All for One

Software Architecture Document

Version 0.4

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 15/10/2020 | 0.1 | Tela com lista de produtos e botão para adicionar loja | Luiz Guerra |
| 21/10/2020 | 0.2 | Tela com lojas cadastradas que leva a tela de compras do usuário com a loja e backend | Vitor |
| 21/10/2020 | 0.3 | Tela para pedir relatório dado um filtro específico | Equipe |
| 22/10/2020 | 0.4 | Correção de bugs | Equipe |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 4

4. Use-Case View 4

5. Logical View 4

5.1 Overview 4

5.2 Architecturally Significant Design Packages 4

5.3 Use-Case Realizations 4

6. Process View 4

7. Deployment View 4

8. Implementation View 4

8.1 Overview 4

8.2 Layers 4

9. Data View (optional) 4

10. Size and Performance 4

11. Quality 4

Software Architecture Document

# Introduction

Este documento apresenta a arquitetura proposta para o aplicativo All for One. A arquitetura é apresentada através de um conjunto de visões que juntas visam cobrir os principais aspectos técnicos relativos ao desenvolvimento e implantação do sistema em questão. O objetivo é capturar e formalizar as principais decisões tomadas com relação à arquitetura do sistema.

**Purpose**

Este Documento de Arquitetura de Software se aplica ao projeto All for One, desenvolvido pelos alunos do curso de Engenharia de Software da PUCRS na disciplina Projeto e Arquitetura de Software.

## Scope

O escopo desse documento é documentar as partes significativas do ponto de vista da arquitetura, como sua divisão em camadas e pacotes.

## Definitions, Acronyms, and Abbreviations

US = User Story

CA = Critérios de aceitação

MVVM = Model-View-ViewModel

JSON = JavaScript Object Notation

## References

[This subsection provides a complete list of all documents referenced elsewhere in the **Software Architecture Document**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

[This subsection describes what the rest of the **Software Architecture Document** contains and explains how the **Software Architecture Document** is organized.]

# Architectural Representation

[This section describes what software architecture is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]

# Architectural Goals and Constraints

Os critérios utilizados para a seleção da solução arquitetural foram:

|  |  |
| --- | --- |
| **Requisito** | **Solução** |
| Linguagem | Swift |
| Plataforma | Localhost |
| Persistência | Jason Data Reader |

# Use-Case View

Esta seção lista as especificações centrais e significantes para a arquitetura do sistema.

Lista de casos de uso do sistema:

|  |  |  |
| --- | --- | --- |
| **#UC** | **Nome UC** | **Descrição UC** |
| UC01 | Cadastrar um ecommerce | Salvar um ecommerce no app utilizando nome da loja, login e senha do usuário |
| UC02 | Consultar pedidos do cliente | Usuário pode visualizar todos os pedidos cadastrados no app |
| UC03 | Cadastrar pedidos do cliente | Usuário pode cadastrar uma compra no app |
| UC04 | Consultar pedidos por ecommerce | Usuário pode consultar as compras por loja no app |
| UC 05 | Remover um ecommerce de produtos | Usuário pode remover uma loja e todos os pedidos desta loja |
| UC 06 | Gerar relatório de compras do cliente | Usuário pode solicitar um relatório de todas os seus pedidos |
| UC07 | Gerar relatório por e-commerce | Usuário pode solicitar um relatório de todos as compras de um ecommerce específico |
| UC08 | Gerar relatório por data | Usuário pode solicitar um relatório de todas as compras a partir de uma data |
| UC09 | Gerar relatório por agilidade de entrega | Usuário pode solicitar um relatório para saber se entrega foi ágil |
| UC10 | Gerar relatório por cumprimento de prazo de entrega | Usuário pode solicitar um relatório para verificar se o prazo de entrega foi cumprido |

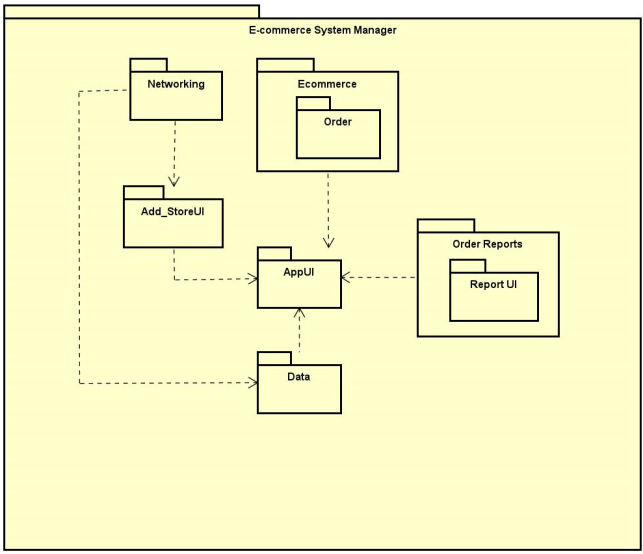
# Logical View

[This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities. You should introduce architecturally significant classes and describe their responsibilities, as well as a few very important relationships, operations, and attributes.]

