

6 Conclusão

Trabalhando colaborativamente, pelo menos potencialmente, pode-se produzir melhores resultados do que se os membros do grupo atuassem individualmente (Fuks et al., 2002). Groupware é um tipo de software que oferece suporte à interação entre indivíduos possibilitando o trabalho colaborativo. Learningware é um tipo de groupware usado no aprendizado colaborativo.

Um groupware envolve aspectos multidisciplinares em sua construção e é difícil de aplicar e testar, sendo especialmente vulnerável a falhas (Gerosa et al., 2004). Como cada grupo tem suas próprias necessidades que se modificam ao longo do tempo, é importante que o groupware seja desenvolvido de forma iterativa e que a experiência do seu uso seja usada para guiar seu desenvolvimento, criando novos serviços ou melhorando as funcionalidades dos serviços existentes. Através de uma arquitetura componentizada, o groupware pode ser adaptado plugando e desplugando componentes, compondo uma aplicação que melhor satisfaça um grupo de trabalho.

O AulaNet é um learningware desenvolvido pelo Laboratório de Engenharia de Software (LES) da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) desde 1997 e desde o lançamento da versão 2.0 a sua arquitetura nunca sofreu reengenharia. Alunos de graduação, mestrado e doutorado desenvolvem suas monografias, dissertações e teses usando o AulaNet para realizar suas pesquisas. O AulaNet é usado no curso Engenharia de Groupware, onde atua como repositório de informações, e no curso Tecnologias de Informação Aplicada A Educação (TIAE), onde é usado integralmente a distância. O AulaNet é distribuído pela EduWeb, que também realiza customizações para atender as demandas de seus clientes. Atualmente o AulaNet é usado em diversas universidades e empresas no mundo inteiro. Com o passar do tempo o AulaNet acumulou código legado que não segue as principais boas práticas e tecnologias conhecidas atualmente. Como consequência, as equipes do LES e da EduWeb vêm enfrentando dificuldades para desenvolver novos serviços e manter os

existentes, sendo apontadas causas como a baixa modularidade e o uso de um paradigma funcional (Pimentel et al., 2005). Para solucionar estas questões, iniciou-se o desenvolvimento de uma nova versão, o AulaNet 3.0.

Gerosa (2006) em sua pesquisa de doutorado na PUC-Rio concebeu uma arquitetura baseada em componentes para o AulaNet 3.0. Esta arquitetura possui dois níveis de componentização: o de serviços e o dos componentes 3C. Os serviços, como a Conferência e o Debate são plugados e desplugados em um framework de componentes, o Groupware Component Framework, compondo um groupware específico que atenda às necessidades de um grupo de trabalho. Os serviços por sua vez são compostos por componentes 3C, que são reusados na construção de diversos serviços.

Ao desenvolver componentes, o desenvolvedor se depara com inúmeros desafios. Além de entender do seu domínio de conhecimento ele deve lidar com questões como persistências de dados, controle de transações, segurança e muitas outras. O desenvolvimento de groupware já é difícil por si só devido a seu caráter multidisciplinar e a heterogeneidade dos diversos grupos de trabalho e não deveria ser complicado por questões de baixo nível.

Frameworks de infra-estrutura de sistemas simplificam o desenvolvimento de sistemas de infra-estrutura portáteis e eficientes (Fayad et al., 1999b). Estes frameworks possibilitam uma visão mais alto nível destas questões. Nesta dissertação, foi visto como estes frameworks oferecem serviços que simplificam o desenvolvimento de groupwares. Tomando uma abordagem em camadas, foram vistos frameworks usados na camada de negócios e frameworks usados na camada de apresentação.

Ao buscar por frameworks para lidar com determinada questão de infra-estrutura muitas vezes o arquiteto de software se depara com diversos frameworks similares que se propõe a realizar a mesma coisa. A leitura da documentação do fabricante pode ajudar na escolha, mas muitas vezes as informações fornecidas pelo fabricante são subjetivas. Realizando uma análise técnica, possivelmente implementando uma prova de conceito, o arquiteto obtém indícios sobre qual framework atende melhor a suas necessidades. Entretanto nem sempre a implementação da prova de conceito é possível devido à limitação natural de tempo e recursos que todo projeto tem.

