# Marco Teórico

En este capítulo de presentan los conceptos más significativos para la definición de un método inteligente, sensible y adaptado al contexto para la integración de procesos.

## 2.1 Proceso de negocio

Un proceso de negocio (BP – *Por sus siglas en inglés*) “c*onsiste en un conjunto de actividades que se ejecutan en un ambiente técnico y organizacional, y están organizadas de manera lógica para llevar a cabo un objetivo de negocio*” (Weske, 2010).

Los BP representan la dinámica de una organización, ya que definen los flujos de trabajo organizacional, incluyendo a las personas que intervienen, todo ello con el fin de generar valora agregado al producto y/o servicios que ofrece la organización. La Figura 2.1 presenta un ejemplo de BP relacionado con la *Venta de productos.* Se observa un conjunto de actividades ordenadas de una manera lógica para lograr la venta de un producto.

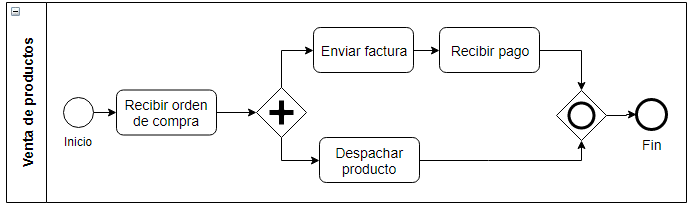


Figura 2.1: Proceso de Negocio *Venta de Producto.* Tomado de (Weske, 2010)

La gestión de procesos de negocio (BPM – *por sus siglas en inglés*) se realiza con base en el ciclo de vida de un BP, el cual propone fases de: (1) Diseño y análisis, (2) configuración, (3) Implementación y (4) Evaluación. Todas las fases están centradas en lograr la automatización y mejoramiento constante de los BP.

Los BP buscan principalmente modelar los objetivos de negocio y estrategias relacionadas, sin embargo, pueden ser usados para tratar con procesos de tipo operacional, como por ejemplo: compra de suministros de aseo (*i.e proceso no estratégico*). A lo anterior se suma que los BP pueden ser empleados para el control de actividades de implementación y monitoreo de ambientes donde son implementados, como por ejemplo *procedimientos de instalación de tecnologías*.

Como instrumento principal para lograr todo lo anteriormente señalado, se tiene al *Modelo de Proceso,* el cual consiste la representación de la estructura del proceso a través de sus actividades, flujos y restricciones asociadas (Dumas et al., 2013). El modelo proceso, sirve como plantilla a seguir por las instancias generadas.

## 2.2 Integración de procesos de negocio

La integración de procesos de negocio (*BPI – por sus siglas en inglés*) es un concepto que ha sido empleado en distintos ámbitos de la informática y la computación. Por ejemplo, en el comercio electrónico suele asociarse a B2B - *Business to Business*, en el desarrollo de software es visto como EAI – *Enterprise Application Integration*, para los sistemas de información como Sistemas de gestión de flujos de trabajo – *WorkFlow System* y para el caso de los ambientes web puede asimilarse como Composición de Servicios Web.

Sin importar el enfoque de la BPI, se pretende lograr la consolidación o agrupamiento de elementos de un sistema, con el fin de obtener una visión holística que facilite la realización de una de las siguientes acciones:

- Fusionar uno o más procesos de negocio con el fin de generar un nuevo modelo de proceso que consolide la funcionalidad de sus antecesores.

- Identificar un comportamiento global de los procesos, a partir del análisis de su intercambio de mensajes.

- Lograr objetivos complejos a partir del intercambio de información, funciones y recursos dentro de un grupo de elementos.

- Actualizar procesos a partir de cambios en su modelo de referencia. La actualización de cambios se realiza mediante procedimientos de propagación.

Con base en lo anterior, la integración de procesos de negocio se puede definir como: *la consolidación de un conjunto de modelos de proceso de manera parcial o definitiva, con el fin de poder analizar y propagar cambios relacionados con su estructura, ambiente de ejecución e información dependiente del dominio.*

Según Morrison et al. (2009) la integración de procesos se caracteriza por dos fases, la similitud y la de consolidación. La primera enfatiza en la clasificación de objetos a partir de rasgos comunes. Sebu y Ciocarlie (2015) plantean que para comparar dos o más procesos se debe tener en cuenta su estructura y semántica.

