**ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE (SRS)**

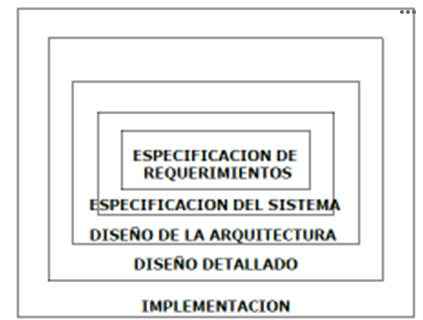
Hoy verá la importancia de la especificación de requerimientos y las características que tienen que cumplir.

**¿Que es SRS?**

Es la documentación de los requerimientos. También podemos decir que es una herramienta de comunicación de:

* la comprensión del dominio
* el negocio
* el sistema
* Es parte de un acuerdo contractual
* Papel clave en el testeo

La especificación de requerimientos es parte del acuerdo contractual y tiene un papel clave en el testeo.



En este esquema vemos como a partir de la especificación de requerimientos van apareciendo distintas capas que tienen que ver con el desarrollo de software.

Los beneficios de tener un documento de especificación de requerimientos son: (IEEE 830)

* Establece las bases del acuerdo consumidores-proveedores sobre qué debe hacer el producto
* Reduce el esfuerzo de desarrollo
* Provee una base para estimaciones
* Provee una “baseline” para validación y verificación
* Facilita la transferencia del producto
* Sirve como base para mejoras

Las especificaciones clásicas u ortodoxas, incluyen características como lo que tiene que proveer el sistema:

Núcleo: Conjunto de definiciones de funciones que establecen las funciones a ser ejecutadas por el sistema y están soportadas por las descripciones de las estructuras de datos a ser procesados

Típicamente incluyen:

* Funcionamiento normal: entradas, salidas, operaciones y control
* Funcionamiento anormal, excepciones
* Propiedades de las que depende: ej. Seguridad
* Performance
* Calidad: mantenibilidad, modificabilidad
* Cambios esperados
* Especificaciones de datos
* Definiciones de estructuras de datos
* Volúmenes (tasa de flujo de datos)
* Fuentes y destinos de datos

Pero estamos subestimando algunos conceptos importantes:

* objetivos
* aspectos deseables
* restricciones en su desarrollo

Estos aspectos son lo que potencia las metodologías ágiles. Cuando escribimos una historia de usuario no sólo se describe la funcionalidad sino también el objetivo de esa funcionalidad. Entonces se puede decir que en una especificación clásica que describe lo que se necesita también necesita el para que. Ese Para que no da un contexto para ayudarnos a tomar decisiones.

**UNA VISION AMPLIA DE LA SR**

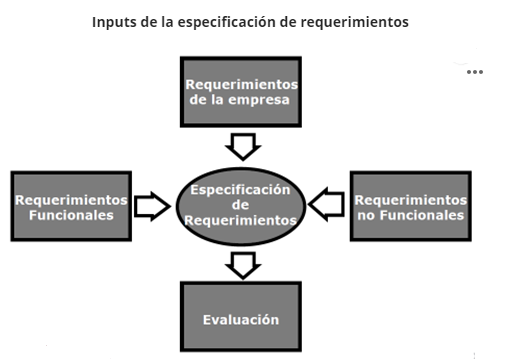
Complementar la especificación funcional con:

* La comprensión del dominio
* La comprensión de las restricciones

Incluir: La descripción de los requerimientos en la empresa expresados en términos de los fenómenos comunes a la empresa y al dominio del sistema.

No incluir en la SRS:

* Requerimientos del proyecto
* Diseños
* Planes de aseguramiento de calidad



En este cuadro puede observar las distintas entradas que tiene la especificación de requerimientos.

Los requerimientos funcionales, son las funcionalidades, operaciones, módulos que tiene que implementar la aplicación.

Los atributos de performance, disponibilidad, modificabilidad, entre otros, son requerimientos no funcionales.

Los requerimientos de la empresa se refieren al contexto de la empresa, el dominio y por otro lado las restricciones a nivel de negocio.

**¿Quiénes son los usuarios de las SRS?**

* Ejecutivos
* Gerentes
* Expertos en el dominio
* Arquitectos
* Diseñadores e implementadores
* Testeadores

**CARACTERISTICAS DE UNA BUENA SRS (IEEE)**

Una buena especificación de requerimientos debe ser:

* Correcta
* No ambigua
* Completa
* Consistente
* Rankeada
* Verificable
* Modificable
* Traceable

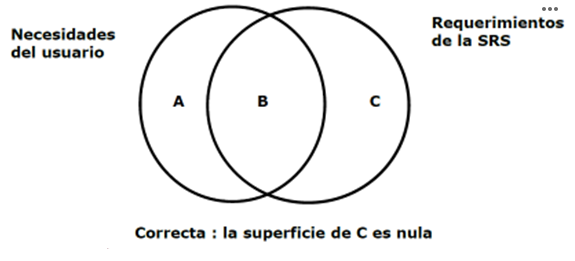
- Entenderlas pero no memorizar todo lo siguiente que explica cada una.

**Correcta**

Una especificación es correcta si no tiene errores en términos de correctitud.

“Sí y sólo si todo requerimiento formulado en ella debe ser satisfecho por el software”

Pero puede ser correcta pero no abarcar todas las necesidades de usuario



**No ambigua**

Según IEEE, Una SRS es inequívoca si, y solo si, cada requisito establecido en él tiene una sola interpretación. Como mínimo, esto requiere que cada característica del producto final se describa utilizando un solo término único.

En los casos en que un término utilizado en un contexto particular pueda tener múltiples significados, el término debe incluirse en un glosario en el que su significado se haga más específico.

Según la RAE, encontramos en su 22da edición , la siguiente definición:

Ambigüedad: cualidad de ambiguo

Ambiguo, gua:

(Del lat. ambigŭus).

1. adj. Dicho especialmente del lenguaje: Que puede entenderse de varios modos o admitir distintas interpretaciones y dar, por consiguiente, motivo a dudas, incertidumbre o confusión.

2. adj. Dicho de una persona: Que, con sus palabras o comportamiento, vela o no define claramente sus actitudes u opiniones.

3. adj. Incierto, dudoso

Entonces podemos decir que **una SRS es no ambigua:**

* “Sí y sólo sí todo requerimiento formulado en ella tiene una sola interpretación”, equivale a que no hay incertidumbre ni más de un significado
* Ello requiere que al menos cada característica del producto final se describa con un único término
* Debe evitarse toda afirmación ambigua.
* “Todos los clientes tienen el mismo campo de control”
* “Todos los archivos son controlados por un bloque de control de archivo”
* El lenguaje natural es fuente de ambigüedades
* Los lenguajes de Especificación Formal ayudan a evitarlas
* Tener presenta la SRS como documento de interfase

**SRS Completa.**

“Una SRS es completa si incluye los siguientes elementos:

a)Todos los requerimientos significativos que impone al software

b)Todas las respuestas del software a todas las clases posibles de inputs

c)Todas las etiquetas y referencias a las figuras, tablas y diagramas de la SRS y las definiciones de los términos y unidades de medida”

* Incluye todo lo que se supone que hace el software
* La “completitud” es el atributo más difícil de definir y de detectar violaciones. Se trata de detectar ausencias a partir de analizar lo presente en la SRS.
* Ventajas del uso de prototipos.
* **Un principio de redacción: el software hace sólo lo que dice en la SRS.**
* También define todas las respuestas del software, a todas las clases de datos de input en todas las clases de situaciones realizables.
* Incluye los inputs válidos y los no válidos.
* El output también puede ser función del estado actual del sistema.

No podemos dejar de lado el seguimiento de los estándares que la organización haya resuelto aceptar.

Si no se utiliza un standard, señalarlo y explicar por que se omite.

**SRS Consistente.**

Aquí hablamos de consistencia interna:

“Una SRS es internamente consistente sí y sólo sí ningún subconjunto de requerimientos está en conflicto”

Existen tres tipos posibles de conflictos entre:

a) Características de objetos del mundo real

b) Acciones

c) Descripciones de objetos del mundo real

Otros tipo de inconsistencias:

* Comportamiento conflictivo: dos partes especifican diferentes y conflictivos:
* estímulos para inducir una respuesta:
* respuestas a un mismo estímulo y condición
* Términos conflictivos: dos términos diferentes se utilizan para definir lo mismo
* Características conflictivas: dos partes requieren rasgos contradictorios
* Inconsistencia temporal: dos partes requieren comportamientos temporales contradictorios

**SRS rankeada (por importancia y/o estabilidad) 🡪 (hay algunas más importantes que otras pero son todas importantes)**

“Si cada requerimiento tiene una identificación que indica la importancia o estabilidad de cada requerimiento particular” esta identificación ayuda a:

* que los clientes consideren mejor cada requerimiento
* que los desarrolladores toman mejores consideraciones de diseño y asignan esfuerzos más adecuados
* Estabilidad: se puede expresar en términos del número de cambios que se esperan
* Necesidad: Los requerimientos se pueden clasificar en

a) Esenciales: el software no es aceptable sin estos

b) Condicionales: pueden mejorar el software, pero no es inaceptable si está ausente

c) Opcional: esa función puede o no estar presente

**SRS verificable**

Todo requerimiento incluido en ella es “verificable”.