Além de identificar se o framework atende às necessidades técnicas do projeto, há outros fatores importantes que devem ser considerados. É importante saber se há documentação disponível, se há suporte disponível, se há ferramentas compatíveis com o framework, se o framework está sendo usado por outras instituições e se há profissionais aptos a trabalhar com o framework. Estas questões não-técnicas usualmente são tão importantes quanto as questões técnicas, principalmente no caso do AulaNet devido à rotatividade de desenvolvedores no LES e a integração com a equipe da EduWeb. Esta dissertação apresentou uma forma de comparar estes fatores.

Na camada de negócios, as principais questões de infra-estrutura resolvidas por frameworks são: persistência de dados, gerenciamento de transações, gerenciamento de segurança e exposição de serviços remotamente. Estas questões são tratadas no framework de componentes Enterprise Java Beans (EJB) e, por isso, ele foi avaliado como opção.

Após a análise técnica do EJB chegou-se a conclusão que ele apresentava inúmeras desvantagens. É preciso manter uma grande quantidade de arquivos para realizar manutenção em um componente EJB. Componentes EJB dependem de servidores de aplicação que possuam containeres EJB que geralmente são mais caros e complexos. Além disso, o modelo EJB resulta em componentes altamente intrusivos e conseqüentemente mais difíceis de serem testados unitariamente. As restrições de programação impostas pela especificação EJB 2.1 impossibilitam a realização de atividades corriqueiras. A especificação EJB 3.0 vem se inspirado em um modelo de POJOs e frameworks como o Hibernate para solucionar estas questões mas, como seu futuro é incerto, optou-se por seguir uma arquitetura de POJOs com frameworks.

Para tratar a persistência de dados e dos desafios decorrentes do conflito dos paradigmas Objeto/Relacional, é usado o framework *Object Relational Mapping* (ORM) Hibernate (2005). Antes de adotar o Hibernate foi realizada uma análise técnica onde foram analisadas as características deste framework. Após a implementação de uma prova de conceito, um protótipo do serviço Conferências do AulaNet, percebeu-se que este framework supria as necessidades de persistência do AulaNet.

Após a análise técnica foi realizada a análise não-técnica, que iniciou-se pela seleção dos concorrentes. Os frameworks ORM iBATIS (2005), Apache OJB

(OJB, 2005) e JDOMax (2005), uma implementação da especificação JDO (2005) são soluções de mapeamento completo (Fussel,1997) e gratuitos. A análise determinou o Hibernate como o framework com mais documentação disponível, com mais suporte disponível (é a comunidade mais ativa e possui suporte pago), com mais ferramentas compatíveis, com mais empresas usando e com mais profissionais aptos a trabalhar com ele disponíveis no mercado. Como os outros não apresentavam nenhuma vantagem técnica significativa, a escolha do Hibernate se manteve.

Para solucionar às questões de gerenciamento de transações, gerenciamento de segurança e exposição de serviços remotamente, foi escolhido o framework Spring (2005). A análise técnica mostrou que o Spring fornecia meios para lidar com estas questões e, além disso, fornecia um módulo de *Dependency Injection* que torna os componentes mais fracamente acoplados e mais fáceis de serem testados unitariamente. Spring possui também um módulo de Programação Orientada a Aspectos, possibilitando que pesquisas nesta área sejam realizadas com o AulNet. Spring também integra-se ao Hibernate, facilitando o seu uso.

O Spring também se manteve após a na análise não-técnica. Foram encontrados 4 frameworks gratuitos com características similares ao Spring: Avalon (2005), que se transformou no projeto Excalibur (2005), o HiveMind (2005), NanoContainer (2005) e o PicoContainer (2005). A análise mostrou que o Spring é superior em todas as categorias e, novamente, a escolha se manteve.