Una vez se determina si dos procesos pueden integrarse se da paso a la fase de consolidación; la cual se basa principalmente en la alineación estructural y semántica, a partir de diferencias identificadas y busca la generación de un nuevo modelo de proceso.

Cabe señalar, que el proceso obtenido en la integración debe garantizar los mismos niveles de rendimiento, dado en tiempo de ejecución y respuesta, lo cual garantiza que se respeten las condiciones y restricciones de la lógica de negocio, así como variables externas al proceso que influyen en su ejecución.

A modo de ejemplo, en la Figura 2.2, se presenta un ejemplo de la integración de procesos de negocio, en donde a partir de dos procesos antecedentes (Proceso A y Proceso B), se pretende generar un tercer proceso (Proceso C) que sea capaz de reemplazarlos.

La integración comprende los procesos A (líneas continuas) y B (líneas discontinuas); el primero compuesto por dos actividades secuenciales y el segundo por dos actividades separadas por una condición.

Es así como el proceso generado puede estar compuesto por una actividad del proceso A y por la condición y una actividad del proceso B. En este caso las actividades A1 y B2 son seleccionadas ya que son las más apropiadas para la integración. Adicionalmente, el proceso B contiene un condicional C1 que es seleccionado debido a que enriquece la descripción del modelo de proceso generado.

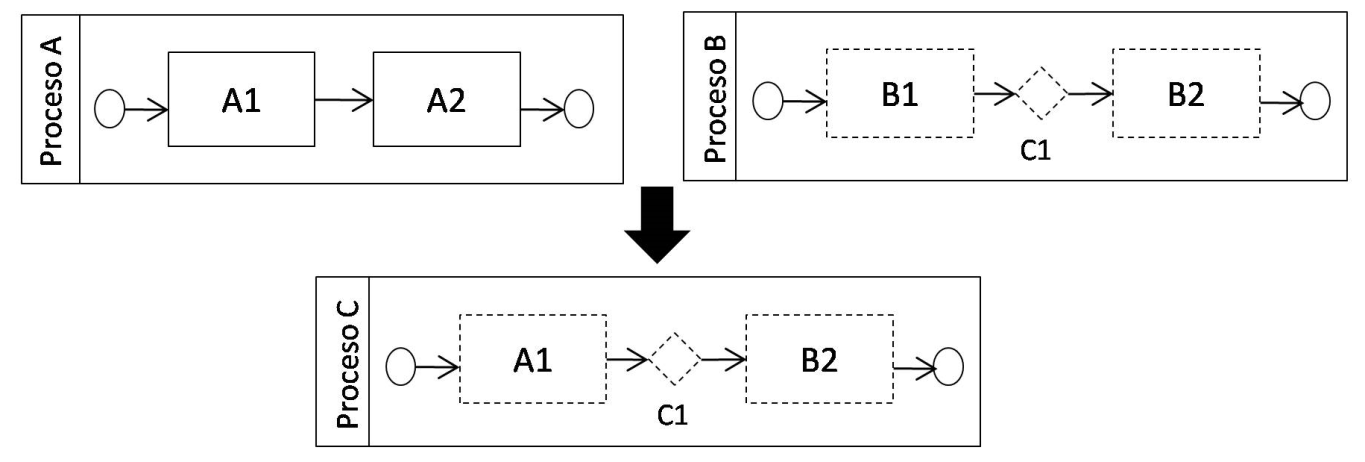


Figura 2.2: Integración de Procesos de Negocio.

## 2.3 Computación sensible al contexto

Según Dey (2001), se define el contexto como “*cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que se considera relevante en la interacción entre el usuario y la aplicación, incluyendo al usuario y a la aplicación misma*”.

Por lo anterior la Computación Sensible al Contexto tiene como objetivo principal el uso del contexto para el mejoramiento de la interacción entre computadores y usuarios. La sensibilidad al contexto radica en la capacidad de adaptarse a situaciones contextuales; por tanto un sistema sensible al contexto es aquel que tiene la capacidad de comprender el ambiente en el cual es ejecutado y adaptar sus operaciones para proveer la mejor experiencia al usuario (Krumm, 2016).

Un modelo de contextohace referencia al sistema que soporta la captura, almacenamiento y procesamiento de datos contextuales, con el fin de relacionar situaciones con los dominios de ejecución. Bettini et al. (2010) concluyen que el uso de Ontologías es una aproximación apropiada para la representación del contexto, ya que permite mayor expresividad y facilita la interoperabilidad de términos, así como la recuperación de información relacionada con los procesos.

