
Relatório de Especificação: Desenho

Projeto:	TaskMate
Preparado por:	Grupo 4 Diogo Baptista (nº 79405) Francisco Morgado (nº 85009) Rúben Menino (nº 89185) João Génio (nº 88771) João Rodrigues (nº 88856)
Data de preparação:	Aveiro, 20 de Novembro de 2019
Circulação:	Docentes e Discentes de AMS.

Controlo de versões

Quando?	Responsável	Alterações significativas
20/11/2019	João Génio, Ruben Menino	Introdução, Tecnologias/ Implementação e protótipos dos produtos
22/11/2019	Diogo Baptista	Objetivos e restrições da arquitetura
22/11/2019	João Rodrigues	Vistas de Arquitectura

Índice

1	4
2	7
2.1	7
2.2	7
2.3	8
3	10
3.1	10
3.2	11
4	13
4.1	13
4.2	13
4.3	13
5	15
6	16
7	Erro! Marcador não definido.

1 Introdução

Neste trabalho de projeto, propomos e especificamos um sistema de informação. Apresentamos também a oportunidade que sustenta o desenvolvimento do projeto e, de forma geral, o âmbito das funcionalidades esperadas.

Neste documento realça-se objetivos da arquitetura face aos requisitos do domínio, os atributos de qualidade, vários modelos de comportamento e algumas das especificações acerca das tecnologias usadas e variantes na implementação da aplicação do projeto.

O sucesso deste projeto vai ser medido de acordo com o seu impacto junto do cliente.

Os resultados considerados mais determinantes prendem-se com o esquema dos vários diagramas de sequência que permitiram a aproximação dos requisitos que foram especificados à implementação da solução-

Índice de diagramas

[actualizar o índice quando o documento completo. Todos os diagramas devem ser legendados.]

Diagrama 3: Modelo do domínio.

Erro! Marcador não definido.

Índice de tabelas

[atualizar o índice quando o documento completo. Todas as tabelas devem ser legendadas.]

Tabela 1: Principais benefícios do CourseRegistrarSystem. **Erro! Marcador não definido.**

2 Objetivos e restrições da arquitetura

2.1 Objetivos da arquitetura face aos requisitos do domínio

[Identificar, de uma forma geral, as características críticas do problema/solução que vão determinar as grandes decisões da arquitetura.

Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues?

Are there hardware dependencies that should be isolated from the rest of the system?

Does the system need to function efficiently under unusual conditions?

Are there integrations with external systems?

Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,...)?

Are there special requirements concerning the extent or the type of the data managed (e.g.: large collection of videos)?

Etc.]

E.g.:

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- ☐ The existing legacy Course Catalog System at Wylie College will be the main source for all course information; it must be accesses to retrieve the course details for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [see Report X].
- ☐ The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [see Report X].
- ☐ The C-Registration System will be implemented as a scalable system, deployed in the cloud. The users should access using only their browser, with no required configuration, to the web portal.

Existem vários requerimentos chave no sistema que necessitam de uma arquitetura específica, como:

1. Cada utilizador requer criação de conta e toda essa informação tem de ser armazenada numa base de dados, tal como toda a gestão de troca/prestação de serviços.
2. O sistema será uma aplicação mobile pois queremos uma fácil e intuitiva utilização por parte do utilizador e queremos um sistema que esteja disponível a toda a hora e em qualquer lugar para os nossos utilizadores.
3. A localização de cada utilizador e a localização de cada tarefa a ser realizada necessita de um sistema de GPS.
4. O pagamento de tarefas prestadas necessita de um método de pagamento online, móvel e rápido.

2.2 Atributos de qualidade com impacto na arquitetura

[Enumerar atributos de qualidade (naturalmente já formulados como requisitos não funcionais em relatórios anteriores) com impacto nas escolhas relativas à arquitetura. Só incluir os que contribuem para influenciar a arquitetura do sistema.

