

Título
EL MÁGICO MUNDO DE LA PROGRAMACIÓN FUNCIONAL EN HASKELL
Resumen
<p>Programar es el acto de expresar, de forma que sea comprensible por un computador, el proceso para calcular alguna función de interés. En su esencia, la computación no es sino matemática en movimiento y, como tal, puede aprovechar una gran parte de la abstracción que el pensamiento matemático ofrece. Una gran cantidad de herramientas y tecnologías existe para ayudar a los programadores a plasmar sus ideas y permitir que estas sean comprendidas por computadores. Es nuestra responsabilidad como profesionales del área mantenernos lo más al día posible, explorando las diferentes opciones y aprendiendo siempre nuevas formas de trabajar. De esta forma, siempre podremos seleccionar la herramienta más adecuada al enfrentarnos a problemas de la vida real. Sin embargo, muchos programadores manejan únicamente lenguajes imperativos como C, Java, Python, PHP o Javascript. Este tutorial busca presentar una alternativa, la programación funcional, a través de su más notable miembro en la actualidad: el lenguaje Haskell. La programación funcional permite escribir programas muy concisos y eficientes, con facilidades para la creación y el manejo de estructuras de datos de todo tipo, acercando un poco más la computación al nivel del razonamiento matemático abstracto. Definitivamente, es una herramienta poderosa que todo programador debe poseer.</p>
Contenido
<p>Parte 1 (Introducción): Breve revisión histórica de la evolución de los lenguajes de programación. Bases teóricas de la programación funcional. El <i>Haskell Platform</i> y <i>ghci</i>. Funciones y tipos, secciones, curriificación. <i>Pattern Matching</i> y condicionales. Listas y tuplas. Funciones de orden superior. Creación de tipos de datos recursivos. Módulos. Ejemplo completo: Codificación de Hoffman.</p> <p>Parte 2 (Avanzado): Morfismos: <i>fold</i>, <i>unfold</i> y <i>refold</i>. Contenedores: <i>Applicative</i>, <i>Functor</i> y <i>Monad</i>. Ejemplos de <i>Monads</i>: <i>Identity</i>, <i>Maybe</i> y <i>Lista</i>. Notación “<i>do</i>”. Ejemplos de <i>Monads</i>: <i>Writer</i>, <i>Reader</i>, <i>State</i> e <i>IO</i>. <i>Parsec</i>. Transformadores de <i>Monads</i>. Ejemplo Completo: Máquina Virtual de Logo. Más allá de Haskell, el mundo de programadores funcionales.</p>
Duración
Un día completo
Público Objetivo
Profesionales o estudiantes de Computación, Informática o Sistemas que tengan curiosidad en aprender nuevos paradigmas. Es recomendable que se tenga experiencia programando en al menos un lenguaje y conocimiento básico de algoritmos y estructuras.
Equipos Requeridos
<p>Para el presentador: Un proyector y una pizarra (preferiblemente acrílica).</p> <p>Para el presentador y los asistentes: Un computador (preferiblemente con sistema operativo Linux), con la última versión del <i>Haskell Platform</i> instalada y algún editor de texto (preferiblemente al estilo de <i>gedit</i> o <i>Notepad++</i>).</p>
Breve CV de los presentadores
<p>Ricardo Monascal. Magister en Ciencias de la Computación, Universidad Simón Bolívar (USB), Caracas, Venezuela (2014). Ingeniero en Computación, Universidad Simón Bolívar (USB), Caracas, Venezuela (2009). Miembro del personal académico de la USB (desde 2009). Actualmente Profesor Asistente (desde 2014). Miembro de la sección de Algoritmos y Lenguajes.</p> <p>Entrenador de los equipos participantes de la USB en los maratones de programación ACM-ICPC (desde 2010). Investigador en el área de semántica formal y lógica, con la guía del prof. Jesús Ravelo (desde 2010). Miembro del grupo de investigación en fundamentos de la matemática y la computación, con la guía del prof. Jorge Baralt (desde 2012). Miembro del grupo de investigación en bases de datos difusas, con la guía del prof. Leonid Tineo (desde 2014). Miembro de la (SVC) Sociedad Venezolana de Computación (desde 2014).</p>