Los principales elementos de un modelo de contexto son: Entidades contextuales (Ejemplo: ambiente), atributos contextuales (e.g. temperatura), valores contextuales (e.g. temperatura alta), situaciones contextuales (e.g. temperatura en hora del día).

El principal elemento para razonar sobre el contexto, son las situaciones, ya que están ligadas directamente con el dominio y ambiente de ejecución. Por lo anterior, las instancias de ejecución determinan las situaciones presentes.

Una situación se define como la relación de conceptos de un dominio con el ambiente y con otras instancias, los valores de las relaciones indican la presencia o no de la situación. En Sikounmo et al. (2017) se presenta un ejemplo la relación entre una instancia de la clase *persona* con una instancia de la clase *silla,* dependiendo de la proximidad física entre las instancias, se decide si la persona de puede sentar en la silla, de lo contrario caería al suelo.

Lee et al. (2011) presentan un método generalizado para el procesamiento del contexto para un sistema sensible al contexto (ver Figura 2.3). El método comprende la adquisición de la información, almacenamiento, control y ajuste de niveles de abstracción y la utilización de dicha información en distintos servicios y dispositivos.

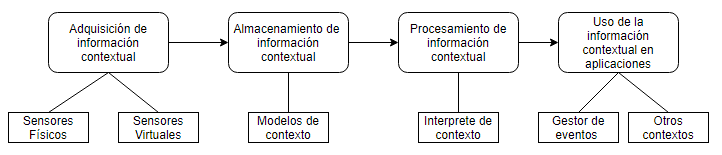


Figura 2.3: Método generalizado de procesamiento del contexto.   
Tomado de (Lee et. al 2011)

La adquisición de información contextual presenta retos interesantes con el uso de sensores, especialmente los virtuales, ya que se pueden aprovechar tecnologías de tipo distribuido, como los servicios web o agentes de software, así como herramientas de representación como lo son las ontologías para el desarrollo de soluciones.

Las ontologías se emplean en el modelamiento del contexto, principalmente para determinar los conceptos e instancias involucrados en una situación. A partir de dicha identificación, se razona por medio de restricciones con el fin de validar la presencia o no de la situación.

## 2.4 Procesos de negocio sensibles al contexto

La representación y modelado del contexto ha tomado auge dentro de las organizaciones, debido a su capacidad de implementación directa con los procesos de negocio. Ploesser et al. en (2009) plantean que, a partir de la información disponible en el contexto es posible mejorar el funcionamiento de los procesos.

Un proceso de negocio, además de estar asociado con un dominio de ejecución, no es ajeno a la presencia de situaciones propias del dominio que restringen los valores de variables del dominio, lo que conlleva a generar comportamientos no esperados en los procesos y con ello los inconvenientes que pueden presentarse en términos la satisfacción de los usuarios de dichos procesos.

Saidani & Nurcan (2009), proponen la representación de las posibles variables del proceso que pueden ser afectadas por el contexto. En la Figura 2.4, se presenta una jerarquización del contexto relacionado basado en cuatro componentes a saber: tiempo, ubicación, recursos y organización.

Se observa que la entidad tiempo, en términos históricos, se compone de un atributo contextual *frecuencia,* el cual puede tener valores contextuales definidos. Así mismo para la categoría de ubicación se determina como valor contextual las zonas geográficas, que a su vez puede ser cambiante.

Cabe resaltar la importancia de la información de los recursos que emplea el proceso, así como la organización donde se ejecuta. La primera de ellas resalta la importancia del actor del proceso y la segunda, procesa información relacionada directamente con la gestión organizacional (i.e. interacciones, comunicación, cercanía dentro del flujo).