Na camada de apresentação, há uma família de frameworks denominada web frameworks que oferecem uma implementação do padrão de projetos Model View Controller (MVC) (Fowler, 2002), além de tratar também desafios decorrentes do uso do HTML na construção de interfaces com o usuário. Estima-se que há 54 web frameworks disponíveis gratuitamente na plataforma Java (Manageability, 2005). Devido ao grande número de web frameworks disponíveis, foi preciso adotar uma outra abordagem. Foram selecionados 3 para serem analisados tecnicamente. O Struts (2005), por ser o mais popular, o Spring MVC, módulo do framework Spring (2005), por ter sido escolhido para a camada de apresentação, e o JavaServer Faces (JSF) (2005), por especificar um modelo de componentes e ser um padrão da Sun.

A análise técnica indicou cada um com suas vantagens. O Spring MVC, o menos intrusivo de todos, o Struts, com o melhor módulo de validação de entrada

de dados e o JSF sendo o único dos três a especificar um modelo de componentes. Após a análise técnica, escolheu-se o JSF, pois ele aparenta ser um modelo de componentes promissor e porque começa a se desenhar um mercado de componentes JSF com o surgimento de projetos como MyFaces (2005) e WebGalileo Faces (WebGalileo, 2005), que são gratuitos, e o WebCharts (2005), que é comercial. Alguns destes componentes, como o editor HTML encontrado no projeto MyFaces, atendem a demandas atuais de usuários do AulaNet.

Ao realizar a análise não-técnica notou-se que Struts liderou todas as categorias exceto a categoria de ferramentas compatíveis. JSF, que liderou esta categoria, ficou em segundo lugar em todas as outras exceto a categoria de disponibilidade de suporte onde foi terceiro. JSF foi também o framework cujo número de oportunidades requerendo habilidade no framework mais cresceu ao longo das pesquisas. O bom desempenho do JSF na análise técnica foi crucial para que ele se mantivesse como escolha para a camada de apresentação. Mesmo tendo a sua primeira versão estável lançada em 2004, JSF ficou atrás apenas do Struts, que parece ser o web framework mais aceito atualmente, mas que não define um modelo de componentes e por isso não será usado.

Apesar da arquitetura do AulaNet 3.0 prever apenas estes 3 frameworks, é possível incorporar outros frameworks, oferecendo infra-estrutura para tratar questões não previstas. Nesta dissertação foi visto como incorporar o framework de agentes Jade a arquitetura do AulaNet 3.0. Para integrar um novo framework na camada de negócios da arquitetura do AulaNet 3.0 é preciso integrá-lo ao Spring. Como o Spring não é intrusivo e é baseado em boas práticas, como a programação para interfaces (Bloch, 2005) e encapsulamento através de propriedades JavaBeans (2005), normalmente isto não é um problema. Como visto nesta dissertação, o Jade pode ser integrado à arquitetura através de um adaptador construído com poucas classes, possibilitando o uso de agentes de software no AulaNet tanto para fins de pesquisa quanto para fins de mercado.

Além disso, através dos *Proxys* remotos do Spring é oferecido suporte a aplicações móveis. Com a nova arquitetura, é possível integrar o AulaNet e o AulaNetM, extensão do AulaNet para dispositivos móveis, no nível de serviços, aumentando o reuso entre os dois projetos.

Desenvolver groupware é complexo e não deveria se tornar mais complexo devido a aspectos de infra-estrutura de baixo nível como persistência de dados,

gerenciamento de transações, etc. Frameworks de infra-estrutura simplificam estes aspectos ao tratarem os problemas em baixo nível e fornecerem interfaces em alto nível. Estes frameworks são projetados e desenvolvidos por especialistas nesta área enquanto que os componentes de groupware são projetados por especialistas em groupware.

Esta pesquisa mostrou como uma arquitetura em três camadas pode se beneficiar pelo uso de frameworks de infra-estrutura. Mostrou-se também uma forma de avaliar frameworks que leva em conta tanto aspectos técnicos quanto não-técnicos. Espera-se que esta pesquisa possa auxiliar arquitetos de groupware e de software a construir aplicações, auxiliando na tomada de decisões sobre a estruturação da arquitetura técnica e na escolha de frameworks de aplicação que tratem de aspectos de infra-estrutura de baixo nível. Espera-se também que este trabalho guie os desenvolvedores LES/EduWeb na construção do AulaNet 3.0, para que este possa continuar rendendo mais uma década de pesquisas e aprendizagem.