Esta secção é diferente da anterior. Na anterior, trata-se de restrições gerais; nesta secção, a ideia é fazer um “apanhado” de requisitos, formulados como tal (S.M.A.R.T.), já identificados antes]

E.g.:

1. The desktop user-interface shall be Windows 95/98 compliant.
2. The user interface of the C-Registration System shall be designed for ease-of-use and shall be appropriate for a computer-literate user community with no additional training on the System.
3. Each feature of the C-Registration System shall have built-in online help for the user. Online Help shall include step by step instructions on using the System. Online Help shall include definitions for terms and acronyms.
4. The C-Registration System shall be available 24 hours a day, 7 days a week. There shall be no more than 4% down time.
5. Mean Time Between Failures shall exceed 300 hours.
6. Upgrades to the PC client portion of C-Registration shall be downloadable from the UNIX Server over the internet. This feature enables students to have easy access to system upgrades.

A aplicação TaskMate necessita dos seguintes requisitos:

1. As aplicações móveis existentes devem ser compatíveis com Android 5.0 (lollipop).
2. O apoio ao cliente deve ser acessível em todas as páginas na aplicação.
3. O manual de instruções deverá estar disponível para download ou visualização online.
4. O apoio ao cliente deverá estar disponível 24 horas por dia, 7 dias por semana. Não deverá haver mais de 5% de indisponibilidade.
5. Deverá haver manutenções de sistema semanais.
6. Na realização de manutenções de sistema todas as transações e pedidos de tarefas serão fechados para evitar qualquer tipo de falha.
7. As manutenções de sistema não deverão ser maiores que 1 hora.

2.3 Decisões tomadas e justificação

[List the decisions that have been made regarding architectural approaches and the constraints being placed on the way that the developers build the system.

These will serve as guidelines for defining architecturally significant parts of the system. Justify each decision or constraint so that developers understand the importance of building the system according to the context created by those decisions and constraints. This may include a list of DOs and DON'Ts to guide the developers in building the system.]

E.g.:

- **compatibilidade com OneM2M:** a camada de IoT será baseada na utilização de *middleware* já existente, para implementar a compatibilidade com o standard OneM2M. A implementação *opensource* do Eclipse OneM2M será usada como referência. A versão do *middleware* não poderá ser revista a meio de uma iteração.

Após os requerimentos chave enunciados anteriormente decidimos:

- Usar a aplicação MBWay: Com MBWay todas as transações e pagamentos de tarefas serão realizados de forma rápida, móvel e simples.
- Usar Google Maps: Para disponibilizar a localização de cada utilizador e de cada tarefa a ser realizada.
- Utilizar como framework da aplicação android a Firebase.
- Utilizar a base de dados existente na Firebase para armazenar todo o tipo de dados usados pela aplicação e informação de utilizadores.

3 Vistas de arquitetura

3.1 Arquitetura lógica da solução

[Como é que a solução de software vai estar organizada?

Explicar a organização prevista do software em módulos, camadas ou subsistemas. Para além dos diagrama, é necessário texto de explicação.

Sugestão: apresentar uma arquitetura lógica por camadas (layered architecture), visualizada com um diagrama de pacotes]

Na descrição da arquitetura, explicar as Tecnologias e ambientes e suporte que se prevê utilizar: web, mobile,...