## Overview

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

## Architecturally Significant Design Packages



## Use-Case Realizations

Fluxos principais

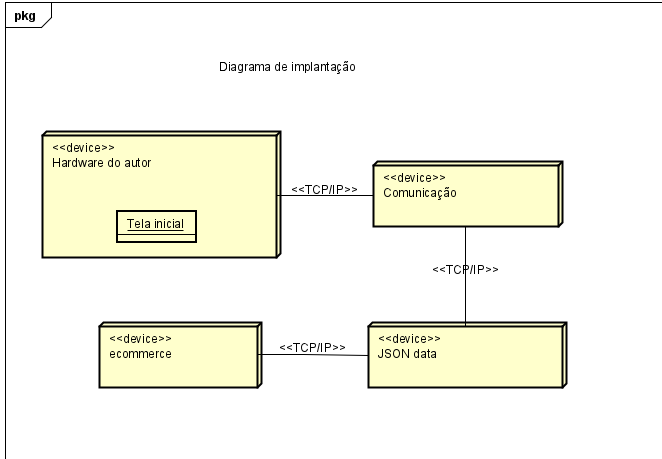
1. Cadastrar ecommerce
   1. Usuário abre tela para adicionar ecommerce
   2. Usuário digita ecommerce
   3. ecommerce existe
   4. Login e senha do usuário estão corretos
   5. Aplicativo se conecta com ecommerce
   6. Aplicativo volta pra tela de lista de ecommerce cadastrados
2. Ver pedidos por ecommerce
   1. Usuário clica em um ecommerce
   2. Aplicativo carrega a tela com pedidos feitos pelo usuário de um ecommerce
3. Gerar report de todas as compras feitas
   1. Clica no botão para gerar relatórios
   2. Clica no botão para gerar relatório de todos os pedidos
   3. Sistema gera uma tela com o relatório

# Process View

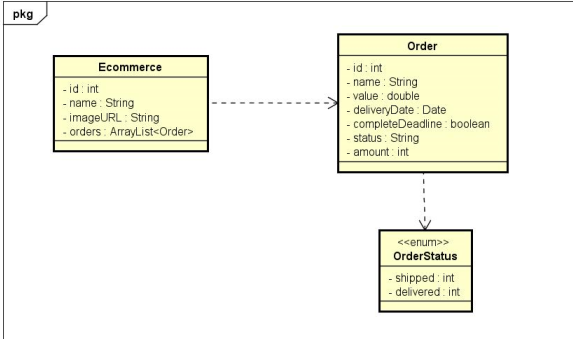
[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

# Deployment View

O software foi implantado e executado em localhost



# Implementation View



## Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]

## Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

1, HomeScreen

1.1 HomeViewModel

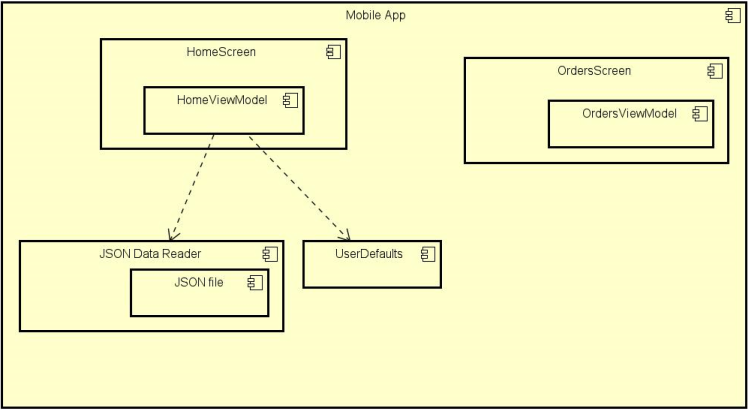
2. JSON DataReader

2.1 JSON file

3. User Defaults

4. OrderScreen

4.1. OrderViewModel



# Size and Performance

A aplicação por ter um escopo específico, é pequena. A arquitetura utilizada é MVVM, dado que existem várias telas que podem ser reutilizadas, dado ao número de tabelas utilizando um mesmo padrão de células.

Apesar de não ser testado o tempo exato de carregamento de telas e informações do JSON, o aplicativo é bem rápido e fluido. Atribuímos essa velocidade principalmente pelo fato da aplicação ser pequena, mas um fator a considerar também é por termos utilizado as boas práticas da linguagem e arquitetura para evitar um desgaste na aplicação.

# Quality

A arquitetura escolhida foi MVVM, pois com ela é possível adquirir um bom nível de abstração nas telas e lógicas da ViewModel, garantindo que se for necessário implementar algo novo que mantenha algo que já existe, é só reutilizar o que for necessário. Da mesma forma, se for necessário adicionar novas camadas entre o que já existe, o trabalho adicional será pequeno.

Introdução

Requisitos e restrições arquiteturais

Visão de Caso de uso (user stories)

Visão lógica (pacotes/camadas/serviços)

Visão de Implantação (Diagrama de deploy)