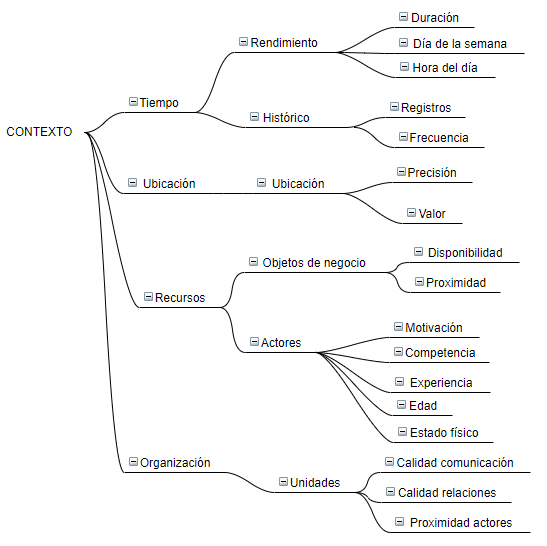


Figura 2.4: Clasificación del contexto para un proceso.   
Tomado de (Saidani & Nurcan, 2009)

Con base en lo anterior, es evidente que para un dominio de aplicación, se deben seleccionar los atributos sensibles al contexto, de esta manera se evita consultar información no prioritaria para el proceso.

Por tanto, el contexto de un proceso de negocio depende directamente de las variables del dominio de ejecución. Ya que estas últimas afectan conceptos del dominio con alertas contextuales que generan la activación de situaciones. Esto quiere decir, que cada situación afecta una o más entidades relacionadas con el proceso.

Santo et al.(2015) definen una situación contextual como el conjunto de valores contextuales *Ce,* representado por medio de un valorespecífico en el tiempo asociado a un dominio *Dom(Ce, tiempo)*.

## 2.5 Problemas de verificación de restricciones

Un Problema de Verificación de Restricciones -PVR-, puede ser expresado de la siguiente manera (Russel & Norvin, 2004): *Dado un conjunto de variables junto con sus posibles respectivos valores y una lista de restricciones, encontrar los valores para las variables que satisfagan las restricciones definidas*.

Un PVR consiste de: un conjunto de variables *X = {x1,..., xi};* por cada variable *x*, un dominio finito *Di* de posibles valores para las variables que aborda; por último, un conjunto de restricciones que determinan el valor que las variables pueden tomar simultáneamente.

Un PVR consta de dos pasos; primero se descubren las restricciones y se propagan a través del sistema. La propagación de las restricciones se conoce como el proceso en el cual se analizan las restricciones en el sistema. Segundo se evalúa si los valores asignados facilitan una solución, de lo contrario se registran como restricciones. Entonces, la propagación continúa con esta nueva restricción hasta encontrar una solución

Una solución a un PVR es una asignación de valores a todas las variables de manera que satisfaga todas las restricciones. Realmente los objetivos que se persiguen con el uso de restricciones en la solución de problemas se centran en encontrar una solución, sin preferencia alguna, encontrar todas las soluciones, o quizás una buena solución según una función objetivo.

El ejemplo clásico asociado a la PVR es el llenado del mapa; básicamente consiste en llenar de colores las zonas geográficas de un mapa. La Figura 2.5, presenta el estado inicial y uno de los estados objetivos.

Cabe señalar que la manera de representar los estados de un problema, es por medio de grafos, lo que facilita la representación de flujos de procesos y la aplicación de mecanismos de razonamiento, como los algoritmos de recorridos.

Para el caso de la BPI, La verificación de restricciones emplea un grafo representativo del modelo de proceso generado, a partir de ello, realiza una propagación de las restricciones sobre el grafo. La propagación permite evaluar si al grafo generado le afecta en su rendimiento lo cambios propuestos.

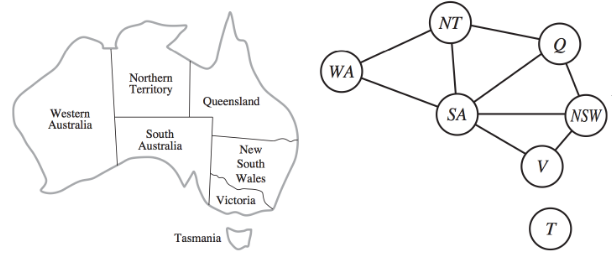


Figura 2.5: Problema de verificación de restricciones. Tomado de (Russel & Norvin, 2004)

## 2.6 Algoritmos de agrupamiento de datos

El agrupamiento -*Clustering-* de datos, consiste en encontrar grupos de instancias con características similares (Van, 2011). Se consideran las técnicas de agrupamiento como parte del aprendizaje *no supervisado,* esto quiere decir, que no se trabaja con datos de ejemplo o históricos.

