Chamada UFG
Documento de Arquitetura de Software

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
26/10/2018	1.0	Criação do Documento	Isaias, Wesley, Matheus

Índice Analítico

Documento de Arquitetura de Software

1. Introdução

1.1 Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

1.2 Escopo

Este documento de Arquitetura de software fornece uma visão geral de arquitetura do projeto Chamada UFG, que está sendo desenvolvido pelo Grupo 2 da disciplina de Projeto de Software da UFG..

1.3 Definições, Acrônimos e Abreviações

UFG - Universidade Federal de Goiás.
QoS – Quality of Service, ou qualidade de serviço. Termo utilizado para descrever um conjunto de qualidades que descrevem as requisitos não-funcionais de um sistema, como performance, disponibilidade e escalabilidade[QOS].
Servidor - é um software ou computador, com sistema de computação centralizada que fornece serviços a uma rede de computadores, chamada de cliente.
Software - é uma sequência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento. "*Software*" também é o nome dado ao comportamento exibido por essa sequência de instruções quando executada em um computador ou máquina semelhante além de um produto desenvolvido pela engenharia de software.
Computador - é uma máquina capaz de variados tipos de tratamento automático de informações ou processamento de dados.
Rede de computadores - é quando há pelo menos dois ou mais computadores, e outros dispositivos interligados entre si de modo a poderem compartilhar recursos físicos e lógicos.
UFGNET - portal da Universidade Federal de Goiás que unifica o acesso a vários sistemas da Universidade.

1.4 Referências

[UFG] https://www.ufg.br/
[QOS] https://docs.oracle.com/cd/E19636-01/819-2326/6n4kfe7dj/index.html
[Software] https://pt.wikipedia.org/wiki/Software
[Computador] https://pt.wikipedia.org/wiki/Computador
[Rede de Computadores] https://pt.wikipedia.org/wiki/Rede_de_computadores
[Automação de Chamada em sala de aula] https://fastformat.co/contests/submissions/5/pdf
[UFGNET] http://www.cercomp.ufg.br/n/50769-novo-portal-ufgnet

1.5 Visão Geral

Este documento está organizado em tópicos relacionados às diferentes visões arquiteturais.

2. Representação Arquitetural

Este documento apresenta a arquitetura como uma série de visões: visão de caso de uso, visão lógica, visão de processo e visão de implantação. Não existe uma visão de execução específica descritos neste documento. Estes são pontos de vista sobre uma base modelo Unified Modeling Language (UML), desenvolvido usando o Astah.

3. Metas e Restrições da Arquitetura

As credenciais usadas pelo usuário para acesso ao aplicativo, são as mesmas usadas para acessar o Portal UFGNET, quando usuário solicitar acesso ao aplicativo, o sistema deve consultar e validar suas credenciais consultando a base de dados do portal UFGNET. O Aplicativo deve ser desenvolvido para ser compatível dispositivos que tenham sistema operacional Android e Ios

4. Visão de Casos de Uso

Os Casos de Uso do Sistema são:

- CSU01 - Fazer Login
- CSU02 - Listar Turmas
- CSU03 - Registrar Presença
- CSU04 - Liberar Chamada

4.1 Realizações de Casos de Uso

Caso de uso	*Descrição*
CSU1	*Neste caso de uso, o usuário inicializa o sistema*
CSU2	*Neste caso de uso o aplicativo lista todas as matérias em que o professor ou aluno está matriculado.*

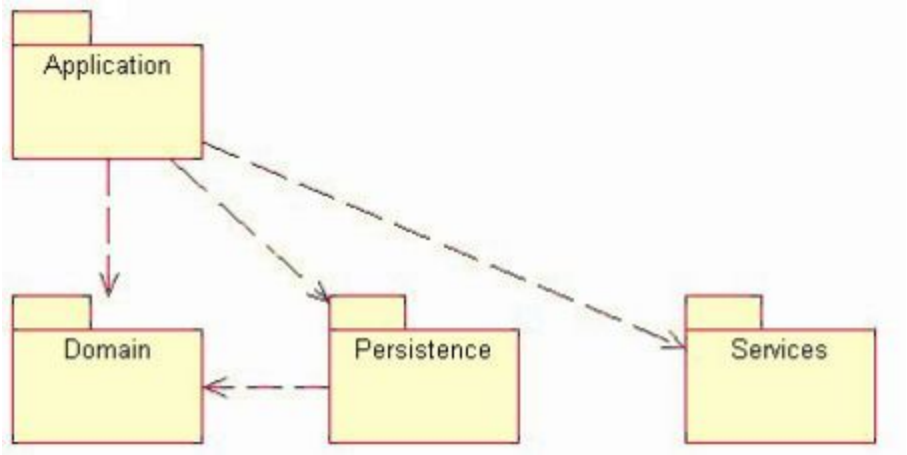
CSU3	*Neste caso de uso o aluno registra sua presença .*
CSU4	*Neste caso de uso o professor libera a chamada para que os alunos registrem sua presença.*

