

---

# Projeto VHS-AM

---

Proposta Preliminar para Trabalho de Graduação



**André Wilson Brotto Furtado**  
*awbf@cin.ufpe.br*  
21 de janeiro de 2004

## 1. Título

“VHS-AM: Extensão do Visual Studio .NET para a integração de Haskell com a ferramenta Assignment Manager”

## 2. Área

Engenharia de Software e Linguagens de Programação

## 3. Orientador

Professor André Luís de Medeiros Santos  
(*alms@cin.ufpe.br*)

## 4. Resumo

A linguagem funcional Haskell é consolidada como uma das principais linguagens em áreas que empregam o paradigma de programação funcional. Entretanto, é reconhecido que algumas limitações provocam uma sub-utilização do potencial da linguagem. A timidez na integração com ferramentas de mercado e a ausência de ambientes de desenvolvimento, objetivando tornar a etapa de programação mais produtiva e menos sujeita a erros, são duas das principais restrições identificadas. Este trabalho, portanto, pretende oferecer um ambiente de programação que dê suporte à utilização prática e produtiva de Haskell, através do modelo de extensão oferecido pela IDE Visual Studio .NET. Além disso, será validada a integração da IDE estendida com a ferramenta Assignment Manager, disseminando o uso e ampliando o campo de aplicação de Haskell. As atividades a serem realizadas envolvem não apenas atividades técnicas, como também de planejamento e gerenciamento da equipe do projeto VHS-AM, além de articulações com um outro projeto da Microsoft (Cambridge) de objetivos similares. A metodologia PRO.NET será utilizada para a condução formal do trabalho.

## 5. Contexto

### 5.1 Surgimento do Projeto VHS-AM e sua relação com este Trabalho de Graduação

Em um dado momento, a idéia original para este Trabalho de Graduação consistia apenas na integração do VS.NET com a linguagem funcional Haskell, o que seria realizado em parceria com a Microsoft Research Cambridge. A esta parceria conjunta foi dado o nome de VHS (Visual Haskell Studio). Entretanto, devido a uma oportunidade captada através da Microsoft Shared Source Initiative e de um *request for proposals* (RFP) da própria Microsoft, foi possível inserir este Trabalho de Graduação em um contexto maior, tornando-o, portanto, um subconjunto do projeto VHS-AM.

A RFP enviada mundialmente pela Microsoft convocava desenvolvedores para a extensão da ferramenta Assignment Manager, que gerencia, para professores e alunos, o processo de publicação e submissão de tarefas acadêmicas que envolvem codificação. O aluno deste Trabalho de Graduação ofereceu sugestões para a proposta inicialmente elaborada pelo seu orientador, em resposta ao RFP, contribuindo, portanto, para a versão final da mesma. Em resumo, a proposta enviada consistiu na extensão da ferramenta Assignment Manager para que a mesma suportasse projetos desenvolvidos na linguagem Haskell, além de alterações no próprio VS.NET que também objetivassem o suporte a Haskell. Adicionalmente, a proposta incluiu a especificação de uma equipe de 5 integrantes para a realização do projeto, sobre a qual o aluno deste Trabalho de Graduação deveria assumir, além das originais tarefas de desenvolvimento, atividades de coordenação e gerência de projeto, sendo, portanto, o responsável pela condução formal do mesmo através de uma metodologia de desenvolvimento de software bem-definida: a PRO.NET.

Como é possível perceber, o projeto VHS-AM possui uma importante interseção com o projeto VHS original, havendo se estabelecido o desafio de combinar e sincronizar os esforços dos dois projetos para garantir a satisfação dos objetivos de ambos da maneira mais produtiva possível.

Observou-se, portanto, que a participação do aluno nos projetos VHS e VHS-AM reunia os requisitos necessários à realização de um Trabalho de Graduação para o CIn/UFPE, dentre os quais é possível citar:

- **Pesquisa sobre o estado-da-arte** referente aos mecanismos de extensão do Visual Studio .NET para o suporte a novas linguagens;
- **Atividades não-triviais de implementação** demandadas pela integração de diferentes conceitos (programação funcional, noções de compiladores, etc.) ferramentas (VS.NET, AM, GHC, etc.) e tecnologias (componentes COM, modelo de extensão do VS.NET, etc.);
- Contribuição para a realização de um **trabalho teórico** referente à análise da adequação da recente metodologia PRO.NET a um projeto externo ao Centro de Tecnologia XML, experiência inédita até então.