Las técnicas de agrupamiento tienen variedad de aplicaciones, entre ellas se cuentan: exploración de datos científicos, extracción de texto, procesamiento de datos espaciales, análisis de datos clínicos, aplicaciones web, análisis de ADN y muchas más (Sajja, 2012).

Dentro de los algoritmos más representativos se tiene a los algoritmos de *aglomeramiento* y de *centroides*. El primero de ellos busca a partir de grupos definidos, validar la pertenencia de los datos a un centro. Por su parte los algoritmos basados en centroides (K-means), busca la definición de grupos a partir de datos dispersos.

El algoritmo de agrupamiento K-means, hace parte de las técnicas de aprendizaje de máquina, junto con el descubrimiento de patrones. Para la determinación de los grupos se define una medida de similitud, que puede ser obtenida por medio de heurísticas.

La Figura 2.6 presenta el funcionamiento general del algoritmo K-means. Incialmente se definen de manera aleatoria centros de agrupamiento. Luego en el paso dos, se asigna cada elemento al centro más cercano a este, generando así K grupos de acuerdo a los K centros generados. El paso tres, valida la distancia y de ser necesario calcula de nuevo las agrupaciones a partir de la modificación de los centros. El paso 4 indica la repetición del procedimiento hasta lo indicado por el criterio de parada.

Las restricciones desempeñan un papel importante ya que permiten la segregación de los procesos, a partir de los valores obtenidos en sus métricas de similitud sintáctica y semántica.

En el área de los procesos de negocio, el agrupamiento se emplea principalmente para procedimiento de comparación y clasificación de bitácoras de ejecuciones de los modelos de procesos. Las ejecuciones representan el comportamiento del proceso, indicando los recursos asociados, los actores involucrados e información del dominio y contexto.

A partir de estos elementos del proceso se realiza una comparación y se puede determinar si un proceso pertenece a uno de los siguientes grupos, **Seleccionable, No Seleccionable o No Clasificado.** El primero indica que el modelo puede ser incluido en la integración, mientras que el grupo *No Seleccionable* significa que el modelo de proceso se relaciona con situaciones que afectan su rendimiento, por tanto se recomienda no incluirlo en la integración. El grupo *No Clasificada* indica que el proceso no es coherente sintáctica y semánticamente con los otros procesos, indicando baja similitud, que llevaría a un incorrecto razonamiento.

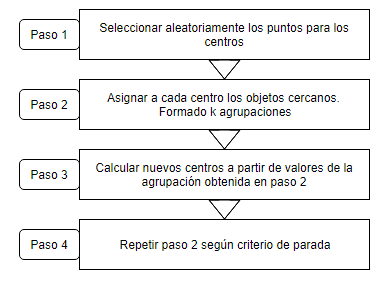


Figura 2.6: Algoritmo K-means. Fuente (Sajja, 2012).

## 2.7 Agentes Inteligentes

Un Agente de software es un sistema reactivo con algún grado de autonomía, que realiza por sí mismo las acciones necesarias para llevar a cabo las tareas asignadas. Se define como agente, debido a su capacidad de robustez, reacción y acción al ambiente por medio de sensores y actuadores (Weiss, 1999).

La Figura 2.7 presenta la interacción del agente con su ambiente, los agentes toman información por medio de sensores de ambiente, generando así percepciones, que permiten el razonamiento y toma de decisiones, produciendo acciones que se realizan en el ambiente por medio de actuadores.

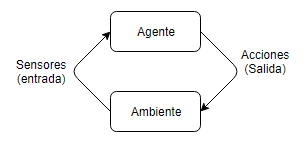


Figura 2.7: Agente inteligente y su entorno. Fuente (Weiss, 1999).

Con base en la capacidad de reacción y acción sobre el entorno, los agentes inteligentes se caracterizan por tener proactividad y habilidad social. La primera se enfoca en la definición de metas claras asignadas a un agente para que pueda tomar la iniciativa para llevar a cabo las tareas necesarias.

Por su parte la característica de habilidad social busca que el agente pueda interactuar con otros agentes, para alcanzar los objetivos planteados. La negociación, colaboración y competencia, hacen parte de las actividades que se realizan en la interacción.