Existe un proceso finito y efectivo en costo con el que chequear que el sistema a construir satisface el requerimiento. En general los requerimientos ambiguos no son verificables

Ejemplos de requerimientos verificables y no verificables:

**Requerimientos no verificables:**

* “El producto debe trabajar bien” o “El producto debería tener una buena interfase”.
* “El programa no debe entrar nunca en un loop infinito”.
* “El output del programa debe estar habitualmente dentro de los 10 seg.”.

**Requerimiento verificable.**

“ A partir de producido el evento X, se obtendrá el output del proceso:

Debe estar dentro de los 20 seg. el 60% de las transacciones; y dentro de los 30 seg. el 99% de las transacciones”.

Las Acciones para corregir los requerimientos no verificables son:

* Eliminar o revisar requerimientos no verificables.
* Transformar en verificables a los no verificables
* Fuentes de no verificabilidad
* Ambigüedades
* Problemas no resolubles
* Cantidades no medibles

**Pregunta de examen: ¿Este requerimiento es verificable?**

**SRS Modificable. 🡪 Tiene que ver con la escalabilidad**

**Con respecto a esta característica se puede indicar que:**

Cualquier cambio puede hacerse fácilmente manteniendo la “completitud” y consistentemente.

También requiere tener una organización coherente y fácil de usar y no ser redundante.

**SRS Rastreable.**

Podemos decir que es rastreable si:

Si es claro el origen de cada requerimientos y facilita la referencia de cada requerimiento en futuros desarrollos o mejoras de las documentación.

**Tipos de rastreabilidad:**

* hacia atrás
* hacia adelante

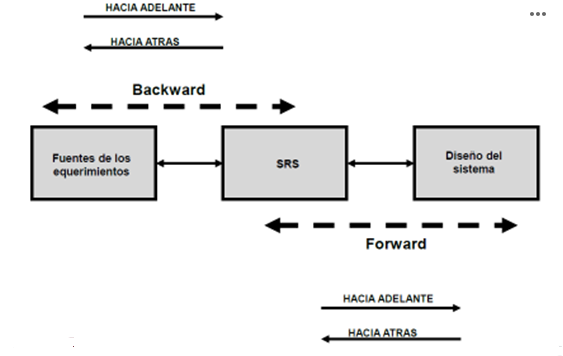
**Beneficios:**

* Importancia en operaciones y mantenimiento
* Claridad del origen de cada requerimiento
* Facilita negociar los requerimientos
* Facilita referenciar cada requerimiento individual establecido en la SRS

**Los objetivos de la “rastreabilidad”**

La rastreabilidad es importante en el diseño de los componentes y en el testing.

A continuación, puede observar la trazabilidad en el ciclo de desarrollo.



**Tipos de rastreabilidad:**

**Podemos utilizar estos métodos:**

* numeración de párrafos
* un requerimiento por párrafo
* identificar cada requerimiento
* convenciones sintácticas

Otros criterios:

Estos criterios son complementarios de los criterios abordados en la IEEE, aunque en algunos casos tienen el mismo sentido:

**Otros criterios**

* Comprensible para los consumidores
* Independiente del diseño
* Concisa
* Organizada
* Anotada
* Utilizable en operación y mantenimiento

**Cuando hablamos de que debe ser compresible para los consumidores**, nos referimos a la utilización del lenguaje natural ya que permiten alcanzar las metas anteriores, pero son imposibles de comprender por los que están fuera del área.

De qué sirve tener una especificación que en realidad no se entiende. Mientras lo especificamos en lenguaje natural podemos identificar inconsistencias y ambigüedades.

•Primeros lectores de una SRS

•Clientes y usuarios

•No ambigüedad

•Es una condición clara de la comprensibilidad

**Independiente del diseño**.

* No implica una arquitectura específica de software o algoritmo.
* No limita a una alternativa de diseño
* Factibilidad de diseños potenciales
* Hay excepciones aceptables

En cuanto a excepciones aceptables, a veces sucede que necesitamos indicar ciertas restricciones porque tenemos que utilizar cierta tecnología r porque es un requerimiento, porque nuestro sistema se tiene que adaptar o integrar con otros sistemas entonces más allá de tecnologías arquitecturas de servicios micro servicios o lo que fuera, tenemos que respetar cierta tecnología para que esta integración suceda ,incluso puede ser que se necesite más integración: no tenemos que usar tal base de datos tal plataforma tales servidores y eso de alguna forma condiciona de a poco.

**SRS Concisa, organizada y anotada**

CONCISA. A igualdad de cualidades, la mejor es la más corta. Tiene que tener lo mínimo indispensable, no debemos agregar cosas que en realidad son agregados .

ORGANIZADA. Una SRS es organizada si los requerimientos que contiene son fácilmente localizables.

ANOTADA. Las anotaciones a una SRS proveen guía al desarrollo de la organización del documento. Hay dos casos de anotaciones requeridas necesidad y estabilidad

**SRS Utilizable en operación y mantenimiento**

Debe estar disponible para alguno de estos casos.

•Uso en el eventual reemplazo del software.

•Para el personal de mantenimiento

•Importante: Modificar requiere conocer el origen de la función

**Estándares disponibles.**

Existen una serie de estándares para poder documentar las especificaciones, por ejemplo:

* Departamento de Defensa (DoD): DI-MCCR-80025A
* NASA: SMAP-DID-P200-SW
* IEEE/ANSI 830-1993
* Naval Research Laboratory: SRS del programa de vuelo del A-7E

Pero el estándar de la IEEE está organizado de una forma clara y simple, por lo que debemos tenerlo en cuenta. **Tenerlo en cuenta nomás**

**Indice del Standard de IEEE**

1. Introducción

1.1. Propósito

1.2. Alcance

1.3. Definiciones, acrónimos y abreviaturas

1.4. Referencias

1.5. Overview

2. Descripción general

2.1. Perspectiva del producto

2.2. Funciones del producto

2.3. Características del usuario

2.4. Restricciones generales

2.5. Supuestos y dependencias

3. Requerimientos específicos

Apéndices

Este índice abarca toda los aspectos de la documentación de una especificación de requerimientos. A continuación se explica cada uno de los puntos.

**1. Introducción**

En este punto debemos proveer una visión global de la SRS.

**1.1. Propósito**

Aquí se delinea el propósito de la SRS y se especifica a quien se dirige.

**1.2. Alcance**

En este capítulo se explica los productos de SW, que hará y que no hará cada uno. Describimos la aplicación.

**1.3. Definiciones, acrónimos y abreviaturas**

Aquí incluimos las definiciones de los términos, acrónimos y abreviaturas requeridas para interpretar las especificaciones de requerimientos de software.

**1.4. Referencias**

Aquí detallamos una lista completa de todos los documentos referenciados.

**1.5. Overview**

EN este capítulo describimosr qué contiene el resto de la SRS y explicar cómo está organizada la SRS.

**2. Descripción General**

Describe los factores generales que afectan al producto y a los requerimientos, facilita su comprensión

**2.1. Perspectiva del producto**

* Detallamos la relación con otros productos o proyectos
* Productos independientes
* Componentes de un sistema o de un proyecto:
* Hardware y equipamiento periférico
* Diagrama de bloques
* Restricciones de diseño

**2.2. Funciones del producto (características del producto)**

* Resumen de las funciones que ejecutará el software.
* Comprensibilidad
* Diagrama de bloques
* No establece requerimientos específicos,

**2.3. Características del usuario**

* Características generales del usuario
* Restricciones impuestas por los interactuantes

Aquí detallamos cuales son los perfiles de usuario en términos de usabilidad o accesibilidad o de nivel de entendimiento necesario o uso del producto.

**2.5. Supuestos y dependencias**

Son los factores que afectan los requerimientos, las restricciones de diseño y los cambios que pueden afectar los requerimientos en la SRS.