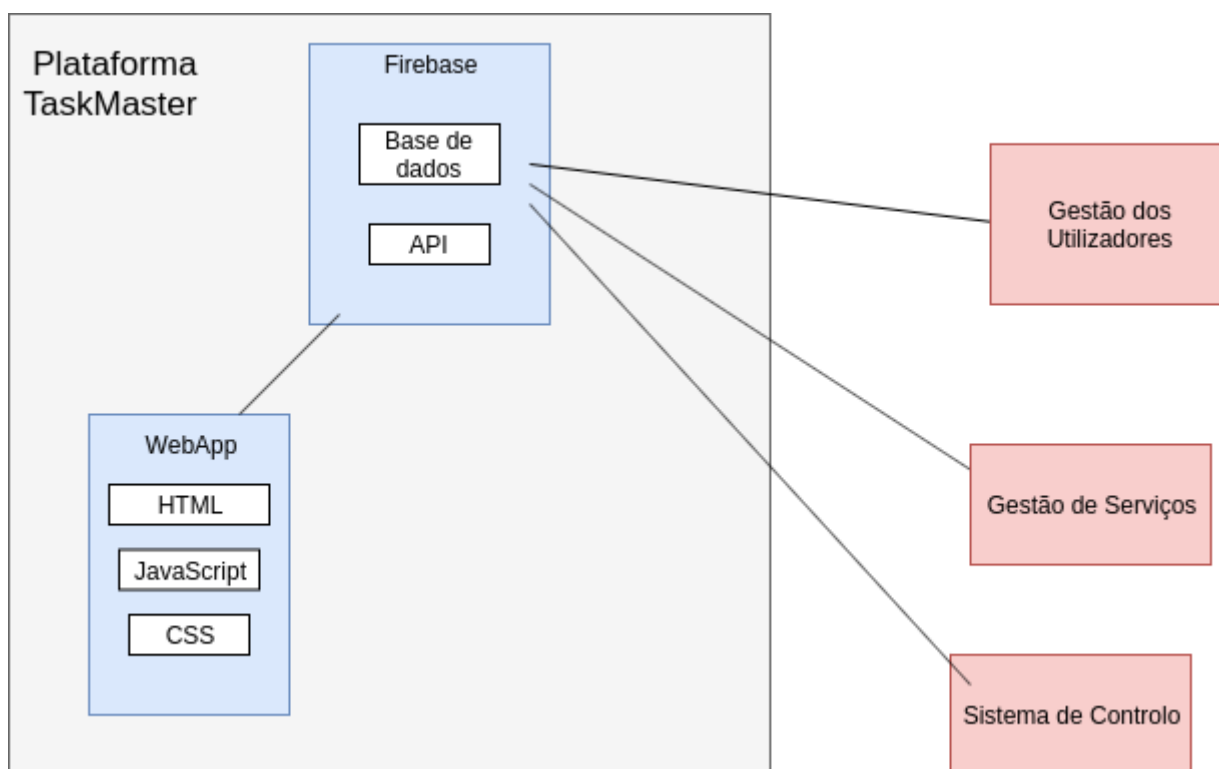


Figura 1 - Diagrama de arquitetura lógica

WebApp: Aplicação ao qual os utilizadores acedem, tendo sido implementada usando HTML JavaScript e CSS

Firebase: Plataforma da Google que permite de forma simples e segura a criação de base de dados e o acesso a esta. Desta forma é a detentora de toda a informação referente a Plataforma TaskMaster.

Gestão dos Utilizadores: Reforça a base de dados e aqui também é armazenada os dados do cliente sendo necessário ter em consideração a segurança e confidencialidade dos mesmos

Gestão de Serviços: Responsável por todo as estáticas referentes ao negócio

Sistema de Controlo: Gere a recolhe da informação relativa às avaliações dos utilizadores bem como a denúncia dos mesmo, filtrando dados importantes para melhor a experiência do utilizador

3.2 Arquitetura operacional

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Modelar num diagrama de *deployment*]