Según Russel & Norvin (2004) las principales características de los agentes inteligentes son:

* Autonomía: Capacidad de realizar tareas a partir de información previa y un conocimiento específico. La información y creencias del agente permiten realizar el razonamiento y definición de planes de solución.
* Reactividad: Los agentes evalúan constantemente el ambiente buscando cambios en los valores de algunas variables, una vez es detectado el cambio se reacciona para llevar a cabo una toma de decisiones.
* Movilidad: La movilidad permite que el agente cambie de ambiente, con el fin de obtener mayor información o comparar el comportamiento de las variables en cada uno de los ambientes que visita.
* Negociación: Es la capacidad que tiene los agentes en interactuar con otros agentes o sistemas, buscando siempre el beneficio propio a partir de reglas de negocio muy estrictas, donde se busca siempre lograr la negociación. No obstante, el diseño de heurística mejora los criterios de decisión, permitiendo que los agentes negocien en distintos dominios.
* Comunicación: El intercambio de información entre agentes, comprende el uso de plantillas, vocabularios compartidos y distintos formatos en los datos. Una correcta comunicación es un factor importante en el diseño de sistemas de razonamiento distribuido basado en agentes.

Un Sistema Multi-Agente -SMA- se conoce al conjunto de agentes que interactúan entre sí para llevar a cabo una o más tareas asignadas. Así mismo, se caracterizan gestionar conocimiento e intercambiarlo por medio de performativas (i.e.primitivas del lenguaje) de comunicación (Wooldridge, 2009).

Las principales características que se aprovechan a partir de un SMA son: Negociación, cooperación, colaboración, competición, deliberación, razonamiento distribuido, entre otros, pero en especial la comunicación. Así mismo, un SMA, debe estar soportado por una arquitectura, la cual determina entre otros aspectos, los comportamientos de los agentes, los sistemas de intercambio de información y sobre todo la estrategia de razonamiento inteligente.

Según Weiss (1999), las principales arquitecturas para el diseño de Sistemas Multi-Agente son: (1) arquitectura basada lógica deductiva, (2) arquitectura basada en reacción y acción a partir de eventos (3) arquitectura basada en Deseos, Creencias e Intenciones (BDI – Belief, Desires and Intentions) y (4) arquitectura basada en capas de agrupación.

Las dos primeras arquitecturas (lógica y reactiva) centran sus soluciones al intercambio de mensajes entre agentes, potenciando la capacidad de comunicación y negociación de los SMA. Por su parte las dos últimas arquitecturas (BDI y Capas) buscan potenciar la capacidad de percepción y razonamiento distribuido de los agentes; permitiendo el agrupamiento por medio de capas.

Específicamente la Arquitectura BDI, es apropiada para el manejo de sistemas sensibles al contexto, ya que permite un mayor razonamiento empleando información del entorno y aprovechando una base de conocimiento (Creencias) un conjunto de opciones factibles (Deseos) que cambian con base a la activación de reglas de producción y un conjunto de acciones (Intenciones). La Figura 2.8 presenta el esquema general de la arquitectura BDI.

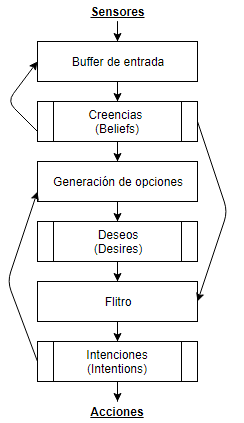


Figura 2.8: Arquitectura BDI para Sistemas Multi-Agentes. Fuente (Weiss, 1999).

Las creencias corresponden a la información que conoce el agente sobre el entorno, a partir de la entrada de los sensores, se pretende entonces que la salida del procedimiento sea una actualización de las creencias. Luego de analizar se generan opciones de solución, las cuales son vistas como los deseos del agente, lo que constituyen planes de acción alternativos. Por último, una vez se selecciones un deseo, se convierte en una Intensión, donde se pretende llevarla a cabo como objetivo principal.

Son varias las metodologías para el desarrollo de sistemas multi-agente (Marchetti et al. 2004), cada una de ellas con características y fortalezas propias para un tipo de arquitectura, no obstante, la metodología Prometheus (Padgham et al. 2005), es la indicada para el desarrollo de sistemas basado en agentes con capacidad de razonamiento sobre el contexto.