## 5.2 Haskell: uma linguagem de virtudes e deficiências

Milhares de linguagens de programação existem atualmente. Entre elas, há uma que se destaca por fazer parte das principais tecnologias utilizadas em áreas que empregam o paradigma de programação funcional: a linguagem Haskell. Apesar de ser bastante difundida neste nicho específico, a linguagem funcional Haskell enfrenta atualmente algumas limitações que impedem uma maior popularização e o aumento de seu campo de aplicação, como performance, integração com outras tecnologias, conjunto limitado de bibliotecas e ausência de ferramentas que tornam o processo de implementação em Haskell mais produtivo, como ambientes integrados de desenvolvimento. Acreditando na linguagem Haskell como uma alternativa concreta para a implementação de aplicações em larga escala, é exatamente através da resolução dessa última deficiência que este projeto se propõe a contribuir para a comunidade Haskell.

## 5.3 A demanda por IDEs

Na mesma época do surgimento das linguagens de baixo e médio nível, foi consolidada a maneira padrão utilizada até hoje por desenvolvedores para programar uma aplicação: a escrita de código-fonte, que substituiu o uso de cartões perfurados e fios elétricos. Nesses estágios iniciais, portanto, a etapa de implementação de um processo de desenvolvimento de software envolvia ferramentas independentes, utilizadas ciclicamente: um editor de texto simples (como o Vi), o compilador da linguagem utilizada e um mecanismo de depuração trivial, quando não a inserção de comandos de impressão de mensagens no próprio código.

As primeiras IDEs nasceram com o objetivo de integrar as ferramentas mais utilizadas pelos desenvolvedores, oferecendo inicialmente apenas as três funcionalidades básicas do ciclo de implementação de uma aplicação: edição, compilação e depuração. Com o tempo, atendendo às crescentes necessidades dos desenvolvedores, as IDEs passaram a oferecer conjuntos muito mais ricos de funcionalidades, como o suporte a diferentes tipos de arquivos (não só texto como também gráficos), analisadores estatísticos, compiladores incrementais, “navegadores” de documentos e a identificação da dependência entre os mesmos. Acompanhando o surgimento de diversas tecnologias da computação, as necessidades dos desenvolvedores aumentaram ainda mais, demandando, conseqüentemente, uma evolução contínua das IDEs e do conjunto de funcionalidades oferecido pelas mesmas. Se, antigamente, o modelo de desenvolvimento era bastante simples (códigos-fonte escritos em uma única linguagem com o objetivo de rodar em apenas uma plataforma específica), hoje em dia o cenário é extremamente diferente.

Atualmente, a popularidade e utilização de uma linguagem de programação estão diretamente ligadas à produtividade oferecida ao programador através de IDEs que suportam a linguagem. Um exemplo típico é o Visual Studio .NET 2003, desenvolvido pela Microsoft, sem o qual o programador teria que implementar uma quantidade praticamente inviável de código para a construção/consumo de Web Services ou acesso a dados através do ADO.NET. Resumindo, um processo de programação suportado por uma IDE é requisito fundamental para uma real difusão de uma linguagem de programação.

## 5.4 Abordagem utilizada

Este projeto não criará uma nova IDE para Haskell, mas estenderá o Visual Studio .NET 2003 para suportar a linguagem. É notável como a arquitetura desta IDE foi concebida levando sua própria extensibilidade em consideração. O projeto VSIP (Visual Studio Industry Partner), por exemplo, é um programa da Microsoft que permite a extensão do Visual Studio .NET 2003, disponibilizando o acesso às interfaces utilizadas na construção do próprio Visual Studio .NET 2003. Este programa, juntamente com o código fonte da ferramenta Assignment Manager (obtida através do Microsoft Shared Source Initiative), serão os principais instrumentos de trabalho a serem utilizados neste Trabalho de Graduação.

## 6. Plano de Trabalho

Como indicado anteriormente, o projeto VHS-AM é um projeto do Centro de Informática submetido e aprovado pela Microsoft, em resposta ao Request for Proposals (RFP) relativo à extensão da ferramenta Assignment Manager, divulgado através do programa Microsoft Shared Source Initiative. Deste modo, as atividades a serem executadas pelo aluno no Trabalho de Graduação aqui proposto consistem em um subconjunto das atividades totais do projeto VHS-AM. Entretanto, por simplicidade, o Plano de Trabalho aqui apresentado consiste no projeto VHS-AM como um todo. A subseção **“Subconjunto das atividades a serem executadas pelo aluno”** conclui que a finalização deste Trabalho de Graduação não implica na conclusão do projeto VHS-AM.

### 6.1 Visão do projeto VHS-AM

“Suportar o uso produtivo de Haskell através da extensão da IDE Visual Studio .NET e integração com a ferramenta acadêmica Assignment Manager, disseminando o uso da linguagem e ampliando seu campo de aplicação.”

