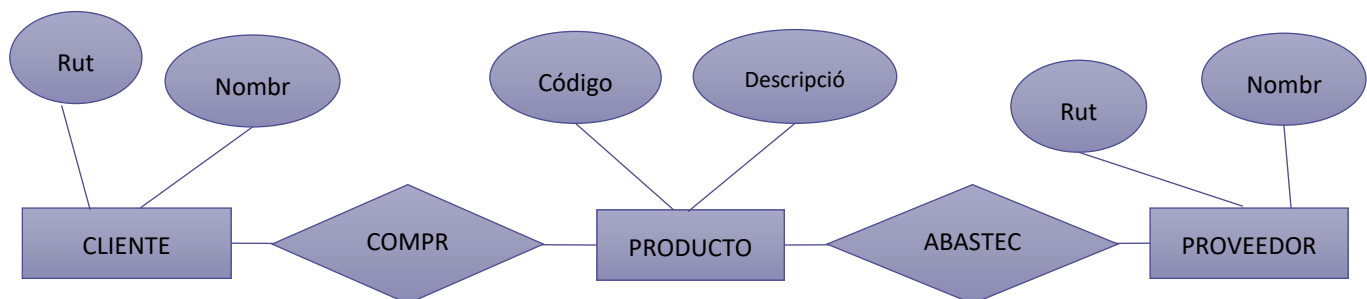


Figura 21. Ejemplo de Modelo entidad relación. Fuente: Elaboración propia, Aida Villamar, 2020.

¹¹ La parte de la compañía a ser representada en la base de datos.



Modelo lógico

El propósito es obtener una representación que use de la manera más eficiente los recursos disponibles en el modelo lógico para estructurar datos y modelar restricciones.

El modelo lógico representa un sistema de manera más formal y técnica que el modelo conceptual. Describe el sistema de manera más específica y se acerca mucho más a la realidad. Este modelo puede ser menos entendible para una persona común pero no para un especialista del área.

Son orientados a las operaciones más que a la descripción de una realidad. Usualmente están implementados en algún Manejador de Base de Datos. El ejemplo más típico es el Modelo Relacional, que cuenta con la particularidad de contar también con buenas características conceptuales (Normalización de bases de datos)

Un modelo lógico debe incluir:

- Todas las entidades y relaciones entre ellos.
- Todos los atributos están especificados para cada entidad.
- La clave principal de cada entidad.
- Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).

El uso de llaves primarias y foráneas permite aplicar una restricción de integridad al modelo de datos.

Aplicado al ejemplo anterior se detallará el modelo lógico, haciendo hincapié en el nivel de detalle de atributos, definición de llave principal y relaciones.