En la Figura 2.9 presenta las fases de la metodología Prometheus, la cual se compone de 3 fases a saber:

* Fase de especificación: Se encarga de la definición de las metas y las funcionalidades básicas del sistema apoyadas por las percepciones y acciones. Así mismo, se define la operación del sistema por medio de escenarios.
* Fase de diseño arquitectónico: Define el número y tipo de agentes, así mismo el mecanismo de interacción, los recursos a emplear y las restricciones de diseño. Se intenta definir plantillas para el monitoreo de los datos, así como la manera de intercambiarlos.
* Fase de diseño detallado: Se describe el funcionamiento individual de cada agente, indicando la manera en cómo alcanza sus metas y cómo interactúa con el sistema completo.

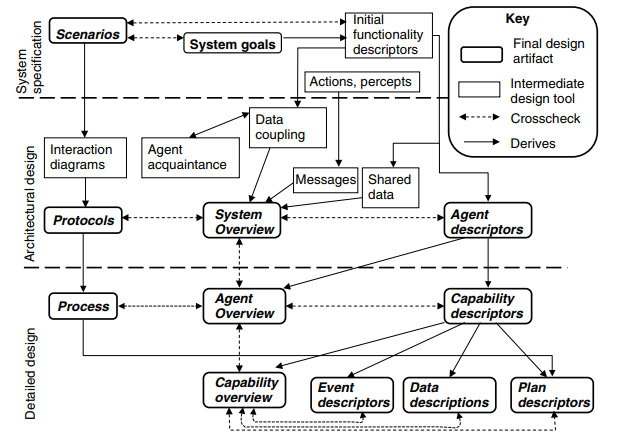


Figura 2.9: Fase de la metodología Prometheus. Tomado de (Padgham et al. 2005)

## 2.8 Arquitectura orientada a servicios

Con el avance en el desarrollo de aplicaciones basadas en patrones arquitectónicos tipo cliente/servidor, toma auge el enfoque entregar el software como un servicio, en donde el usuario del sistema consume el servicio solo cuando lo necesita. Así mismo, se facilita la diversificación de aplicaciones y su contante versionamiento, ya que dichas actividades, quedan a cargo del proveedor del servicio.

Un servicio web, se define como una aplicación modular y auto-descrita que se ejecuta en ambientes web, permitiendo la interacción con otros componentes de software a través del intercambio de mensajes vía protocolos de comunicación de internet, como los son XML/RPC y SOAP (Alonso et al., 2004).

La computación orientada a servicios centra sus fundamentos en el uso del software, más que en la adquisición (Erl, 2008), potenciando las características de los ambientes distribuidos, con el fin de proveer soluciones a usuario de manera transparente. Por tanto, la arquitectura orientada a servicios (SOA – Services Oriented Architecture) es la infraestructura que permite la construcción de servicios web y aplicaciones que consuman dichos servicios.

El modelado de aplicación basadas en SOA gira en torno a los siguientes componentes:

* Registro de servicios UDDI: Su función principal es la gestión de las descripciones WSDL de los servicios web. El registro permite el descubrimiento y recuperación de servicios, ofreciendo los metadatos necesarios para su consumo.
* Proveedor del servicio: Se encarga de la construcción y mantenimiento del servicio. Una vez el proveedor construye el servicio, debe registrar la descripción del servicio en el registro UDDI. El proveedor debe garantizar la calidad del servicio, así como las actualizaciones que se den en el tiempo.
* Cliente del servicio: Encargado de realizar las peticiones a los servicios una vez es ubicado en el registro de servicios. El cliente debe conocer la información básica del servicio para así construir su aplicación cliente que lo pueda consumir.

La Figura 2.10 presenta la interacción de los elementos descritos. Nótese que la interacción con el registro es unidireccional, permitiendo así la autogestión, por parte del proveedor y el cliente. Por su parte la interacción entre estos dos últimos componentes, se realiza de manera bidireccional, pues se basa en comunicaciones petición/respuesta descritas con el protocolo SOAP.

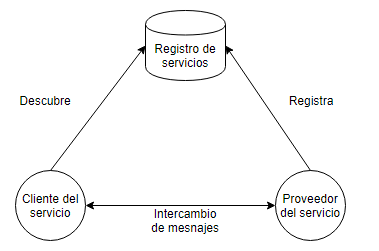


Figura 2.10: Arquitectura Orientada a servicios. Tomado de (Erl, 2008)

En el campo de los procesos de negocio, es importante resaltar la capacidad de ejecución de los servicios web en ambientes distribuidos. Cabrera et al. (2017) destacan la importancia de unir el área de los servicios web y las ontologías para su aprovechamiento en ambientes de ejecución de procesos de negocio.