**2.4. Restricciones generales**

Son los límites a las opciones para diseñar el sistema:

* Políticas regulatorias
* Limitaciones de hardware
* Interfases con otras aplicaciones
* Operaciones paralelas
* Funciones de auditoría
* Funciones de control
* Requerimientos de lenguajes de alto nivel
* Protocolos de “signal handshake” (ej: XON/XOFF)
* Criticalidad de la aplicación
* Consideraciones de seguridad (Safety and Security)

**3. Requerimientos específicos**

* El sector mayor y más importante de la SRS
* Presentación y conceptualización del desarrollo de los requerimientos
* El contexto de la ingeniería de requerimientos
* Se organiza en torno a un conjunto de conceptos

**Conceptos involucrados**

**Interfaces externas**

En este punto se reflejar una descripción detallada de todos los inputs y outputs del sistema:

Refleja contenidos y formatos:

* Nombre del ítem
* Propósito
* Fuente/destino del input/output
* Rango de validez, precisión, tolerancia
* Unidad de medida
* Timing
* Relaciones con otros inputs/outputs
* Formatos y organización de pantallas y ventanas
* Formatos de datos y comandos
* Mensajes de finalización

**Requerimientos funcionales**

Definen las acciones del sistema al aceptar y procesar los inputs y en generar los outputs. En general tienen la forma “el sistema debe”

Incluye (entre otras):

* Chequeos de validez de los inputs
* Secuencias de operaciones
* Respuestas a situaciones anormales (overflow, manejo de errores)
* Relaciones entre inputs y outputs
* Puede convenir el particionamiento en subfunciones, sin que ello signifique que el diseño se particione así.

**Performance y restricciones de diseño**

**Requerimientos de performance:**

Son requerimientos estáticos y dinámicos del software o de la interacción humana del software como un todo

Estos requerimientos estáticos incluyen:

* Cantidad de terminales
* Usuarios simultáneos
* Cantidad y tipo de información a manejar
* Requerimientos dinámicos incluyen:
* Número de tareas y transacciones a procesar por unidad de tiempo, carga normal y picos
* Restricciones de diseño: se imponen por estándares, hardware, software disponible, etc.

**Atributos del sistema**

Hay atributos del software que pueden funcionar como requerimientos. Son atributos tales como:

* Confiabilidad
* Disponibilidad
* Seguridad
* Mantenibilidad
* Portabilidad

**Reglas de Negocio 🡪 Cosas que se pueden y no hacer en mi sistema**

Desde hace al menos veinte años, existen artículos referidos a las Reglas de Negocio que las describen desde una perspectiva del negocio o con un enfoque de sistemas. En este artículo se muestra como incorporarlas en un proyecto de implementación de procesos de negocios, para ello se incluyen algunas definiciones comúnmente aceptadas a objeto de tener un mismo entendimiento, presentando un método de recopilación y una lista de los usos posibles de éstas.

En primer lugar, tenemos que entender que las Reglas de Negocio siempre han estado presentes, lo que ocurre es que se han denominado de distintas maneras y se han tratado de manera parcial. Es así como las tenemos en:

* Parámetros, por ejemplo tasas de impuestos, porcentajes de descuentos, listas de valores posibles, etc.
* Procedimientos, flujos, rangos de autorización, instrucciones, manejo de excepciones, etc.
* Datos, características del dato, clasificaciones, elementos que indican como tratarlo, p. ej: activo fijo, producto para la venta, materia prima, etc.

Al no estar presente el concepto de Reglas de Negocios en un proyecto de implementación tradicional de un ERP, se genera una dispersión de la información correspondiente a las Reglas de Negocio, y una dilución de la responsabilidad de su mantenimiento y documentación.

La dispersión de la información se genera porque al no considerar el concepto en cuestión, las reglas se tratan en distintos objetos como ser: herramientas para la parametrización de un sistema, programas (codificación en duro *hardcore*), estructuras de datos, reglamentos, manuales de procedimientos, conocimiento de los usuarios, entre otros. A su vez la dilución de las responsabilidades se genera por el proceso de generación; que nos es sistemático en cuanto a la recopilación, identificación de los responsables y a la forma de documentar. En resumen, se mezclan responsabilidades propias del Negocio con las de Informática.

Por tanto, en un **proyecto de implementación de procesos de negocio** se incluye una tarea con un método sistemático para la recopilación y documentación de las Reglas de Negocio, tal que tanto las áreas del negocio, de auditoría e informática cuenten con una fuente única y válida para estos efectos.

**Definiciones**

En general, todo el mundo sabe que es una Regla de Negocios, literalmente, es aquello que usamos para operar un negocio. Son las guías que determinan como se lleva el día-a-día de las operaciones. Sin reglas se estaría en una situación en la que cada decisión se resuelve en el momento, eligiendo alternativas caso a caso o *ad-hoc*, Y, este modo de operar es muy lento, costoso y puede generar resultados inconsistentes.

Para el término Regla de Negocio se tiene que para un especialista en procesos de negocio puede significar un conjunto de requerimientos asociados con los procesos, que están o no formalizados en una gramática o taxonomía por algún tipo de mecanismo. Para el especialista en Base de Datos, puede ser un requerimiento específico que se satisface mediante la definición de alguna característica de los datos, que expresará en los valores posibles de un determinado campo. Y, para la gente del negocio no son más que la **manera de hacer las cosas**.

Una Regla de Negocio define o limita un aspecto del negocio con el objeto de establecer una estructura o un grado de influencia que condiciona el comportamiento de los actores del negocio. A menudo las Reglas de Negocio están focalizadas en el control, en la forma de realizar los cálculos, otras permiten establecer las políticas, y así se tienen en cualquier actividad del negocio, que requiera que la gente actúe de una forma pre-establecida.

**Qué NO son las Reglas de Negocio**

Las Reglas de Negocio **no son software**. Las Reglas de Negocio pueden ser **implementadas** en el software, pero esto a menudo representa sólo una parcialidad de sus definiciones, ya que partes importantes quedan expresadas en los Procedimientos Manuales. También existe la posibilidad, poco común por ahora, de habilitarlas mediante una Máquina de Reglas o un Sistema Experto. De modo que se debe entender que efectivamente las Reglas del Negocio, como su nombre lo indica, son parte del negocio y, no de alguna plataforma particular de hardware / software que pudiera soportarlas.

Las Reglas de Negocio de **no son Proceso**, de ninguna forma. Roger Burlton indica que: Se debe separar el flujo del conocimiento. Dónde las Reglas de Negocio representan la parte conocimiento que guía al flujo, de aquí su nombre de Regla de Negocio.

**Recopilación**

Dado que las Reglas del Negocio son **conocimiento**, su formulación “la acción de explicitarlas por escrito- es un proceso de recopilación, entendiendo que están en la cabeza de los empleados, a diferencia del **flujo** que habitualmente se documenta en un manual de procedimiento o en un *workflow*.

Podemos, entonces, decir que las Reglas de Negocio corresponden a un conocimiento **tácito**, por consiguiente el proceso de recopilación consistirá en convertir este conocimiento que tienen las personas en conocimiento **explícito**.

*Es necesario tener presente que el poseer el conocimiento de determinadas Reglas de Negocio es poder. Y, que por tanto su recopilación puede tener complicaciones.*

Para proceder a la recopilación de las Reglas de negocios resulta un buen punto de partida tener en consideración los principios denominados *The business rule approach* desarrollados por Ronald Ross, estos son:

* Deben ser explícitas y escritas.
* Expresadas en términos sencillos.
* Existen independientemente de los procedimientos y workflows (ej.: modelos).
* Se construyen a partir de hechos, éstos se definen a partir de conceptos, los que a su vez se representan por medio de términos (ej.: glosarios).
* Guían o influencian el comportamiento conforme a una forma pre-establecida.
* Son motivadas por factores de negocios identificables e importantes.
* Son accesibles a las partes autorizadas (ej.: tienen dueños).
* Están en una fuente única (ej.: repositorio de reglas).
* Son especificadas por las personas que tienen directa relación con ellas y que poseen el conocimiento relevante (ej.: los usuarios claves).
* Son gestionadas “administradas"- (ej.: son parte de la Gobernabilidad de los Procesos de Negocios)

Las **reglas de negocio de software** son directrices o condiciones que definen cómo funciona un sistema y qué comportamiento debe seguir para cumplir con los objetivos de la organización. Son independientes de la implementación tecnológica y describen aspectos del negocio que deben ser respetados al desarrollar una solución de software. Estas reglas las impone el cliente o el producto owner que las consigue.

**Clasificación de las Reglas de Negocio**

**Reglas permisivas**: Definen condiciones bajo las cuales ciertas acciones o comportamientos están permitidos. Estas reglas amplían el conjunto de acciones que un sistema o usuario puede realizar bajo ciertas circunstancias.

**Ejemplo**: En un sistema bancario, "Un cliente puede solicitar un préstamo si tiene más de 18 años y una cuenta activa con un historial crediticio positivo".

