

## NOMENCLATURA DE ONTOLOGÍAS

Propósito: Guiar la forma en que se nombran los elementos que conforman una ontología.

Una Ontología es una manera formal de representar computacionalmente lo que existe en el mundo real y son generadas en el área de nuevos productos con la finalidad de orientarlas a desarrollo de código para productos nuevos y o componentes que se integren a productos existentes.

A continuación se listan algunos de los elementos que debemos considerar al momento de asignar la nomenclatura de ontologías.

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"><li>• Uri base de la herramienta SWB4, la cual es la siguiente:<ul style="list-style-type: none"><li>◦ <a href="http://www.semanticwebbuilder.org/swb4/">http://www.semanticwebbuilder.org/swb4/</a></li></ul></li><li>• 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.</li></ul> <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 herramienta SWB4, sería el siguiente:</p> <p><a href="http://www.semanticwebbuilder.org/swb4/social">http://www.semanticwebbuilder.org/swb4/social</a></p>
3	Clases	<p>Una clase es un tipo de concepto, puede verse como un agrupador o una categoría para varios conceptos. El nombrado de las Clases de la ontología deberá ser con estilo CamelCase y describiendo la finalidad de uso de la</p>

		<p>clase, de tal manera, que pueda intuirse el uso y funcionalidad de la misma.</p> <p>El siguiente, es un ejemplo del nombre de una clase: SocialNetwork</p>
4	Interfaces	<p>Las interfaces definen propiedades asociadas a un comportamiento específico y se utilizan como mecanismo de "Herencia Múltiple" de propiedades. 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, con la finalidad que pueda ser intuitiva 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>Un recurso semántico es un componente de un sitio web o una aplicación que permite tener interacción con un usuario. 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". Ejemplo: org.semanticwb.social.base.resources</p>
6	Propiedades	<p>Para el nombrado de las propiedades, deben considerarse los siguientes aspectos:</p> <ul style="list-style-type: none"> <li>• El nombre de una propiedad debe ser descriptivo al dominio que dicha propiedad vaya a almacenar.</li> <li>• El nombre de las propiedades debe iniciar con una letra minúscula y posteriormente cumplir con el estilo CamelCase.</li> <li>• Deberá evaluarse si el nombre de una propiedad podría repetirse o si ya existe en alguna otra clase, de ser así, deberá crearse dentro de una interfaz y compartirse.</li> <li>• Si la propiedad contiene múltiples instancias de otra clase, esta debe empezar con el prefijo "has" seguido de una letra mayúscula y con estilo CamelCase. Es importante que el nombre de la propiedad tenga una terminación en singular, ya que el generador de código agrega automáticamente una terminación en plural.</li> </ul> <p>Un ejemplo de una propiedad que está relacionada a múltiples instancias de otra clase (Cardinalidad Múltiple) es el siguiente: 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: 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”: 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: Pg_generalData</p>
10	Elemento de Forma (FormElement)	<p>Los identificadores de los elementos de forma deben ser nombrados considerando los siguientes aspectos:</p> <ul style="list-style-type: none"> <li>• De forma descriptiva a la funcionalidad del tipo de elemento que se vaya a implementar programáticamente.</li> <li>• Debe cumplir con el estilo CamelCase, es decir, como cualquier otro nombre de clase.</li> <li>• Debe empezar con el prefijo “fe_”.</li> </ul> <p>El siguiente, es un ejemplo del nombre de un elemento de forma, considerando los aspectos mencionados: 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 de dicha instancia, deberán ser las mismas que para el nombrado de la clase.</p>
11	Paquete Base de Clases	<p>El nombre del paquete base en el que se generará código, basado en swbPlatform, debe ser el siguiente: org.semanticwb</p>
12	Modelos	<p>El nombrado de este elemento deberá representar el dominio que define.</p>