### 6.2 Requisitos do projeto VHS-AM

Nessa seção estão descritos os principais requisitos da aplicação a ser desenvolvida, de acordo com a classificação abaixo:

- **Essencial:** Essencial é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante:** Importante é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável:** Desejável é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Requisito	Categoria
1. Suporte à edição de código Haskell (Syntax highlighting, IntelliSense, etc.).	Essencial
2. <i>Template</i> para criação de um projeto do tipo “Haskell Console Application”	Essencial
3. Integração com o Assignment Manager	Essencial
4. <i>Template</i> para a adição de módulos Haskell a projetos “Haskell Console Application”	Essencial
5. Análise estática e dinâmica de código <i>on-the-fly</i>	Essencial
6. Module viewer	Essencial
7. Integração com o <i>lexer/parser</i> do GHC	Importante
8. Ligação de mensagens de erro ao código-fonte	Importante
9. Ir à definição de um termo ( <i>go to definition</i> )	Importante
10. Adição de comentários (TODO/HACK/UNDONE) à <i>tasklist</i> a partir de código Haskell.	Importante
11. Navigation bar	Desejável
12. GHCi executado em uma <i>tool window</i> à parte	Desejável

### 6.3 Deliverables do projeto VHS-AM

1. Conjunto de pacotes de extensão do VS.NET contemplando os requisitos exigidos.
2. Código modificado do Assignment Manager contemplado os requisitos exigidos.
3. Manual de instalação dos pacotes desenvolvidos.
4. Guia contendo dicas e lições aprendidas relacionadas à extensão do VS.NET para o suporte de novas linguagens e integração com o Assignment Manager.

### 6.4 Infra-estrutura do projeto VHS-AM

O laboratório C6 será utilizado para montar o ambiente de desenvolvimento e de testes do projeto VHS-AM. A função específica de cada máquina do ambiente de desenvolvimento segue abaixo .

Hardware	Software	Finalidade
<b>Nome da máquina:</b> c6c01 <b>Configuração:</b> Athlon XP 2000+ 786Mb <b>SO:</b> Windows 2000 Professional	VS.NET 2003 VSIP VSIP Extras Cygwin	Suportar desenvolvimento de VSPackages
<b>Nome da máquina:</b> c6c02 <b>Configuração:</b> Athlon XP 2000+ 786Mb <b>SO:</b> Windows 2000 Professional	AM Server	Servidor AM
<b>Nome da máquina:</b> c6c06 <b>Configuração:</b> Athlon XP 2000+ 786Mb <b>SO:</b> Windows 2000 Professional	AM Faculty Client	Simular o uso do Professor AM
<b>Nome da máquina:</b> c6c03 <b>Configuração:</b> Athlon XP 2000+ 786Mb <b>SO:</b> Windows 2000 Professional	AM Student Client	Simular o uso do Aluno AM

## 6.5 Papéis e responsabilidades do projeto VHS-AM

A tabela abaixo revela que o aluno envolvido neste Trabalho de Graduação é apenas um membro de um total de 5 que conduzirão o projeto. O foco das atividades do aluno será:

- Planejamento e Gerenciamento do projeto:
  - Conduzir a definição da Visão do projeto;
  - Elaborar Especificação Funcional;
  - Elaborar Plano de Projeto;
  - Elaborar Cronograma do Projeto;
  - Gerenciar riscos;
  - Gerenciar comunicação interna e externa;
  - Gerenciar mudanças;
  - Alocação de atividades aos demais membros.
- Coordenação do processo de desenvolvimento:
  - Desenvolver funcionalidades com prioridade à implementação (em Haskell) de pacotes de extensão do VS.NET (VSPackages);
  - Conduzir de validações tecnológicas;
  - Integrar código dos demais membros do projeto;
  - Auxiliar o processo de configuração do ambiente.

Nome	Papel (metodologia Pro.NET)	Contato
André Santos	<ul style="list-style-type: none"><li>• Gerência de Produto</li><li>• Testes</li></ul>	<a href="mailto:alms@cin.ufpe.br">alms@cin.ufpe.br</a>
Paulo Borba	<ul style="list-style-type: none"><li>• Experiência do Usuário</li><li>• Testes</li></ul>	<a href="mailto:phmb@cin.ufpe.br">phmb@cin.ufpe.br</a>
<b>André Furtado</b>	<ul style="list-style-type: none"><li>• <b>Desenvolvimento</b></li><li>• <b>Gerência de Programa</b></li></ul>	<a href="mailto:awbf@cin.ufpe.br">awbf@cin.ufpe.br</a>
Mauro Araújo	<ul style="list-style-type: none"><li>• Desenvolvimento</li><li>• Gerência de Release</li></ul>	<a href="mailto:mscla@cin.ufpe.br">mscla@cin.ufpe.br</a>
Marden Mezenes	<ul style="list-style-type: none"><li>• Desenvolvimento</li><li>• Experiência do Usuário</li></ul>	<a href="mailto:mmc3@cin.ufpe.br">mmc3@cin.ufpe.br</a>