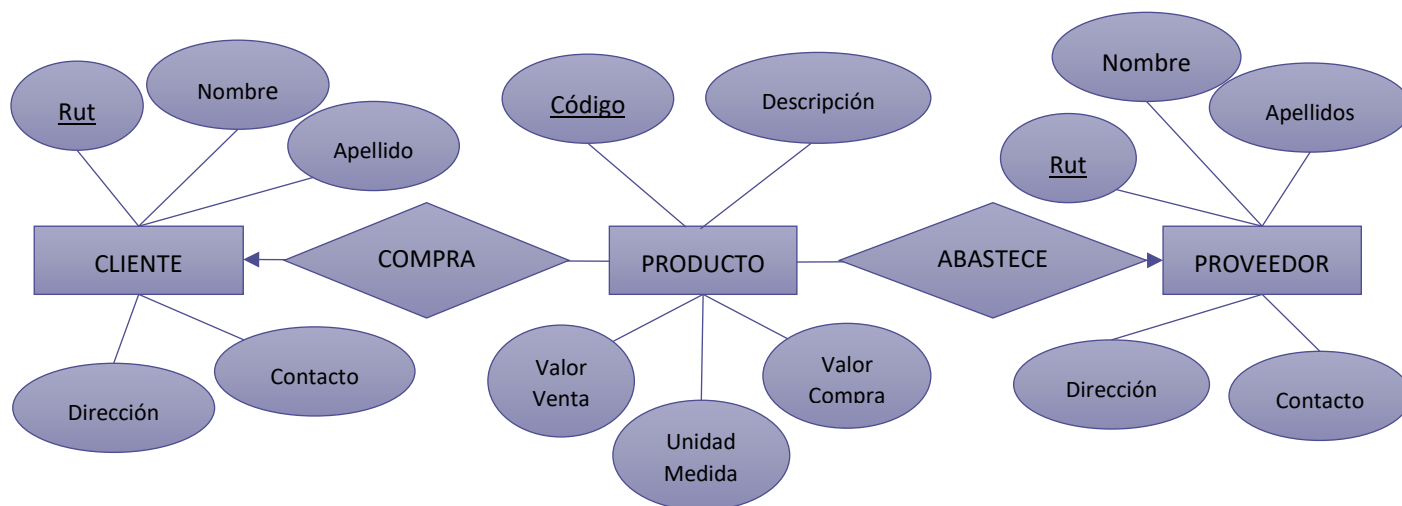


Figura 22. Ejemplo Modelo entidad relación completo. Fuente: Elaboración propia, Aida Villamar, 2020.

Cuando una entidad participa en una relación puede adquirir un papel de entidad fuerte o débil.

- Entidad débil: Una entidad débil es aquella que no puede existir sin participar en la relación, en el ejemplo anterior **producto** es este tipo de entidad ya que no puede existir sin un cliente que lo suministre.
- Entidad fuerte: Un tipo de entidad es fuerte si la existencia de sus ocurrencias no depende de ningún otro tipo. Para el caso analizado este tipo de entidad corresponde a **cliente** o **proveedor** debido que sus datos de existencia no dependen de una tercera entidad.

Un ejemplo utilizando el modelo lógico anterior es la relación entre las entidades “**proveedor**” y “**producto**”. La restricción del modelo es que, no puede existir un producto que no se encuentre asociado a un proveedor, la interpretación es que PROVEEDOR es la entidad fuerte o padre y por ende **producto** es la entidad débil o hija, si se simplifica la definición resultaría que un hijo no puede existir sin su padre, pero por otra parte el padre si puede existir sin sus hijos (Modelo Jerárquico).



Modelo Físico

Se expresa haciendo uso del lenguaje de definición de datos del sistema de gestión de base de datos. El modelo de datos físicos representa cómo se construirá el modelo en la base de datos. Este modelo detalla completamente la estructura de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.

Las características de un modelo de datos físicos incluyen:

- Especificación de todas las tablas y columnas.
- Las claves externas se usan para identificar relaciones entre tablas.
- La desnormalización puede ocurrir según los requisitos del usuario.
- Las consideraciones físicas pueden hacer que el modelo de datos físicos sea bastante diferente del modelo de datos lógicos.

El modelo de datos físicos será diferente para diferentes Sistemas de Gestión de Base de datos. Por ejemplo, el tipo de datos para una columna puede ser diferente entre MySQL y SQL Server.

Los pasos básicos para el diseño del modelo de datos físicos son los siguientes:

- Convertir entidades en tablas.
- Convertir relaciones en claves externas.
- Convertir atributos en columnas.
- Modificar el modelo de datos físicos en función de las restricciones / requisitos físicos.



Requerimientos iniciales y reales para la mejora de un modelo de datos en una base de datos

El proceso de captura de requerimientos es una fase de suma importancia dentro del proceso de desarrollo de software. Éste se preocupa de detectar y estudiar las necesidades del usuario del sistema a desarrollar.

Análisis de los requerimientos

Es el conjunto de técnicas y procedimientos que permiten conocer los elementos necesarios para definir un proyecto de software.

El análisis de requerimientos proporciona una ruta para que los clientes y los desarrolladores lleguen a un convenio sobre lo que debe hacer el sistema. La especificación, fruto de este análisis provee las pautas a seguir a los diseñadores del sistema.

Reconocer las necesidades del sistema

La identificación de los requerimientos no es una tarea sencilla ya que, el analista de sistemas tiene que recopilar información de diversas fuentes, además de encontrarse de forma constante con problemas técnicos y sociales para obtener la información.

El analista debe tener algunas características para poder alcanzar la meta de esta fase, entre ellas está la imparcialidad, la capacidad de considerar que cualquier cosa es posible, estar atento a los detalles y ser flexible, ya que cada usuario tiene una visión diferente.

Recopilación de información

Existen diversas formas de reunir la información necesaria para poder distinguir los requerimientos. Los métodos más tradicionales consisten en las entrevistas, cuestionarios o formularios, la observación de los trabajadores y la documentación escrita de la organización.



Conclusión

A través de los contenidos de la semana 1 has podido comprender cómo es el proceso de toma de requerimientos con los clientes y la importancia fundamental de esta etapa, ya que son la base de todo lo que se hará posteriormente. Además, conociste los procesos que siguen una vez que tenemos clara la información por parte del cliente.