**Características**:

* Establecen situaciones en las que ciertas operaciones son válidas.
* Permiten flexibilidad o acceso a funcionalidades específicas.
* Generalmente abren el abanico de posibilidades dentro de ciertos límites.

**Reglas restrictivas**: Son reglas que imponen limitaciones o restricciones sobre las acciones que un sistema o usuario puede realizar. Estas reglas establecen condiciones que no deben ser violadas.

**Ejemplo**: En un sistema de ventas, "No se puede procesar una venta si el inventario del producto está en cero".

**Características**:

* Limitan acciones bajo circunstancias específicas.
* Garantizan que ciertas condiciones se cumplan antes de que algo pueda realizarse.
* Evitan comportamientos indeseados o incorrectos en el sistema.

**Ejemplos de Casos Reales**

1. **Amazon (e-commerce)**:

* **Regla permisiva**: Los usuarios pueden realizar compras internacionales si su país está incluido en la lista de destinos permitidos.
* **Regla restrictiva**: No se puede procesar un pedido si el usuario no ha registrado una dirección válida en su cuenta.

1. **Airbnb (alquiler de propiedades)**:

* **Regla permisiva**: Los propietarios pueden aceptar o rechazar solicitudes de reserva basadas en las fechas disponibles.
* **Regla restrictiva**: Un usuario no puede reservar una propiedad si no ha completado la verificación de identidad.

1. **Uber (servicios de transporte)**:

* **Regla permisiva**: Los usuarios pueden solicitar un viaje si tienen saldo suficiente en su cuenta o una tarjeta de crédito válida asociada.
* **Regla restrictiva**: Un conductor no puede aceptar más viajes si ha superado el límite de horas de conducción permitidas por la legislación local.

Este tipo de reglas garantiza que los sistemas sigan los procesos de negocio de manera eficiente y segura, evitando errores o malentendidos que podrían afectar a la organización.

Además de la clasificación en **reglas permisivas** y **restrictivas**, las reglas de negocio también se pueden clasificar de diferentes maneras según su naturaleza y propósito dentro de un sistema. A continuación, te presento otras clasificaciones comunes:

**1. Reglas de cálculo:**

Son reglas que definen fórmulas o métodos para obtener un valor a partir de ciertos datos de entrada.

* **Ejemplo**: En un sistema de facturación, "El total de la factura se calcula sumando el subtotal y el impuesto (IVA) correspondiente".

Estas reglas establecen cómo se obtienen valores críticos para el negocio y son fundamentales en áreas como la contabilidad o los precios.

**2. Reglas de validación:**

Se utilizan para garantizar que los datos introducidos o procesados cumplen con ciertos criterios o condiciones específicas.

* **Ejemplo**: En un formulario de registro de usuarios, "El campo de correo electrónico debe tener un formato válido antes de permitir el registro".

Las reglas de validación son clave para evitar errores y asegurar la calidad de los datos.

**3. Reglas de desencadenamiento (trigger):**

Estas reglas especifican acciones que deben ejecutarse automáticamente en respuesta a ciertos eventos o condiciones.

* **Ejemplo**: En un sistema de gestión de inventarios, "Si el inventario de un producto cae por debajo de 10 unidades, se debe generar automáticamente una orden de reposición".

Son muy útiles en sistemas que requieren acciones automáticas basadas en cambios en los datos.

**4. Reglas de procesamiento secuencial:**

Definen el orden en el que deben ejecutarse ciertos procesos o tareas dentro de un sistema.

* **Ejemplo**: En un sistema de aprobación de préstamos, "El proceso de evaluación crediticia debe completarse antes de la aprobación final del préstamo".

Este tipo de reglas ayuda a gestionar procesos donde el orden es crítico para el resultado correcto.

**5. Reglas de integridad:**

Se enfocan en mantener la consistencia y exactitud de los datos en el sistema, asegurando que no haya incoherencias.

* **Ejemplo**: En una base de datos de empleados, "No puede haber dos empleados con el mismo número de identificación".

Estas reglas son esenciales para evitar errores que comprometan la confiabilidad de la información.

**6. Reglas de autorización:**

Establecen qué acciones o accesos están permitidos a diferentes usuarios o roles dentro del sistema, basándose en sus privilegios.

* **Ejemplo**: En un sistema de gestión de proyectos, "Solo el gerente del proyecto puede modificar las fechas de entrega del proyecto".

Garantizan que solo los usuarios con el nivel de autorización adecuado puedan ejecutar ciertas acciones.

**7. Reglas temporales:**

Se basan en condiciones que dependen del tiempo o eventos cronológicos para ser aplicadas.

* **Ejemplo**: En un sistema de promociones de ventas, "Un descuento del 20% solo es aplicable durante los primeros 10 días de cada mes".

Estas reglas son útiles en sistemas que gestionan eventos temporales o condiciones basadas en el calendario.

**Clasificación Adicional**

* **Reglas obligatorias**: Son reglas que **siempre** deben cumplirse sin excepción. Ejemplo: "Un pedido no puede ser procesado si el cliente no tiene un método de pago válido".
* **Reglas opcionales**: Son reglas que aplican en casos específicos, pero pueden no estar presentes todo el tiempo. Ejemplo: "Si el cliente tiene un cupón de descuento, puede aplicarlo al realizar una compra".

Estas clasificaciones permiten estructurar mejor las reglas de negocio en sistemas complejos, garantizando que cada aspecto del proceso de negocio esté correctamente cubierto y que los comportamientos esperados se mantengan dentro de los límites establecidos.

**SO DEL LENGUAJE NATURAL**

En esta clase verá la importancia en la forma en que se describen los requerimientos, cuales son las limitaciones del lenguaje y la manera de solucionarlos

**La importancia del lenguaje.**

El lenguaje natural es el más utilizado para especificar requerimientos o especificar conocimiento en general. Pero la precisión con que lo hacemos, es muy importante para poder comunicarnos.

En un proyecto donde hay muchas personas, con características de pensamiento diferentes, por ejemplo el personal de IT, el cliente y el encargado del negocio, tienen visiones muy diferentes acerca de lo mismo. Aquí vemos cuantas variables hay en juego para que la comunicación del conocimiento, de los requerimientos, lleguen a plasmarse en una línea de código.

Podemos decir que en un dominio específico, tenemos un lenguaje específico que hay que aprender para poder entender el dominio. Por eso el proceso de ganar conocimiento es un proceso iterativo donde tenemos que ganar conocimiento y validarlo para saber si el conocimiento adquirido es el correcto.

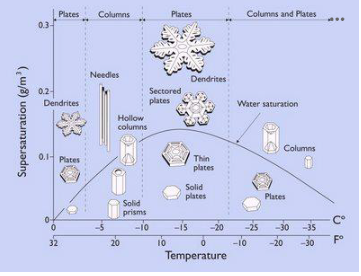
"Se construye el mundo a partir del lenguaje. Si no tiene nombre, no existe"

**Constructivismo lingüístico**



Veamos un ejemplo:

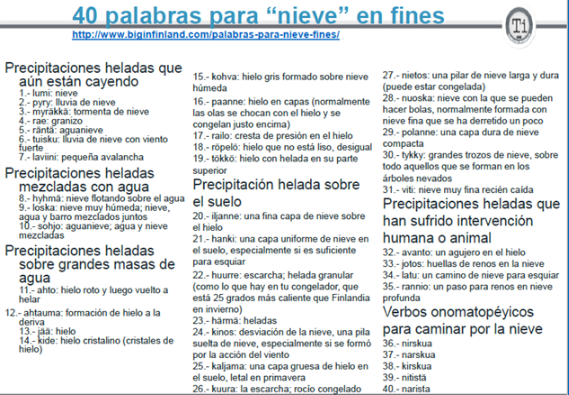
En la figura podemos ver un volcán (Lanín) , con nieve. Este es un conocimiento general que fue adquirido y lo podemos representar. Lo mismo lo podemos encontrar en un refrigerador y también llamamos nieve a ese elemento blanco y con características similares.



Sin embargo un autor hizo un análisis de los cristales de nieve. Y descubrió que a partir de diferentes condiciones del entorno, como la temperatura humedad, etc. se producen diferentes tipos de cristales que les da características diferentes a la nieve.

Con esto queremos decir que cada persona puede ver lo mismo o pueden entender el significado de una palabra que puede ser distinto al significado de otra persona,

A continuación veremos los distintos significados de la palabra "nieve"



Esto quiere decir que cuanto más entendemos del tema, tenemos más palabras o herramientas para describir los requerimientos. Cuanto más sepamos del dominio, con más detalle podemos describirlo. De la misma manera si hablamos el mismo lenguaje del cliente, tenemos más posibilidad de entender y describir lo que ese cliente necesita.