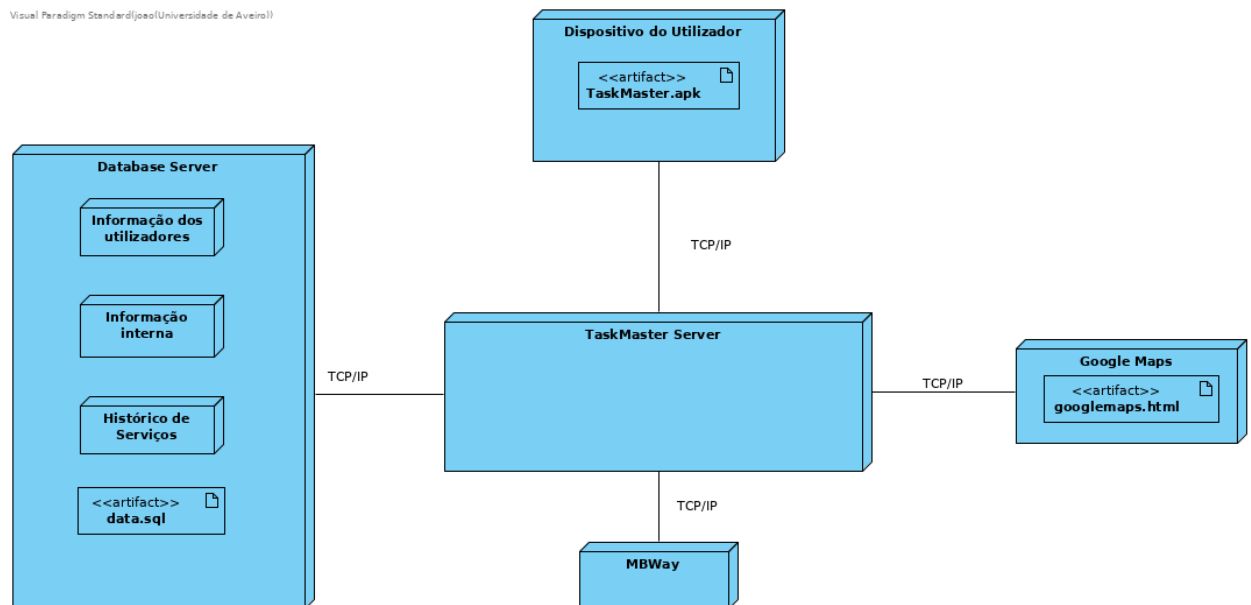


Figura 2 - Diagrama de Deployment

TaskMaster Server : O centro do sistema, responsável pela interligação de todos os outros componentes.

Dispositivo do Utilizador: Terminal que o utilizador usa para interagir com a aplicação.

Database Server: Armazena todos os dados do sistema, tanto relativos aos utilizadores como os serviços.

Informação de utilizadores : Guarda informação referente aos utilizadores.

Histórico de serviços: Armazena os históricos de serviços do utilizador.

Informação interna: Guarda toda a informação referente a anúncios e serviços em progresso.

Google Maps: Fornece todos os dados relativos à localização.

MBWay : Plataforma que permite o pagamento pelos serviços prestado

4 Modelo de comportamento

4.1 Interação entre módulos

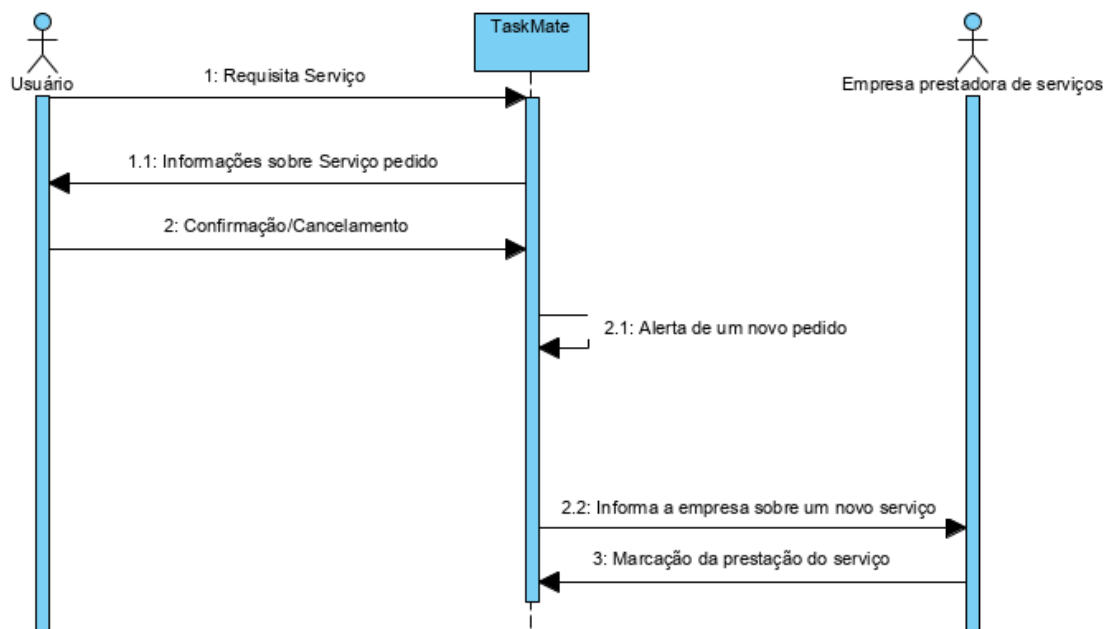
[Tomando como referência os módulos identificados no ponto 3.1, explicar as principais interações que acontecem entre os módulos, a alto nível. Podem ser usados D. de Sequência para mostrar a interação. Não é preciso ser exaustivo, mas deve ficar claro, para um programador que integre a equipa, como é que as partes definidas na arquitetura vão comunicar/colaborar.]