Ideas clave

1. Cuadro sinóptico con las fases genéricas que se practican durante el proceso del software.

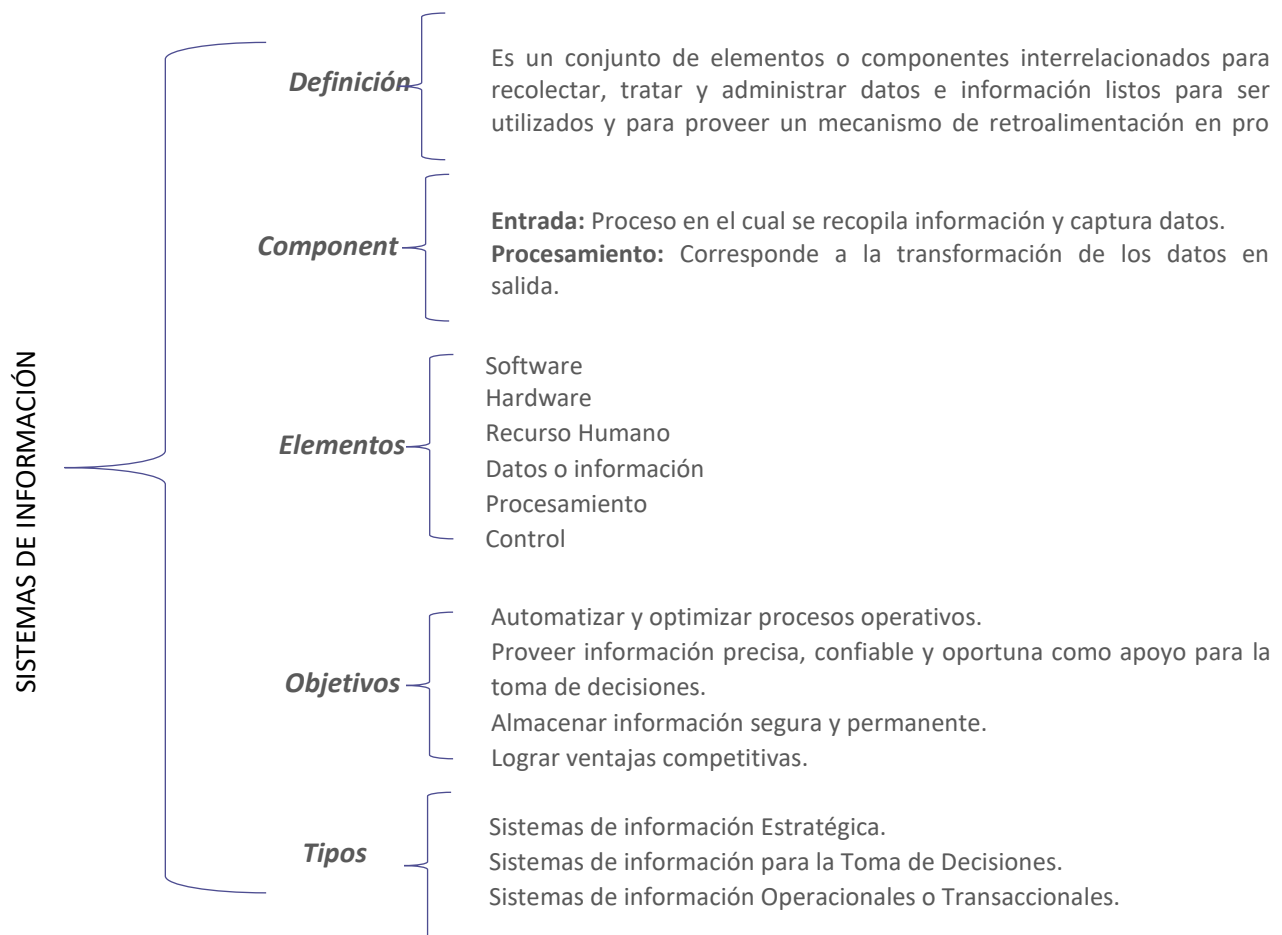


Fuente: El Informático Nica, Blog IngSoftkePC93, 2012.