**Características de ciertos lenguajes**

Existen distintos lenguajes con distintas raíces que tienen distintas características.

Hay lenguajes donde no existen las palabras izquierda, derecha y en cambio utilizan par referirse a ellas, los puntos cardinales. Lenguajes donde no existen palabras para nombrar colores, entonces utilizan analogías.

**El aspecto de las traducciones.**

Los diferentes significados de las palabras, no se reducen a cuestiones de los diferentes lenguajes. También sucede cuando cruzamos diferentes dominios, diferentes negocios. Si necesitamos trabajar en un sistema bancario, necesitamos entender el "idioma" de ese dominio. Al principio generalizamos y eso no es del todo correcto, porque cuando se generaliza, perdemos detalle.

Ejemplos:

⋄Escalera versus ladder + stairs

**A veces las traducciones no existen:**

⋄Mate, asado

**O pueden variar el significado ligeramente**

⋄Durazno, Melocotón, Albaricoque, Damasco, Chabacano, Nectarina

⋄Ciervo, venado, reno

⋄Leopardo, Guepardo, Jaguar

**O puede variar mucho el significado**

⋄Puntualidad

»Abstracciones

⋄elimina información no relevante

**False Friend**

Este término se refiere a palabras que, por nuestro lenguaje pensamos que tienen un determinado significado pero en realidad el significado es distinto.

A continuación veremos distintos significados de la palabra Palestra:



Por esto es tan importante el concepto de validar.

También tenemos el problema de expresiones ambiguas.



Si analizamos literalmente el término toro mecánico y nunca vió el juego, uno asocia con el sesgo de su conocimiento y puede dar un sentido distinto.

**Evolución del lenguaje / comunicación**

Es importante conocer como fue la evolución del lenguaje de la comunicación.

»Señas: ideas concretas, comunicación en contacto directo

»Lenguaje simbólico: ideas abstractas, comunicación oral

»Escritura: comunicar ideas mas allá del contacto directo

**La traducción**

Cerramos la idea del lenguaje natural comprendiendo la importancia de ser preciso, la importancia de comprenderlo, el tema de cómo evitar generalizaciones o abstracciones para evitar perder información perder conocimiento

Con respecto a la traducción veremos cuál es el proceso mental cognitivo, qué sucede cuando uno necesita hacer una traducción.



¿Por qué tenemos una imagen de un hospital y una clínica con doctores y enfermeros?

Lo que sucede es que uno podría pensar que los enfermeros lo que hacen es ayudar a los doctores a aplicar medicamentos con algunas mediciones y realizan algunas tareas delegadas por los doctores pero también es cierto que hacen una tarea muy importante que es la de ser intermediarios entre el paciente y los doctores.

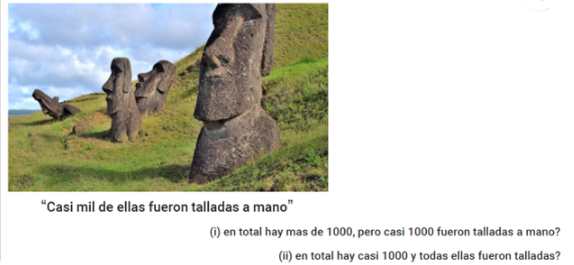
Y eso es lo que sucede con los analistas. Son intermediarios entre el equipo de desarrollo y los expertos del dominio. Y el ser intermediario implica que esa persona tiene que comprender el lenguaje de los 2 mundos. Y comprender el mundo significa darle un nombre a cada cosa y entonces poder manejar ese lenguaje.

La traducción en ese ámbito no se ve como una simple búsqueda en un diccionario de una palabra, machear y asociarla a otra. A veces con algunas cosas funcionan. Pero la traducción no puede verse como una reescritura de cada palabra. La traducción tiene que verse como un proceso en donde se comprende si se escucha, se transforma eso a ideas. La verbalización se transforma en ideas y se vuelven a expresar esas ideas utilizando otro idioma otro código. Podemos observarlo cuando estamos codificando, estamos entendiendo que hay que hacer y lo estamos transformando en un código de programación determinado. Es decir que tenemos que comprender para volver a expresarlo pero de otra forma y es ahí donde tenemos que manejar los lenguajes. El lenguaje del dominio y el lenguaje de los informáticos, en ese marco, tenemos que ser precisos para que justamente no no se pierda el conocimiento, no se pierda información importante.

**ESTILOS DE ESCRITURA**

Siguiendo con las especificaciones en lenguaje natural, vamos a ver algunos aspectos de estilos de escritura. Prestemos atención a la siguiente afirmación:

**Casi 1000 de ellas fueron talladas a mano**



La expresión hace referencia a las estatuas de la isla de pascua que tenemos en la imagen anterior, Piensen , reflexionen cómo se interpreta "casi 1000 de ellas". La verdad es que podría tener 2 interpretaciones: por un lado se podría interpretar que hay muchas de esas estatuas, que hay más de mil pero que casi 1000 fueron talladas a mano. Otra interpretación es que en total hay casi 1000 y todas ellas fueron talladas a mano. La diferencia es sutil pero claramente son 2 situaciones muy distintas: Esto tiene que ver con la ambigüedad de la expresión del lenguaje natural y es un tema crítico. imagínense que estamos especificando y la especificación se puede interpretar de distintas formas. Claramente una regla de negocio ambigua no lleva un error en la o por lo menos a una posibilidad de error en la codificación. En esta actividad controlada, el ejemplo puede ser una especificación real y fue construida fue escrita justamente pensando en escribirlo de la mejor forma posible, de forma de que no haya ambigüedades y hemos encontrado que a pesar de que en forma explícita se prestó atención para escribirlo de la mejor forma, de todas maneras hay aspectos que por ahí dejan librado a la duda y esto es un tema importante porque si en este caso un enunciado corto de 3 párrafos escrito en forma explícita se escribió pensando en todas estas características y se deslizaron errores, imaginen qué pasa con una especificación mucho más extensa escrita, construir una especificación sin errores . Vemos entonces que va a ser mucho más crítica.

y mucho más este posible que se dicen cuestiones a mejorar eh y vamos a ver vamos a ver en esta cuestión son cuestiones que entren cuando estamos redactando especificaciones técnicas sí entonces un aspecto considerar la estructura de las oraciones de todas formas son poetas estas recomendaciones no van a ayudar para escribir mejor no esté cualquier cosa así para escribir un mail aunque sea dirigido a personas son son son guías básicas prácticas no pero bueno pero está bueno volverá a recordar estructura de una oración que tiene sujeto y predicado y que dentro del predicado tenemos un verbo y tenemos un objeto que realiza la acción y a veces hay 2 objetos sí pero pero bueno la versión sencilla arranca por un sujeto que hacen acción y un verbo que describe la acción es el punto de partida lo que puede pasar es que el verbo es un verbo transitivo que la acción recaiga sobre un objeto

**Especificación**

Lea la siguiente especificación:

Sea una empresa que ofrece travesías en Kayak. La misma ofrece distintas travesías que varían en duración (medido en cantidad de días) y en el itinerario (un recorrido que se describe con texto). La empresa admite kayakistas inexpertos como también de experiencia. Cualquier persona que quiera hacer una travesía le solicita a la empresa y la empresa le informa el arancel.

El cálculo del arancel se realiza de la siguiente forma. A los kayakistas inexpertos se le cobra el valor de la travesía sumado 100 pesos. Mientras que los de experiencia, tienen un descuento del 25% si la travesia es de 1 dias y se descuenta 15% si es de mas de 1 dia.

Luego, los kayakistas reciben beneficios a partir de las travesías contratadas anteriormente (es un programa de fidelidad). Los kayakistas inexpertos que contrataron mas de 10 travesias o contrataron travesias en donde la suma de días supera 20 dias, reciben un descuento del 15%. Los kayakistas expertos que contrataron sólo travesias que superan los 2 días y además la suma total supera los 30 días, reciben una bonificacion de 500 pesos.

En este caso, la empresa cotiza travesías. El objeto en cuestión es "travesías" entonces el verbo cotizar sí recae sobre las travesías sobre elementos que cotizan pero también puede ser que tengamos 2 objetos objeto directo e indirecto. El objeto directo es la cosa sobre la cual recae la acción y el objeto indirecto es a la persona a la que le sirve la acción. Entonces la empresa cotiza travesías a los pasajeros. "A los pasajeros" sería el objeto directo de la esto es para comenzar, para para pensar en una oración que tiene la estructura sencilla y completa con sujeto y predicado y con las partes que nos interesan dentro de predicado. Lo que puede suceder es que en realidad muchas veces nos vemos tentados a escribir en voz pasiva: la empresa cotiza la travesía activa porque la cotización la realiza la empresa y quien realiza la acción la empresa es el sujeto en la oración. En voz pasiva podríamos escribir: "las travesías son cotizadas". En esta construcción falta el sujeto y precisamente se suele utilizar voz pasiva cuando no se conoce el sujeto. En una especificación técnica debería estar siempre el sujeto, porque si no lo conocemos lo tenemos que averiguar y si es algo general que todos lo conocemos es preferible que quede en forma explícita. Esta bien utilizar la voz activa y está bueno repetirlo para que no hayan inconvenientes.

