



**UNIVERSIDAD SIMÓN BOLÍVAR**  
**COORDINACIÓN DE COMPUTACIÓN**

**DESCRIPCIÓN DE PROYECTO Y ETAPAS**

**TÍTULO DEL PROYECTO:**

Apuntadores y Tipos de Datos Algebraicos para GraCieLa

**NO. DEL PROYECTO:**

**CÓDIGO DE ÁREA:**

**FECHA:**

04/03/2016

**PROPUESTO POR:**

Ricardo Monascal & Ernesto Hernández-Novich

**1. RESULTADOS ESPECÍFICOS A LOS QUE SE QUIERE LLEGAR (EL PRODUCTO QUE SE DESEA OBTENER) Y SU POSIBILIDAD DE APLICACIÓN:**

La enseñanza de algoritmos se beneficia ampliamente al estar sustentada sobre lenguajes de programación con sistemas de tipos estrictos, que reduzcan la cantidad de errores simples cometidos por el estudiante, pero al mismo tiempo con el empleo de sistemas formales basados en semántica axiomática que permitan al estudiante razonar sobre los algoritmos que implanta.

El objetivo central de este proyecto es extender el compilador del lenguaje GraCieLa, una variante del GCL (*Guarded Command Language*) de Dijkstra, implantado por Araujo y Jiménez.

Esta extensión deberá soportar tipos de datos definidos por el programador, apuntadores explícitos, manejo de memoria automático, y un mecanismo de aserciones verificables que soporte estas nuevas capacidades.

**2. ACTIVIDADES QUE INVOLUCRA EL PROYECTO:**

- Revisión bibliográfica relacionada con implantación de compiladores, sistemas de tipos y semántica axiomática.
- Especificación formal de la extensión a GraCieLa a implantar.
- Evaluación de herramientas que faciliten la construcción del ambiente de ejecución final para manejo automático de memoria.
- Implantación del compilador de la extensión de GraCieLa a código de máquina nativo.
- Extensión del manual de usuario con las nuevas funcionalidades.

**3. PUNTOS DE INTERÉS QUE HAN DE SER TRATADOS DURANTE LA EJECUCIÓN DEL PROYECTO:**

- Estudio de generadores de código multi-destino existentes.
- Estudio de alternativas para proveer herramientas de apoyo al estudiante de algoritmos que usa el lenguaje (*debugger, profiler, cross-referrer*).
- Estudio de alternativas para la generación de código de tipos de datos definidos por el programador.
- Estudio de alternativas para el manejo automático de memoria.
- Estudio de alternativas para la formulación y verificación de aserciones en presencia de apuntadores.

**4. ÁREAS O CAMPOS DE LA TÉCNICA QUE SE ABORDA EN EL DESARROLLO DEL PROYECTO:**

- Lenguajes de Programación.