Sistemas de Información

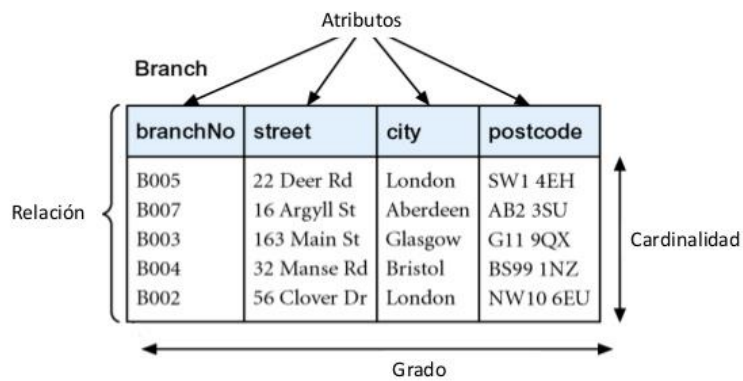
Un sistema de información se puede definir técnicamente como un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización.

A continuación, un cuadro sinóptico con sus fundamentos básicos:



Los sistemas de información son una poderosa herramienta que dentro de las organizaciones revoluciona el desarrollo de virtualmente todas las actividades, afectando de manera directa el desempeño de la organización dentro del mercado.

El modelo relacional



Links y material de profundización

MÓDULO: “Bases de datos relacionales”		Unidad: 1
Recurso	Descripción	
 Video	Estimados estudiantes, a continuación, podrán revisar el video donde se refuerzan los conceptos vistos en este tema:	
	Análisis de requerimiento: https://www.youtube.com/watch?v=50aSAUFRAlk	
	¿Qué es un modelo de negocios? https://www.youtube.com/watch?v=kEg8nh9_AvI	
	Ciclo de vida de desarrollo de Software https://www.youtube.com/watch?v=k8byUApEc4c	
	Ciclo de vida del software https://www.youtube.com/watch?v=Z9LNk12ndm4	
 Presentación	Conceptos De Calidad https://www.youtube.com/watch?v=nnZnMjorDrY	
	Para identificar, más fácilmente, cuáles son las necesidades de nuestro sector productivo frente a las TIC, puedes visitar este estudio: https://www.uv.es/~bellochc/pdf/pwtic1.pdf	
	Otros enlaces: <ul style="list-style-type: none">✓ http://sisinformacion.obolog.es/definicion-clasificacion-sistema-informacion-2011378✓ https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum✓ https://proyectosagiles.org/que-es-scrum/✓ https://techlandia.com/clave-foranea-base-datos-relacional-hechos_353336/✓ https://www.ecured.cu/%C3%8Dndice_de_base_datos✓ https://www.academia.edu/37283341/Marketing_internacional_-_Michael_Czinkota.pdf	



Lecturas complementarias

En los capítulos 1 y 2 encontraras acerca de los sistemas de base de datos y los modelos de datos.

Abraham Silberschatz; Henry Korth; S. Sudarshan, (2006). *Fundamentos de bases de datos. Quinta Edición.* Mcgraw-Hill/Interamericana de España, s. A. U.



Bibliografía

Abad, F. (2006). *¿Do re qué?* Córdoba: Berenice.

Actividades básicas de un sistema de información. (s.f.). Obtenido de <https://sites.google.com/site/tablooooooooooon/sistema-de-informacion/4-actividades-basicas-de-un-sistema-de-informacion>

aspgems. (s.f.). Obtenido de <https://aspgems.com/metodologia-de-desarrollo-de-software-iii-modelo-en-espiral/>

ASPGEMS. (s.f.). Obtenido de <https://aspgems.com/metodologia-de-desarrollo-de-software-iii-modelo-en-espiral/>

auraportal.com. (s.f.). Obtenido de <https://j6v7c9j6.stackpathcdn.com/wp-content/uploads/2019/07/aprendizaje-por-refuerzo-automatico-machine-learning-inteligencia-artificial.png>

Bedini González, A. P. (2005). *Gestión de Proyectos de Software.* Valparaíso: UTFSM.

Dato, proceso e información. (s.f.). Obtenido de <http://agusdelpapa13.blogspot.com/2015/09/dato-proceso-e-informacion.html>

Diseña bases de datos simples. (s.f.). Obtenido de <http://munozbasesdedatos.blogspot.com/2016/07/que-es-una-independencia-fisica.html>

Diseña bases de datos simples. (s.f.). Obtenido de <http://munozbasesdedatos.blogspot.com/2016/07/que-es-una-independencia-logica.html>

Diseño de bases de datos. (s.f.). Obtenido de <http://munozbasesdedatos.blogspot.com/2016/07/que-es-una-independencia-fisica.html>