A maneira como os módulos interagem é eficaz, segura e sobretudo simples. Como foi referenciado no ponto 3.1 existe uma classe que tem como objetivo fazer a ligação/paralelismo entre gestão de serviços/controlo e a webApp dita em si. Ou seja, a Firebase tem como objetivo fazer a ligação entre Base de Dados e a API com o HighEnd(html) e com a gestão de serviços.

4.2 Operações de sistema

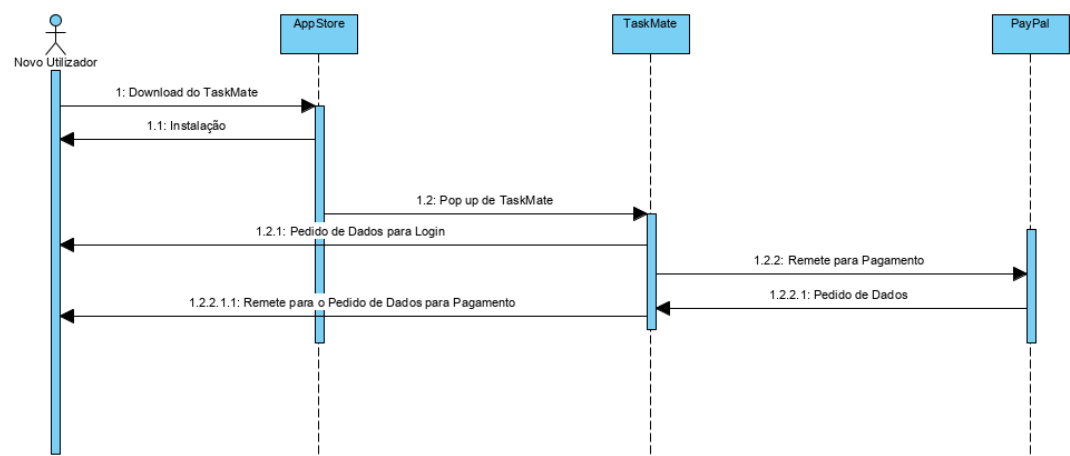
[utilizar Diagramas de Sequência de Sistema para identificar as operações de sistema. Nota: como explorado nas aulas TP, as operações de sistema identificam os pontos de entrada no sistema, a partir dos fluxos dos casos de utilização.

Recomendação: pode-se fazer apenas para um subconjunto dos casos de utilização, justificando que o subconjunto considerado diz respeito aos casos de utilização a implementar na iteração imediata]



4.3 Integrações com sistemas externos

[explicar as colaborações entre o nosso e outros sistemas de software, identificando (a interação com) a **troca de mensagens**., visualizada com diagramas de sequência]



5 Tecnologias e implementação

A framework a utilizar no desenvolvimento da aplicação android será a Firebase. O Firebase é uma plataforma de desenvolvimento de aplicativos para dispositivos móveis e também web.