**Estructuras simples (kernel sentences)**

Sujeto + Verbo. El verbo no es transitivo, la acción no recae sobre ningún elemento.

●La empresa cotiza.

Sujeto + Verbo + Objeto. El verbo es transitivo, la acción recae sobre algún elemento.

●La empresa cotiza travesías

Sujeto + Verbo + Objeto directo + Objeto indirecto. El objeto indirecto, es el elemento o persona sobre el cual recae el provecho o daño

●La empresa cotiza travesías para los pasajeros

**Voz pasiva**

En una oración con voz activa, el sujeto realiza la acción que recae en un objeto.

En una oración en voz pasiva, el objeto sobre el cual recae la acción se convierte en el sujeto de la oración.

●La empresa cotiza travesías (voz activa, la empresa es quien hace la cotización)

●Las travesías son cotizadas por la empresa (voz pasiva, las travesias reciben la acción de ser cotizadas).

**En el caso de Sujeto tácito el sujeto esta implícito.**

●Juan Pérez contrata una travesía. Va con sus amigos.

**Los Adjetivos y adverbios**

Son Indicios de que hay mas información por descubrir.

**Conectores**

Los conectores unen oraciones para dar semántica o bien para omitir elementos redundantes.

●Juan Pérez contrata una travesía para ir de excursión. “Para” indica la finalidad.

●Juan Pérez, quien contrata una travesía, es inexperto. “Quien” introduce una cláusula subordina para evitar repetir Juan Pérez.

**Pronombres**

Se utilizan para no repetir los sujetos

Pronombre personales: yo, tú, él, nosotros, vosotros, ellos

Adjetivos demostrativos: este, ese, aquel y sus variantes

●Juan Perez contrata una travesía. Él va con sus amigos.

**Sinónimos**

El lenguaje del dominio, incluye expresiones que refieren a lo mismo, aunque no sean estrictamente sinónimos.

●Travesia y travesia en kayak.

●Kayakista, cliente y pasajero. No son sinónimos, pero en el dominio o negocio si lo son.

Hiperónimos es un término más general. Por ejemplo, animal es hiperónimo de insecto.

●Kayakista es hiperonimo de experto e inexperto.

**Rescritura primer parrafo**

* Hoy duermo afuera es una empresa.
* La empresa ofrece travesías en kayak.
* Kayakistas inexpertos contratan travesías en kayak.
* Kayakistas expertos contratan travesías en kayak.
* La empresa ofrece travesías en kayak a Kayakistas inexpertos.
* La empresa ofrece travesías en kayak a kayakistas expertos.
* Las travesías en kayak tienen duración.
* Las travesías en kayak tienen itinerario.
* Las travesías en kayak tienen un costo.
* La duración es un tiempo.
* El tiempo es medido en días.
* El itinerario es una secuencia de lugares.
* El kayakista solicita una cotización a la empresa.
* La empresa informa el arancel al kayakista.

**Reescritura segundo párrafo**

* La travesía en kayak para los kayakistas inexpertos cuesta 100 pesos mas el costo propio de la travesía.
* La travesía en kayak para los kayakistas expertos cuesta entre el 15% y 25% del costo propio de la travesía.
* La travesía en kayak para los kayakistas expertos cuesta 15% menos del costo de la travesía en kayak para travesía en kayak de 1 día de duración.
* La travesía en kayak para los kayakistas expertos cuesta 25% menos del costo de la travesía en kayak para travesía en kayak de más de 1 día de duración.

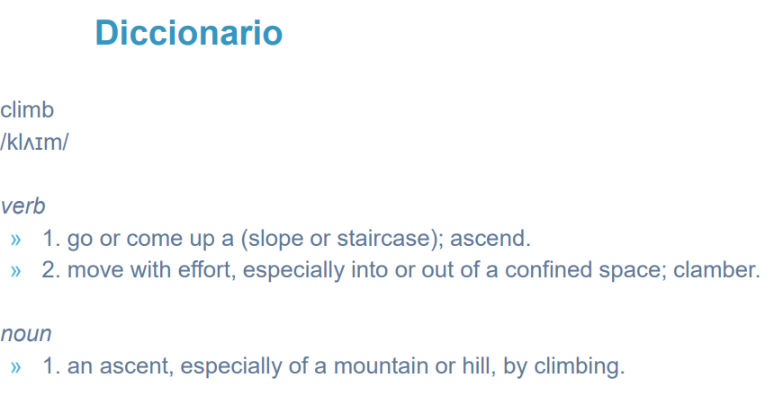
**Rescritura tercer parrafo**

* Hoy duermo afuera es una empresa que ofrece travesías en kayak.
* La empresa ofrece travesías en kayak a kayakistas inexpertos y a kayakistas experimentados.
* Kayakistas inexpertos y kayakistas expertos contratan travesías en kayak.
* Las travesías en kayak tienen distinta duración e itinerario.
* La duración es un tiempo medido en días.
* La empresa cobra 100 pesos mas el costo de la travesía en kayak a los kayakistas inexpertos.
* La empresa aplica un descuento de entre 15% y 25% a los kayakistas expertos.
* La empresa aplica un descuento de 15% en las travesía en kayak de 1 día de duración a los kayakistas expertos.
* La empresa aplica un descuento de 25% en las travesía en kayak de más de 1 día de duración a los kayakistas expertos.

**Clase 11: Diccionario, Glosario y Ontología 🡪 Sólo conocer que es cada cosa y diferenciarla.**

En esta clase vamos a estudiar qué es un diccionario, un glosario y una ontología, para luego poder estudiar un glosario en particular.

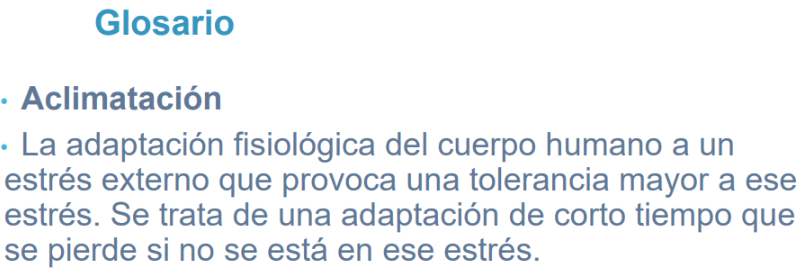
**¿Qué es un Diccionario?**



Los diccionarios (en este caso en inglés) nos muestra la definición (por ejemplo, considerando la palabra como un verbo o como un sustantivo) como se ve en la imágen anterior. Podemos encontrarnos con varias definiciones para un verbo o bien una terca para un sustantivo.

En resumen, tenemos una palabra que necesitamos definir, una categoría gramatical y una definción acorde a esta categoría.

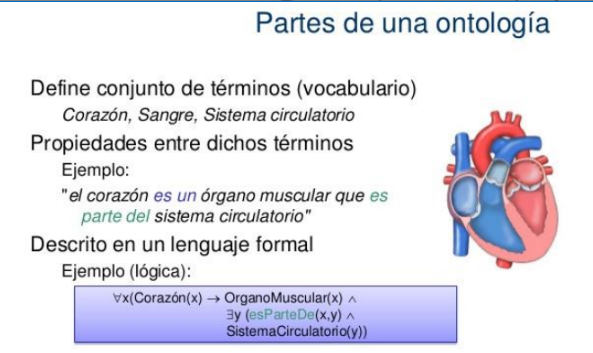
**¿Qué es un Glosario?**

****

En un glosario encontramos una palabra y para ésta encontramos una definición mucho más rica. Un glosario es específico de un dominio y por esto, generalmente encontramos una sola definición muy rica y con mucha información. Veamos otro ejemplo, con más contexto,e n donde encontramos varios sinónimos para describir el concepto:

****

**¿Qué es una Ontología?**

****Una Ontología define un conjunto de términos a partir de un vocabulario. Lo que permite la Ontología es definir propiedades que contextualizan esos términos. En el ejemplo vemos como la ontología nos permite definir que "el corazón es un órgano muscular que es parte del sistema circulatorio". Vemos la relación "Es un" y la relación "Es parte de". Esta ontología nos permite entender como, a partir de las relaciones, nos permite definir el término que queremos entender. Al igual que sucede con el paradigma orientado a objetos, la relación "es un" nos muestra una generalización, es decir mostramos una jerarquía. Es parte de, es una composición. Es decir, una composición de cosas. Por supuesto, existen otras relaciones que permitirían definir una ontología.