5. Visão Lógica

5.1 Visão Geral

Uma descrição da visualização lógica da arquitetura. Descreve as classes mais importantes, suas organizações nos pacotes de serviço e subsistemas, e a organização desses subsistemas em camadas. Também descreve as realizações de casos de uso mais importantes como, por exemplo, os aspectos dinâmicos da arquitetura. Os diagramas de classe podem ser incluídos para ilustrar os relacionamentos entre as classes, subsistemas, pacotes e camadas arquitetonicamente significantes. A visualização lógica do Sistema de Chamada UFG é composta de 4 pacotes principais:

- **Aplicação** - contém classes para apresentar as funcionalidades na tela do aplicativo.
- **Domínio** - contém pacotes com classes que representam o Usuário, Aluno, Professor, Turma, Frequência e Curso.
- **Persistência -** contém classes para persistirem os dados do sistema no Banco de Dados. Nesse ponto no design apenas os registros da chamada são persistidos.
- **Serviços -** contém classes que contém a regra de negócio do sistema.



5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura

- **Aplicação** -

  Classe Login: representa a tela de autenticação do usuário.

  Classe Listar Turma: representa a tela que exibe a listagem de turmas em que o usuário está cadastrado.

  Classe Registrar Presença: representa tela em que o aluno registra sua presença, previamente liberada pelo respectivo professor.

  Classe Liberar Chamada: tela em que o professor permite que o aluno possa registrar sua presença.

- **Domínio** -

  Classe Usuário: representa o usuário do aplicativo, que pode ser o professor ou aluno.

  Classe Aluno: usuário que registra presença.

  Classe Professor: usuário que libera a chamada, para que os alunos possam registrar suas presenças.

Classe Curso: classe vinculada curso em que o aluno está matriculado.

Classe Turma: disciplina que está sendo cursada pelo aluno e ministrada pelo professor.

Classe Frequência: classe que representa a presença registrada pelo aluno.

- **Persistência -**

  Classe Persistência: representa a ação do aplicativo em enviar o registro da presença para o SIGAA realizar a persistência das informações no banco de dados.

- **Serviços -**
  Classe Permitir Liberação: verifica se o professor está liberando a turma no horário e data permitida.

  Classe Permitir Registro de presença: verifica se o professor da turma liberou a chamada.

6. Visão de Processos

O sistema é gerenciado por meio de processos, esses processos podem ser divididos com base na sua capacidade de influência para o sistema como um todo, podendo ser classificados em dois tipos :

Processos leves: São processos de baixa importância dentro do sistema, tais como threads de baixa prioridade criadas para processamento paralelo.
Processos pesados: São processos de alto impacto dentro do sistema como um todo, em que o mau gerenciamento pode comprometer outras áreas do sistema ,tais como threads criadas para a interação com o usuário.

7. Visão de Implantação

O sistema é construído na linguagem java ,com o foco voltado para dispositivos android. A linguagem java utiliza de máquinas virtuais para executar o sistema nas mais diversas

plataformas (mobile, tablets, entre outros). As máquinas virtuais interpretam o código e executam as instruções nele contidas diretamente nos sistemas

8. Visão da Implementação

8.1 Visão Geral

Seguindo o padrão arquitetural MVC ,o sistema pode ser dividido em 3 camadas : Model, view, controller. Nessas camadas são separados as classes do código com base em sua influência no sistema. As classes da camada view são responsáveis apenas pelo front-end do sistema. As classes da camada model são as classes atômicas e as classes compostas essenciais para construção do sistema. As classes da camada controller são responsáveis pela construção do back-end do sistema.

8.2 Camadas

8.2.1 View

A camada view é responsável pela interação do usuário com o sistema ,nessa camada são realizados etapas como construção de interface de usuário e interação para as configuração das funcionalidades.

8.2.2 Model

A camada model é responsável pelo armazenamento das classes mais atômicas do projeto ,nessa camada são realizados ações como a execução básica dos sons e a execução complexa dos sons.

8.2.3 Controller

A camada controller é responsável pela execução de algoritmos complexos como interpretação dos dados da interface para o sistema e o controle do ciclo de vida da aplicação.

9. Tamanho e Desempenho

A arquitetura do software escolhido suporta o dimensionamento e dos requisitos de tempo.

1. O sistema deverá suportar até 5000 usuários simultâneos no aplicativo.
2. O sistema deve ser capaz de completar 90% de todas as operações dentro de dois

minutos.
3. O aplicativo deve exigir menos de 20MB de espaço em disco e 32MB de RAM.

10. Qualidade

A arquitetura do software suporta os requisitos de qualidade.

1. O aplicativo deve estar disponível para IOS e Android.
2. A interface do usuário do aplicativo deve seguir o padrão do Sigaa Mobile para facilitar a utilização do sistema pelos usuários que já estão acostumados com o Sigaa.
3. O sistema deverá estar disponível no mínimo das 08:00 às 22:00 de segunda a sábado.