

NOMENCLATURA DE ONTOLOGÍAS

Este documento describe la manera en que deben ser nombrados cada uno de los elementos existentes en una ontología, tomando en cuenta elementos genéricos de un RDF, como algunos otros agregados a la ontología del producto SWB.

Dichas ontologías, son generadas en el área de nuevos productos, normalmente pensando en orientarlas a desarrollo de código de productos nuevos y o componentes que se integren a productos existentes.

Es importante recordar, que se debe pensar que los productos que desarrollamos en dicha área, pueden ir mas allá de nuestras fronteras como país, es por ello que los nombres de cada uno de los elementos que pueden componer una ontología generada por la misma área, deben ser escritos en idioma Inglés y de manera que puedan ser suficientemente descriptivos para cualquier persona que no conozca en lo absoluto el dominio de dicha ontología.

Los siguientes, son elementos en la ontología que debemos considerar al momento de nombrarlos, sin embargo, esta lista se puede seguir incrementando en la medida que se agreguen nuevos elementos al esquema de la misma.

1	Nombrado de Ontología	<p>La ontología deberá de contar con un nombre en inglés y que describa por sí mismo el dominio que pretende abarcar.</p> <p>Así mismo, el nombre de la ontología deberá empezar con una letra minúscula y con estilo CamelCase.</p> <p>Ejemplo:</p> <p>socialNet.owl</p>
2	Base Uri	<p>El URI base de las ontologías cuyo código generado se piense implementar sobre la plataforma SWBPlatform, deberá componerse de los siguientes elementos, los cuales deberán escribirse con todas sus letras en minúsculas:</p> <ul style="list-style-type: none">• Uri base de la herramienta SWB4, la cual es la siguiente:<ul style="list-style-type: none">◦ http://www.semanticwebbuilder.org/swb4/• Prefijo utilizado para el nombrado de las clases y propiedades de la ontología, cuyo nombre, deberá describir también el uso de la misma, pudiendo ser el mismo nombre de la ontología. <p>Ejemplo:</p> <p>social</p> <p>Así, un ejemplo de un URI base para un producto cuyo código generado se piense implementar en la</p>

		<p>herramienta SWB4, sería el siguiente:</p> <p>http://www.semanticwebbuilder.org/swb4/social</p>
3	Clases	<p>El nombrado de las Clases de la ontología deberá ser con estilo CamelCase y de manera que describa la finalidad de uso de la clase, de tal manera, que cualquier persona que no conozca la ontología y vea el nombre de la clase, pueda intuir el uso y funcionalidad de la misma.</p> <p>El siguiente, es un ejemplo del nombre de una clase: SocialNetwork</p>
4	Interfaces	<p>El nombrado de las interfaces de la ontología deberá ser con estilo CamelCase y de manera que describa la finalidad de uso de la interface, de tal manera, que cualquier persona que no conozca la ontología y vea el nombre de la interface, pueda intuir el uso y funcionalidad de la misma.</p> <p>Es importante mencionar el uso de la terminación “able” en el nombre de cada una de las interfaces que compongan la ontología generada, esto debido a que las ontologías creadas en Infotec, se orientan hacia el desarrollo de código en lenguaje Java.</p> <p>El siguiente, es un ejemplo del nombre de una interface: Oauthable</p>
5	Recursos	<p>El nombrado para los recursos semánticos deberá poseer las mismas características que para una clase ya descrita en este documento, sin embargo, se propone que se almacenen en un paquete que sea formado por el nombre del paquete en el que se almacenan las clases base de la ontología en general, más el nombre de paquete “resources”.</p>
6	Propiedades	<p>Para el nombrado de las propiedades, deben considerarse los siguientes aspectos:</p> <ul style="list-style-type: none"> • El nombre de una propiedad debe ser descriptivo a la información que dicha propiedad vaya a almacenar. • El nombre de las propiedades debe iniciar con una letra minúscula y posteriormente cumplir con el estilo CamelCase. • Es importante evaluar si el nombre de una propiedad que queramos declarar se podría repetir o si ya existe repetido en alguna otra clase, de ser así, debemos evaluar si ambas clases pueden compartir la misma propiedad por medio de una interfaz o si se escribe un prefijo que diferencie a una propiedad de las otras que puedan llamarse igual. • Si se refiere a una propiedad que contenga múltiples instancias de otra clase, esta debe empezar con el prefijo “has” seguido de una letra mayúscula y con estilo CamelCase. Además, es importante que el nombre de la propiedad tenga una terminación en singular, ya que el generador de código con el que se

		<p>cuenta en Infotec, detecta este tipo de nombre de propiedades y le agrega a la clase un método que lista las instancias de la propiedad, al cual se le agrega automáticamente una terminación en plural.</p> <p>Un ejemplo de una propiedad que está relacionada a múltiples instancias de otra clase (Cardinalidad Múltiple) es el siguiente:</p> <p>hasCalendar</p>
7	Propiedades Inversas	<p>En caso de que una propiedad requiera tener una inversa, el nombre para dicha inversa, deberá seguir las mismas especificaciones que para una propiedad normal (véase el apartado anterior), seguido por la terminación “Inv”, para indicar, que se trata de una propiedad inversa.</p> <p>El siguiente, es un ejemplo del nombre de una propiedad inversa:</p> <p>hasCalendarRefInv</p>
8	Propiedad de Despliegue (displayProperty)	<p>La propiedad de despliegue de una propiedad, debe tener el mismo nombre que dicha propiedad, más el prefijo “dp_” (display property).</p> <p>El siguiente, es un ejemplo del nombre de una propiedad de despliegue, considerando que el nombre de la propiedad a la que se le agrega una propiedad de despliegue es “photo”:</p> <p>dp_photo</p>
9	Grupo de Propiedades (propGroup)	<p>El nombre de este elemento debe ser descriptivo de acuerdo al tipo de propiedades que se vayan a agrupar en el mismo.</p> <p>Además, debe iniciar con una letra minúscula y posteriormente cumplir con el estilo CamelCase.</p> <p>Se propone, que el nombre de este elemento contenga, además del nombre descriptivo mencionado anteriormente, un prefijo “pg_”.</p> <p>El siguiente, es un ejemplo del nombre de un grupo de propiedades:</p> <p>Pg_generalData</p>
10	Elemento de Forma (FormElement)	<p>Los identificadores de los elementos de forma deben ser llamados considerando los siguientes aspectos:</p> <ul style="list-style-type: none"> • De forma descriptiva a la funcionalidad del tipo de elemento que se vaya a implementar programáticamente. • Debe cumplir con el estilo CamelCase, es decir, como cualquier otro nombre de clase. • Debe empezar con el prefijo “fe_”. <p>El siguiente, es un ejemplo del nombre de un elemento de forma, considerando los aspectos mencionados:</p> <p>fe_FileUpload</p> <p>Si lo que se desea es crear una instancia de un FormElement ya existente, de tal manera que el generador de la ontología pueda modificar los valores de las propiedades de la clase, las reglas para el nombrado</p>

		de dicha instancia, deberán ser las mismas que para el nombrado de la clase.
11	Paquete Base de Clases	El nombre del paquete base en el que se generará código, basado en swbPlatform, debe ser el siguiente: org.semanticwb