## 6.6 Subconjunto das atividades a serem executadas pelo aluno

Devido ao prazo do Trabalho de Graduação (início de abril) ser anterior ao prazo do projeto VHS-AM (15 de junho), as atividades a serem executadas pelo aluno no Trabalho de Graduação serão exatamente as atividades para as quais ele foi alocado no projeto VHS-AM até o início de abril. Deste modo, espera-se que, na época de finalização do Trabalho de Graduação, o segundo Release Interno do projeto VHS-AM tenha sido lançado e testado. Isto significa que na finalização deste Trabalho de Graduação apenas um subconjunto dos requisitos do projeto VHS-AM terá sido implementado e apenas um subconjunto dos *deliverables* estará concluído. Em resumo, portanto, apesar da finalização do Trabalho de Graduação, o aluno ainda estará mais dois meses e meio envolvido com a evolução do projeto VHS-AM.

## 7. Cronograma

Uma versão inicial do cronograma deste Trabalho de Graduação segue abaixo. Como as atividades deste Trabalho de Graduação consistem em um subconjunto das atividades gerais do projeto VHS-AM, o cronograma geral do projeto VHS-AM consiste numa extensão do cronograma apresentado abaixo. O cronograma completo do projeto VHS-AM pode ser encontrado em [www.cin.ufpe.br/~haskell/vhs-am](http://www.cin.ufpe.br/~haskell/vhs-am).

**notação para status das atividades (SA) em 21 de janeiro de 2004**

	<i>completa</i>		<i>em andamento</i>		<i>não iniciada</i>
---	-----------------	---	---------------------	--	---------------------

Fase de Visão			
SA	%	Evento	Prazo
	100	Reunir equipe do projeto	15 Dez 2003
	100	Elaborar Lista de Riscos	20 Dez 2003
	100	Elaborar o Documento de Visão/Escopo	23 Dez 2003
	100	Elaborar site do projeto	30 Dez 2003
	100	Estabelecer articulação com a MS @ Cambridge	30 Dez 2003
Fase de Planejamento			
SA	%	Evento	Prazo
	100	Elaborar Especificação Funcional	05 Jan 2004
	100	Elaborar Plano de Projeto	12 Jan 2004
	100	Elaborar Cronograma do Projeto	14 Jan 2004
	80	Configurar ambiente de desenvolvimento	21 Jan 2004
	90	Validação tecnológica: COM + Haskell	30 Jan 2004
	50	Validação tecnológica: VSIP	30 Jan 2004
	100	Validação tecnológica VSIP Extras Beta	03 Fev 2004
	30	Análise de código da MS @ Cambridge	06 Fev 2004
	70	Validação tecnológica : SML.NET/MyC	06 Fev 2004
	20	Configurar ambiente de testes	11 Fev 2004
	0	Elaborar prova de conceito	12 Fev 2004

Fase de Desenvolvimento			
SA	%	Evento	Prazo
	40	Completar Release Interno 1	16 Fev 2004
	0	Revisão de código	16 Fev 2004
	0	Testar Release Interno 1	16 Fev 2004
	0	Completar Release Interno 2	15 Mar 2004
	0	Revisão de código	15 Mar 2004
	0	Testar Release Interno 2	15 Mar 2004
	0	Elaborar monografia e apresentação	22 Mar 2004
	0	Revisão da monografia com orientador	23 Mar 2004

## 8. Referências

- [1] Sebesta R. W., *Concepts of Programming Languages*, The Benjamin/Cummings Publishing Company, Inc, 1993
- [2] BYTE.COM. *A Brief History of Programming Languages*, disponível em <http://www.byte.com/art/9509/sec7/art19.htm>
- [3] Haskell.org, Libraries and Tools For Haskell: *Integrated Development Environments*, disponível em <http://www.haskell.org/libraries/#ide>
- [4] Microsoft Research, *Microsoft Research Cambridge*, disponível em <http://research.microsoft.com/cambridge/>
- [5] The Lambda Complex, *Why does Haskell matter?*, disponível em [http://www.haskell.org/complex/why\\_does\\_haskell\\_matter.html](http://www.haskell.org/complex/why_does_haskell_matter.html)
- [6] Haskell.org, *A Gentle Introduction to Haskell, Version 98*, disponível em <http://www.haskell.org/tutorial/>
- [7] Microsoft, *Microsoft Visual Studio Developer Center*, disponível em <http://msdn.microsoft.com/vstudio>
- [8] Cunha C., *Pro.NET- Processos de Desenvolvimento de Software para a Plataforma .NET*, Encontro da Qualidade e Produtividade em Software (EQPS), Fortaleza, setembro/2003.