## 2.9 Ontologías

Las ontologías permiten la representación por medio de conceptos de la información relacionada con un dominio de interés. Básicamente una ontología sirve para validar relaciones entre instancias de un dominio, así mismo entrega información por medio de dichas instancias indicando cuales cumplen o no ciertos valores. Según Gruber et al. (1994) una ontología se define como: “*Una especificación formal y explícita de una conceptualización compartida.*

En la Figura 2.11 se presenta la categorización de ontologías propuesta por Guarino (1998); se observa una dependencia entre la capas, donde las ontologías de alto nivel (meta-ontologías) permiten la definición de meta-vocabularios que son instanciados en ontologías de dominio y de tareas. Las ontologías de aplicaciones se enfocan en la definición de ambientes de ejecución y características tecnológicas asociadas.

El auge de las ontologías y sus aplicaciones en distintos ambientes, generó la necesidad de la definición de metodologías, lenguajes, estándares y tecnologías que guiaran de manera sistemática el desarrollo de ontologías, dando paso a lo que se conoce como la ingeniería ontológica (Antoniou & vanHarmelen, 2004).

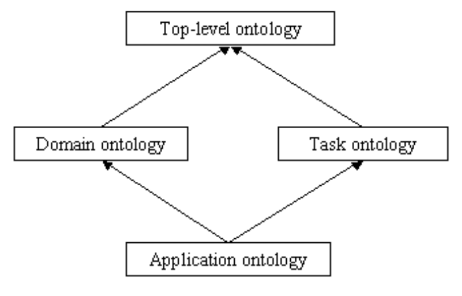


Figura 2.11: Categorización de ontologías. Tomado de (Guarino, 1998)

La ingeniería ontológica, tiene como instrumento principal las metodologías para el desarrollo de ontologías, entre las cuales se destaca “Ontology-101” (Noy & McGuinness, 2001) que facilita por medio de ocho pasos el desarrollo de ontologías, estos son:

* Determinar el alcance.
* Considerar la reutilización e integración con otras ontologías.
* Enumerar términos asociados.
* Definir taxonomía de términos.
* Definir propiedades asociadas a conceptos.
* Definir restricciones
* Crear instancias.
* Verificar incosistencias

En el campo de la BPI, una ontología permite la representación de la información contextual del dominio asociado al proceso, de esta forma facilita los mecanismos identificación de situaciones a partir del análisis de las relaciones entre las instancias del proceso. Igualmente permite comparar parejas de procesos e identificar similitudes semánticas con el fin de mejorar la búsqueda y recuperación de información.

Fan et al. (2009) presenta una aproximación de como a partir de la unificación conceptual de los procesos, se puede llevar a cabo la BPI. La Figura 2.12 la idea propuesta, la cual se caracteriza por la definición de una ontología de dominio para cada uno de los procesos, con el fin de estandarizar los conceptos y generar un solo dominio de ejecución.

Las aplicaciones de las ontologías en el área de los procesos se enfocan principalmente en las siguientes: ejecución de procesos por medio de servicios web (Aslam et. al, 2006), comparación de procesos (Kim & Suhh, 2010) y eliminación de ambigüedades (Fan et al., 2016).

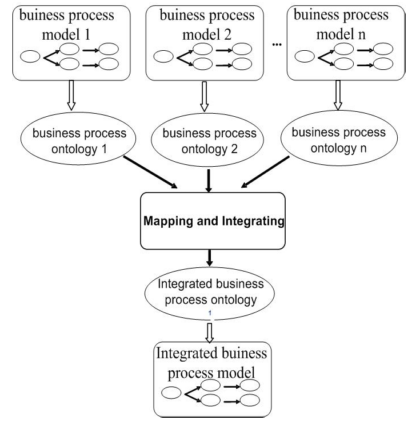


Figura 2.12: Ontologías y la BPI. Tomado de (Fan et al. 2009)

## 2.10 Conclusiones del capítulo

En este capítulo se presentaron los conceptos principales que se manejan en la presente tesis. como concepto central se encuentra la integración de procesos de negocio, seguido por la Computación sensible al contexto y su aplicación en el dominio de los procesos de negocio, problemas de verificación de restricciones, agrupamiento de datos, agentes inteligentes y la metodología Prometheus; para finalizar con web semántica y servicios web.