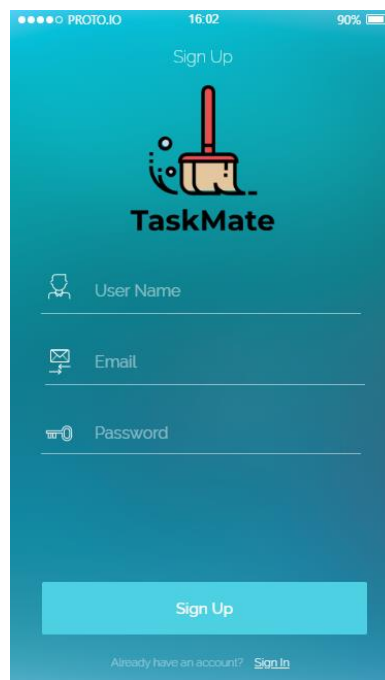
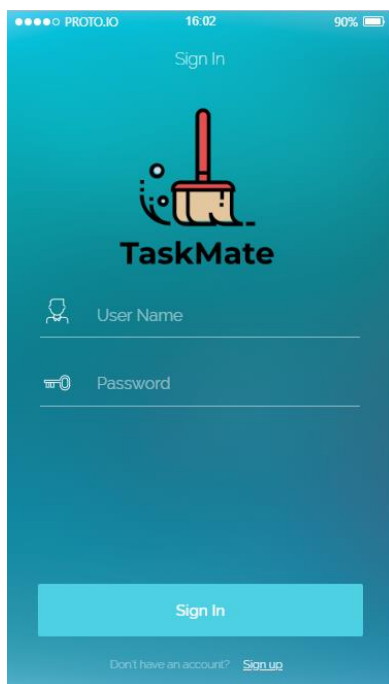
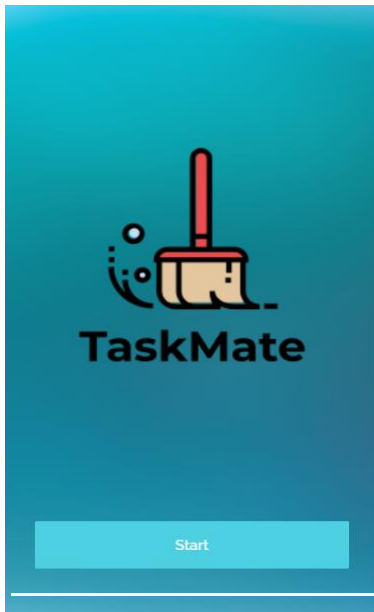
A aplicação a desenvolver está numa plataforma cloud, denominada Cloud Firestore que é um banco de dados de utilizador NoSQL que permite armazenar, sincronizar e consultar dados com facilidade para seus aplicativos móveis e da Web, em escala global.

Como é um ambiente em cloud há a garantia de escalabilidade, desempenho e segurança, permitindo assim a um elevado crescimento do número de clientes.

A autenticação Firebase visa facilitar a criação de sistemas de autenticação segura, e também melhorar a experiência de entrada e integração dos usuários.

6 Protótipos dos produtos

As figuras seguintes apresentam algumas capturas realizadas sobre o protótipo.



5.

6. Portal de Cliente - App Web

Uma vez conectado ao website TaskMate. Irá ter uma página inicial [Figura 6-A] com várias opções de navegação, sendo uma delas “Entrar”.



[Figura 6-A]

Essa subsequente secção [Figura 6-B], permite ao utilizador efetuar o log in na aplicação, utilizando um nome de utilizador e um PIN.



[Figura 6-B]

Aquando da autenticação bem sucedida, será redirecionado imediatamente para a página de anúncios [Figura 6-C], onde o utilizador pode publicar ou visualizar anúncios (e a sua respetiva informação mais relevante) já postados por outros utilizadores.

Peça uma tarefa

21-01-20	Ílhavo
Limpeza	Limpeza simples a entrada de prédio

Anunciar

Anúncios perto de si

31-12-19	Aveiro, Aveiro	Petsitting
Vou para a noite curtir o ano novo. Preciso de alguém que tome conta da minha tartaruga		
12-12-19	Esgueira, Aveiro	Limpeza
Garagem com 9 m² desarrumada e sem manutenção		

[Figura 6-C]

7. 7 Anexos

[listar a documentação anexa, referida ao longo do relatório]

[listar ficheiros externos, nomeadamente versões eletrónicas do relatório, modelos UML a consultar ou sítios com as maquetas]

[anexar cópias de documentos chave relevantes para a documentação de requisitos, e.g., formulários]