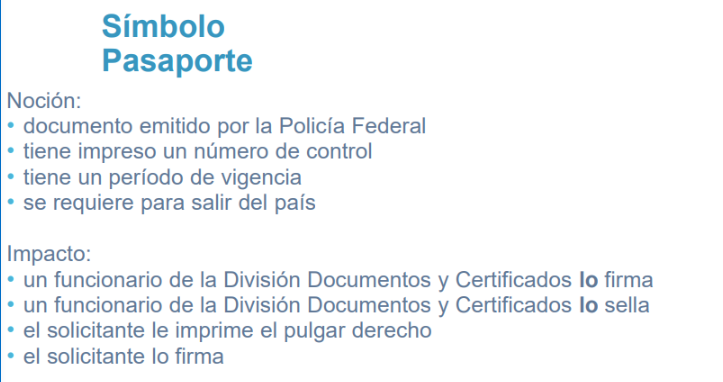
La ontología nos brinda una descripción a partir de un lenguaje formal. Veamos el recuadro coloreado:



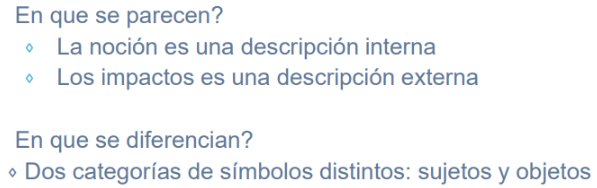
Este lenguaje formal nos permite establecer relaciones. La diferencia entre ontología y diccionario o glosario es que nos permitiría una definición más forma y obtener más información de una forma más automatizada y con cierto nivel de confianza, ya que al ser más precisa da lugar a menos dudas, errores y ambigüedad.

Luego de haber estudiado estos tres conceptos (diccionario, glosario y ontología), vamos a continuar la clase estudiando el glosario para un dominio en particular y con una estructura específica, llamado LEL: **Léxico Extendido del lenguaje**.

Veamos algunos ejemplos para entender mejor el concepto:

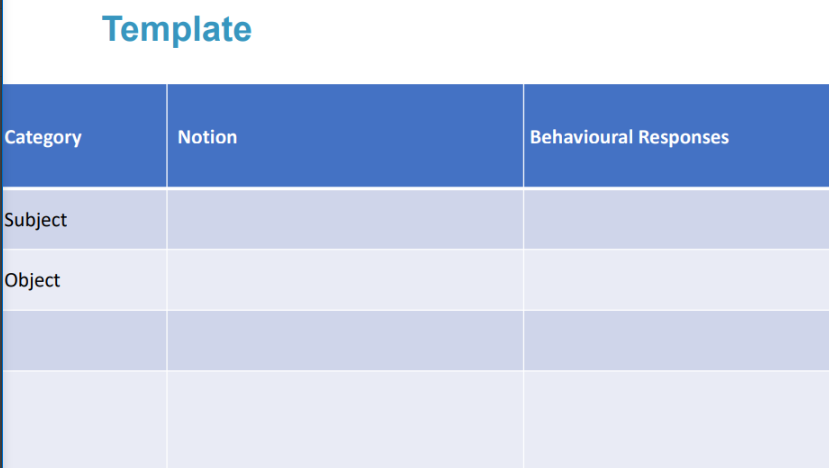


Vemos en el símbolo como se describen 2 sinónimos para el mismo símbolo, tomense dos minutos para reflexionar: ¿Cuales son las diferencias y similitudes entre estos dos símbolos?



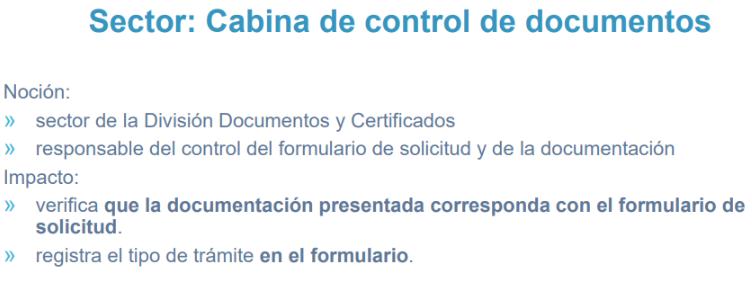
Cuando estudiamos el concepto de diccionario, vimos como a partir de un concepto podiamos encontrar varias definiciones según si era un verbo o un sustantivo. En LEL Los símbolos pueden ser sujetos y objetos, algo similar a lo que sucede con diccionarios. ¡Dos categorías diferentes!

Al final de esta clase vamos a completar la siguiente plantilla a partir de los conceptos estudiados:



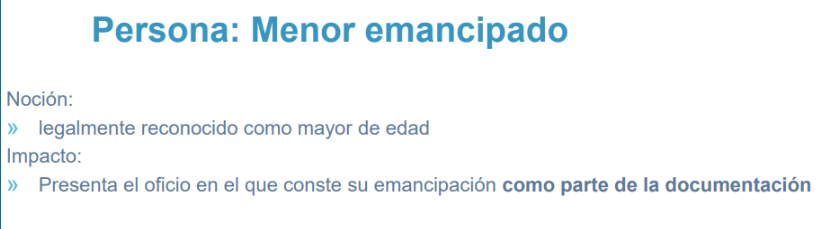
En el campo **Noción** vamos a describir internamente al sujeto y al objeto y en los I**mpactos** vamos a describir las acciones vinculadas.

En esta parte de la clase vamos a continuar estudiando otros ejemplos que nos permitirá ampliar esta tabla con más categorias.



En este ejemplo vemos que aparece la categoría **"Sector"** y con esto ampliamos las categorías sujeto y objeto. En este dominio en particular, cuando uno va a hacer un trámite va a pasar por diferentes sectores (oficinas, por ejemplo). Este sector: se parece a un sujeto o se parece a un objeto? Vean como el impacto permite que esta "Cabina" tenga un rol activo (verifica.. registra). De esta manera, entendemos que se relaciona con un sujeto. Entendemos entonces que un sujeto, es un símbolo activo.

Veamos otro símbolo:



Al igual que un sector (un espacio físico con gente), podría pasar que existan personas que no hagan cosas (receptores de acciones). En ese caso, sería un objeto (no tiene un rol activo).

En el siguiente ejemplo vemos el símbolo Depósito:

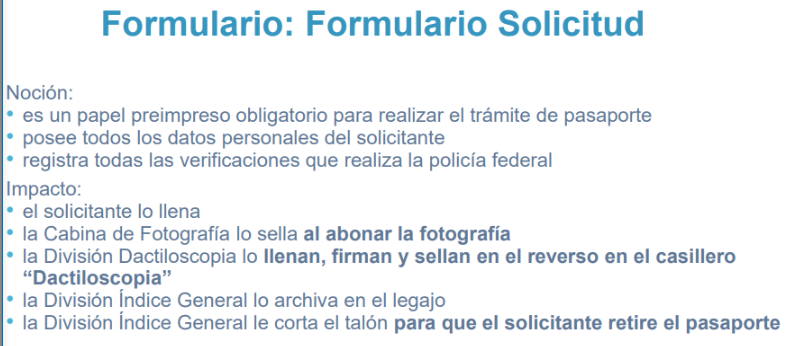


Nos encontramos como otro espacio físico. Sin embargo, este "depósito" puede tener un rol activo o un rol pasivo. ¿Que piensan?. Analizando el impacto, vemos que existe una división externa que utiliza el archivo, por lo tanto este deposito es un objeto. El rol activo lo tiene la división Dactiloscopia.

Vemos que puede haber sujetos que no sean personas y puede haber objetos que si lo sean. Para entender esto mejor, tenemos que detenernos en el impácto: si la mayoría de impáctos en los que se vea un rol activo, es un sujeto. Por el contrario, si la mayoría de impactos tienen un rol pasivo, es un objeto.

Con esto damos una herramienta para definir un sujeto: a través del impacto. El impacto de un sujeto son las cosas que hacen, el impacto de un objeto, son las acciones que reciben.