Diseño de bases de datos. (s.f.). Obtenido de <http://munozbasesdedatos.blogspot.com/2016/07/que-es-una-independencia-logica.html>

docsity.com. (s.f.). Obtenido de <https://www.docsity.com/es/documentacion-del-sistema-de-gestion-de-la-calidad-apuntes-gestion-de-la-calidad/330095/>

e-ducativa. (s.f.). Obtenido de http://e-ducativa.catedu.es/44700165/aula/archivos/repositorio/1000/1080/html/2_diseo_de_una_base_de_datos.html

e-ducativa. (s.f.). Obtenido de http://e-ducativa.catedu.es/44700165/aula/archivos/repositorio/1000/1080/html/2_diseo_de_una_base_de_datos.html

Fandom. (s.f.). Obtenido de <https://vignette.wikia.nocookie.net/so2/images/8/8c/Sistema-operativo.png/revision/latest/scale-to-width-down/400?cb=20120928034940&path-prefix=es>



Fandom. (s.f.). Obtenido de <https://vignette.wikia.nocookie.net/so2/images/8/8c/Sistema-operativo.png/revision/latest/scale-to-width-down/400?cb=20120928034940&path-prefix=es>

Gestion de Proyectos de Software. (s.f.).

grandespymes.com. (s.f.). Obtenido de <https://www.grandespymes.com.ar/2012/12/07/sistemas-de-informacion-estrategicos/>

Hindemith, P. (2009). *Armonía tradicional*. Bueno Aires: RICORDI AMERICANA.

images.slideplayer.es. (s.f.). Obtenido de https://images.slideplayer.es/8/2273301/slides/slide_11.jpg

itsoftware. (s.f.). Obtenido de <https://i0.wp.com/itsoftware.com.co/content/wp-content/uploads/2017/04/toma-de-requisitos-de-software.jpg?ssl=1>

itsoftware. (s.f.). Obtenido de <https://i0.wp.com/itsoftware.com.co/content/wp-content/uploads/2017/04/toma-de-requisitos-de-software.jpg?ssl=1>

KENDALL, K. E. (2011). *Análisis y diseño de sistemas*. México: PEARSON EDUCACIÓN.

LCC. (s.f.). Obtenido de http://200.57.56.254/lcc/mapa/PROYECTO/libro17/modelo_de_cascada.gif

LCC. (s.f.). Obtenido de http://200.57.56.254/lcc/mapa/PROYECTO/libro17/modelo_de_cascada.gif

losprocesosinformaticos. (s.f.). Obtenido de <http://losprocesosinformaticos.blogspot.com/>

Michels, U. (2009). *Atlas de Música*. Madrid, España: Alianza.

monografias.com. (s.f.). Obtenido de <https://www.monografias.com/trabajos104/metodologia-desarrollo-utilizada-estudios-videojuegos-y-materiales-audiovisuales/metodologia-desarrollo-utilizada-estudios-videojuegos-y-materiales-audiovisuales.shtml>

Pérez, M. (1995). *Diccionario de música y los músicos*. Madrid, España: ISTMO.

repositori.uji.es. (s.f.). Obtenido de http://repositori.uji.es/xmlui/bitstream/handle/10234/135526/TFG_MolesTormo_Aaron-1.pdf?sequence=1&isAllowed=y

Scholes, P. A. (1984). *Diccionario Oxford de la Música*. Barcelona: Edhasa / Hermes / Sudamericana.

scrumenespanol. (s.f.). Obtenido de <https://scrumenespanol.files.wordpress.com/2015/09/diagrama-proceso-scrum.gif?w=553&h=414>

scrumenespanol. (s.f.). Obtenido de <https://scrumenespanol.files.wordpress.com/2015/09/diagrama-proceso-scrum.gif?w=553&h=414>



Slideshare - Analisis De Requerimientos. (s.f.). Obtenido de <https://es.slideshare.net/SergioRios/unidad-13-analisis-de-requerimientos>

Slideshare - Analisis De Requerimientos. (s.f.). Obtenido de <https://es.slideshare.net/SergioRios/unidad-13-analisis-de-requerimientos>

unife.edu.pe. (s.f.). Obtenido de <http://www.unife.edu.pe/evaluacion/udual/Modelo%20V.pdf>

weebly. (s.f.). Obtenido de <http://sistemas-operativos-distribuidos.weebly.com/uploads/3/7/2/7/37275681/1712519.jpg?325>

Weebly. (s.f.). Obtenido de <http://sistemas-operativos-distribuidos.weebly.com/uploads/3/7/2/7/37275681/1712519.jpg?325>

www.emprender-facil.com. (s.f.). Obtenido de <https://www.emprender-facil.com/wp-content/uploads/2019/07/cafteria.png>