**5. ESTIMACIÓN DE LA DURACIÓN DEL PROYECTO Y RECURSOS REQUERIDOS:**

El desarrollo del proyecto contempla aproximadamente tres (3) períodos académicos (1 año) con dos estudiantes por período, para un total de 3 etapas. Se contará con máquinas del LAL (Laboratorio de Algoritmos y Lenguajes) así como de computadores personales.		
<b>DESCRIPCIÓN DE LAS ETAPAS</b>		
<b>6. NOMBRE DE LA PRIMERA ETAPA (EP-1308):</b> Revisión bibliográfica y refinamiento del front-end		<b>TRIMESTRE/AÑO:</b> Abril-Julio 2016
<b>6.1. RESULTADOS ESPECÍFICOS MÍNIMOS A ALCANZAR EN ESTA ETAPA:</b> <ul style="list-style-type: none"> <li>Especificación formal de la extensión a GraCieLa a implantar.</li> <li>Extensión del front-end del compilador.</li> </ul>		
<b>6.2. ACTIVIDADES QUE SE PROPONE REALIZAR PARA LOGRAR ESTOS RESULTADOS:</b> <ul style="list-style-type: none"> <li>Evaluar las recomendaciones del jurado del proyecto de grado de Araujo y Jiménez sobre la semántica del lenguaje propuesta por ellos.</li> <li>Revisión de la bibliografía existente para el manejo de tipos definidos por el usuario en el contexto de programación formal.</li> <li>Implantación de un Lexer y un Parser extendidos que soporten las nuevas funcionalidades propuestas.</li> </ul>		
<b>6.3. DESCRIPCIÓN DE TÓPICOS EN INGENIERÍA DE COMPUTACIÓN I (EP5855):</b> En esta asignatura el estudiante deberá realizar la revisión bibliográfica de implantación de lenguajes de programación existentes y sus sistemas de tipos, combinarlos con la especificación de GraCieLa de Araujo y Jiménez, para producir la definición formal de la extensión a implantar. Además, deberá implantar el front-end de la extensión, incluyendo todas las verificaciones de contexto tanto para los programas escritos en el lenguaje, como para los meta-programas (aserciones) presentes.		
<b>7. NOMBRE DE LA SEGUNDA ETAPA (EP2308):</b> Implantación del back-end.		<b>TRIMESTRE/AÑO:</b> Septiembre-Diciembre 2016
<b>CURSARÁ (EP-2308) Y (EP-3308) JUNTAS</b> <b>SÍ</b> <b>NO</b> <input checked="" type="checkbox"/>		
<b>7.1. RESULTADOS ESPECÍFICOS MÍNIMOS A ALCANZAR EN ESTA ETAPA:</b> <ul style="list-style-type: none"> <li>Implantación inicial del back-end para el compilador, asegurando la emisión de código nativo correcto para la extensión de GraCieLa.</li> <li>Colección de programas de prueba que ejerciten las capacidades del lenguaje.</li> </ul>		
<b>7.2. ACTIVIDADES QUE SE PROPONE REALIZAR PARA LOGRAR ESTOS RESULTADOS:</b> <ul style="list-style-type: none"> <li>Revisión bibliográfica de las herramientas de back-end multiplataforma existentes con soporte para manejo de memoria dinámica y tipos de datos definidos por el programador y de mejoras al código intermedio.</li> <li>Extender el front-end para emitir código intermedio independiente de máquina susceptible de aprovechar las herramientas de back-end existentes.</li> <li>Determinar las mejoras a código intermedio aplicables directamente a través de las herramientas de back-end seleccionadas.</li> <li>Aprovechar las herramientas de back-end para generar código nativo.</li> </ul>		
<b>7.3. DESCRIPCIÓN DE TÓPICOS EN INGENIERÍA DE COMPUTACIÓN II (EP5856):</b> En esta asignatura el estudiante deberá investigar las herramientas existentes para la construcción de back-ends para compiladores, que incluyan la manipulación y mejora de código intermedio, así como la emisión de código de máquina. Explorar las alternativas de implantación de manejo de memoria dinámica y tipos de datos definidos por el programador. Integrará el front-end con el back-end para conseguir la generación de código, y activará o incluirá mejoras al código intermedio y final para		

mejorar el desempeño de los programas. Se comprobará la funcionalidad del lenguaje (sin incluir las aserciones) escribiendo programas de muestra que cubran los algoritmos habituales en la enseñanza.	
<b>8. NOMBRE DE LA TERCERA ETAPA (EP-3308):</b> Refinamiento del ambiente de ejecución.	<b>TRIMESTRE/AÑO:</b> Enero-Marzo 2017
<b>8.1. RESULTADOS ESPECÍFICOS MÍNIMOS A ALCANZAR EN ESTA ETAPA:</b>	
<ul style="list-style-type: none"> <li>● Finalizar la implantación del back-end para el compilador, asegurando la emisión correcta de código nativo para la extensión del lenguaje GraCieLa y del meta-lenguaje de aserciones de semántica axiomática.</li> <li>● Familia de pruebas para la extensión de GraCieLa y el meta-lenguaje de aserciones.</li> <li>● Manual de Usuario del Lenguaje y del Compilador.</li> <li>● Informe final.</li> </ul>	
<b>8.2. ACTIVIDADES QUE SE PROPONE REALIZAR PARA LOGRAR ESTOS RESULTADOS:</b>	
<ul style="list-style-type: none"> <li>● Incorporar la emisión de código para el meta-lenguaje de aserciones.</li> <li>● Explorar e incorporar facilidades provistas por las herramientas de back-end para proveer un depurador y un perfilador al lenguaje.</li> <li>● Elaboración del Manual de Usuario del Lenguaje y del Compilador.</li> <li>● Elaboración del informe final.</li> </ul>	
<b>9. SI LAS ACTIVIDADES DE ESTAS ETAPAS DEPENDEN DE RECURSOS MATERIALES, INDIQUE QUÉ PREVISIONES SE HAN TOMADO PARA LOGRAR LA DISPONIBILIDAD:</b>	
<b>TUTOR ACADÉMICO:</b> Ernesto Hernández-Novich	<b>PROF. GUÍA:</b>
<b>GRUPO DE TRABAJO:</b>	<b>CARNET:</b>
1) Moisés Ackerman	11-10005
2) Carlos Spaggiari	11-10987
3)	
<b>FIRMA DEL TUTOR ACADÉMICO:</b>	
<b>FIRMA DEL COORDINADOR:</b>	