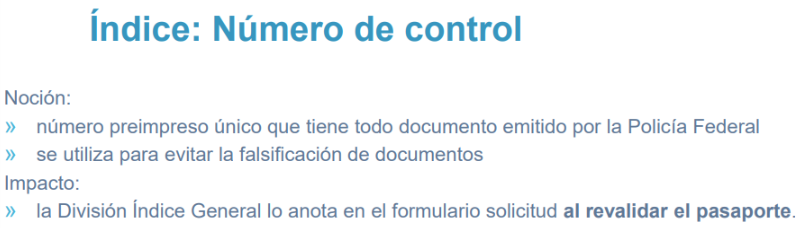
Estudiemos el siguiente símbolo:



Este símbolo es un poco más sencillo. Esta claro que un formulario va a recibir acciones: Por ejemplo: la cabina de fotografía "sella" el formulario. Recibe la acción "sellar". Un formulario es un recurso. Los sujetos (personas, oficinas, divisiones) hacen cosas sobre el mismo. En la noción vemos como se describe el formulario (lo mismo que sucedió con el pasaporte).

Con esto encontramos más pistas de como describir **noción** e **impacto** de **sujeto** y **objeto**.

Sigamos con los símbolos y ejemplos:



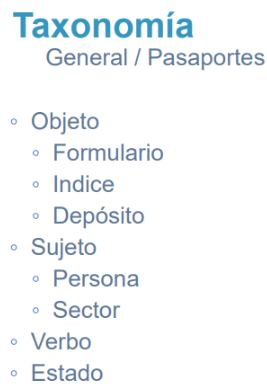
¿Qué rol cumple el número? veamos el impacto: este indice no tiene un rol activo, claramente es un sujeto. Es importante definir un sínbolo aunque solo sea un número. Es importante describirlos de manera uniforme y no ambigüa.

En el siguiente ejemplo, vemos otro símbolo que es diferente a un sujeto y un objeto.



En el ejemplo vemos una serie de pasos que hay que cumplir para realizar la actividad. Esta nueva categoría describe una accion. Sumamos a los sujetos (rol activo, hacen cosas), objetos (rol pasivo, reciben acciones), las actividades que describen las acciones concretas.

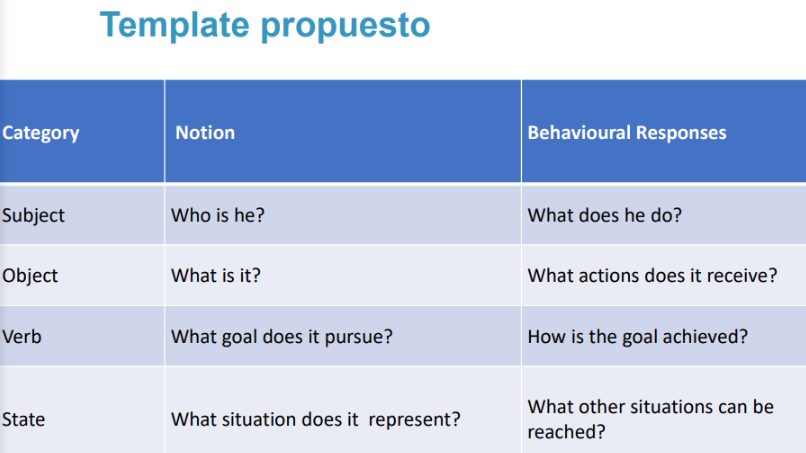
A modos de repaso...



En este resumen, también vemos la categoría "Estado". Esta categoría representa representa el estado de cada paso en una actividad. Esta taxonomía nos ayuda a describir el símbolo. Si describo un objeto, por ejemplo, voy a saber como esta formado y que cosas se hacen con el objeto. Si es un sujeto, digo quién es el sujeto, que características tiene y que cosas hace. Además, nos sirve para darnos cuenta si existen sinónimos. **¿Qué hubiese sucedido con el símbolo solicitante / interesado si los hubiésemos descripto por separado?.**

Con un **glosario LEL** tendremos mucha precisión, tal vez no más que con una ontología, pero sin duda será muy útil a la hora de desarrollar un software.

Repacemos el template propuesto para cada uno de los atributos para cada categoría:



EL **LEL (que es un glosario específico)** TIENE QUE TENER ESTA ESTRUCTURA ES DECIR ESTAS 4 CATEGORIAS 🡪 Subject, Object, Verb, State. Tanto para la Nocion como para el impacto.

La noción describe el símbolo internamente, sus características. E**n un sujeto, decimos ¿quién es? y en un objeto ¿qué es? . En un verbo buscamos describir que acción u objeto percibe y en un estado, que situación representa.**

Los impactos son una descripción externa y su relación con otros símbolos. E**n el caso del sujeto es ¿qué acciones hace? en el caso del objeto ¿qué acciones recibe?. En el caso del verbo es ¿cómo se logra el objetivo? y en el caso del estado es ¿qué otras transiciones podemos llevar a cabo?**

**Ejercicios para terminar de entender los conceptos:**

**Pregunta 1)** Definir el concepto “ticket” de tres formas distintas:

•Diccionario

•Glosario

•Ontologia

**1. Diccionario:**

Un *ticket* se define como un comprobante, recibo o documento que representa la adquisición de un bien o servicio, generalmente indicando detalles como la fecha, el monto pagado y los términos de la transacción. Según su contexto, también puede referirse a una entrada para un evento o transporte.

**2. Glosario:**

En el ámbito de gestión de reservas o ventas, un *ticket* es un comprobante digital o físico que confirma una operación. Contiene información relevante como el código de reserva, el cliente, el estado del pago y la fecha de emisión. Este término es específico de dominios como eventos, transporte, comercio electrónico o sistemas de atención al cliente.

**3. Ontología:**

* **Conjunto de términos (Vocabulario):**
  + Ticket, Cliente, Comprobante, Transacción, Medio de pago, Estado.
* **Propiedades entre dichos términos(Relación):**
  + "El ticket es un comprobante de una transacción efectuada por un cliente, cuyo estado puede ser reservado, pagado o cancelado, y cuyo medio de pago puede realizarse en efectivo o con tarjeta de crédito."
* **Descripción en un lenguaje formal (Lógica):**
  + ∀x(Ticket(x) → Comprobante(x) ∧ ∃y(Cliente(y) ∧ Transaccion(x, y)))
  + ∀z(Ticket(z) → (Estado(z) ∈ {Reservado, Pagado, Cancelado}))
  + ∀w(Ticket(w) → ∃m(MedioDePago(m) ∧ (m ∈ {Efectivo, TarjetaCredito})))

**Pregunta 2)** Con ayuda de internet, defina los siguientes términos:

•Idioma

•Lenguaje

•Dialecto

•Argot

**Idioma:** Un idioma es un sistema estructurado de comunicación utilizado por una comunidad de personas, compuesto por reglas gramaticales, vocabulario y expresiones particulares. Ejemplos incluyen el español, el inglés y el chino.

**Lenguaje:** Es una forma más general de comunicación que puede incluir tanto sistemas naturales (idiomas) como sistemas artificiales (lenguajes de programación). El lenguaje es una herramienta para expresar ideas, sentimientos y transmitir información.

**Dialecto:** Es una variante regional o social de un idioma. Los dialectos pueden diferir en pronunciación, vocabulario o gramática, pero siguen siendo mutuamente inteligibles para los hablantes de la lengua principal.

**Argot:** Es un lenguaje especializado utilizado por un grupo particular de personas, a menudo para facilitar la comunicación dentro del grupo o para mantener la privacidad. Ejemplos incluyen el argot de estudiantes, policías o comunidades artísticas.

**Pregunta 3)** Define en un glosario LEL los siguientes conceptos

•Sujeto: cliente

•Verbos: comprar, pagar, pagar con tarjeta de credito, pagar en efectivo

•Estado: ticket reservado

**Sujeto:**

* **Cliente:**
  + **Noción:** Persona o entidad que realiza una compra o reserva.
  + **Impacto:** Realiza transacciones en el sistema, generando tickets o comprobantes.

**Objeto:** -

**Verbos:**

* **Comprar:**
  + **Noción:** Adquirir un producto o servicio.
  + **Impacto:** El cliente ejecuta la acción de compra, generando un ticket.
* **Pagar:**
  + **Noción:** Efectuar el pago correspondiente a la compra.
  + **Impacto:** El cliente confirma la transacción mediante pago.
* **Pagar con tarjeta de crédito:**
  + **Noción:** Realizar el pago utilizando una tarjeta de crédito.
  + **Impacto:** El sistema procesa el pago y actualiza el estado del ticket.
* **Pagar en efectivo:**
  + **Noción:** Realizar el pago en efectivo.
  + **Impacto:** El cliente entrega dinero físico y el sistema registra el pago.

**Estado:**

* **Ticket reservado:**
  + **Noción:** Situación en la que se ha generado un ticket pendiente de confirmación de pago o uso.
  + **Impacto:** El ticket puede cambiar de estado a 'pagado' o 'cancelado' según la acción posterior del cliente o sistema.