



UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

SISTEMAS DE INFORMAÇÃO

DECISÕES ARQUITETURAIS

David Barros

Rafael Takano

Vanessa Brenda

Tendo como finalidade uma visualização mais clara e assertiva dos relacionamentos, optamos pelo uso de tabelas.

Seguem:

Área	Estrutura da aplicação
Decisão de Arquitetura	Xamarin, que é uma camada de abstração que gerencia a comunicação de código compartilhado com o código de plataforma subjacente.
Justificativa e Relacionamento com o Sistema	O Xamarin é executado em um ambiente gerenciado que fornece conveniências, como alocação de memória e coleta de lixo, além de nos permitir gravar toda a lógica de negócios em uma única linguagem. Software livre e de alto desempenho e como focamos multiplataformas, por ser um sistema bancário, encaixa-se perfeitamente.

Área	API
Decisão de Arquitetura	Java e Spring MVC
Justificativa e Relacionamento com o Sistema	Java, pela experiência da equipe com a linguagem e a integração com o framework subjacente. Spring, por ser leve e não intrusivo além de ter inúmeros serviços à disposição para facilitar o desenvolvimento do sistema (Hubank) como: injeção de dependências, gerenciamento de transações etc.

Área	Persistência
Decisão de Arquitetura	Java Database Connectivity, MySQL
Justificativa e Relacionamento com o Sistema	JDBC é um conjunto de classes e interfaces (API's) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional; Api de baixo nível e base para api's de alto nível; Amplia o que você pode fazer com Java. Possibilita o uso de banco de dados já instalados. MySQL é um SGBD extremamente popular, justificando assim o uso deste, sendo o JDBC uma escolha eficiente e compatível com a estrutura do sistema.

Área	Componente de Segurança e Fachada do mainframe
Decisão de Arquitetura	Spring Bean
Justificativa e Relacionamento com o Sistema	Como estamos utilizando Spring como API, a responsabilidade pela segurança dos dados fica a cargo dos próprios bean's do Spring, que nada mais que é um objeto montado, instanciado e estruturado pelo Spring.

Área	Controle do Resumo de Contas
------	------------------------------

Decisão de Arquitetura	Spring MVC Rest Controller
Justificativa e Relacionamento com o Sistema	O Spring MVC Rest Controller nada mais é que a combinação das annotations @Controller e.ResponseBody, onde o controller o qual referenciamos é a classe que controla as requisições. Sendo utilizado por estar integrado à API do Spring MVC.