

Relatório do trabalho da disciplina de Desenvolvimento Colaborativo de
Software

JACH Resort Hotel

Hélder Martins Carvalho – a15310

João Paulo Figueiredo Carvalho – a15314

António Diogo Rodrigues Fernandes – a15316

Carlos Alberto Monteiro Simões – a15321

Curso Técnico Superior Profissional em Desenvolvimento Web e
Multimédia

Outubro de 2018

Afirmo por minha honra que não recebi qualquer apoio não autorizado na realização deste trabalho prático. Afirmo igualmente que não copiei qualquer material de livro, artigo, documento web ou de qualquer outra fonte exceto onde a origem estiver expressamente citada.

Hélder Martins Carvalho – a15310

João Paulo Figueiredo Carvalho – a15314

António Diogo Rodrigues Fernandes – a15316

Carlos Alberto Monteiro Simões – a15321

Índice

INTRODUÇÃO	1
SUMÁRIO EXECUTIVO	3
Objetivos Principais	3
Principais Funcionalidades	3
Principais Riscos e Obstáculos	3
CONTEXTO DA APLICAÇÃO	5
REQUISITOS AMBIENTAIS	7
Plataforma	7
<i>Hardware</i>	7
<i>Software</i>	7
Linguagens de Programação	7
REQUISITOS FUNCIONAIS	9
REQ 1: Reserva de Quarto	10
Descrição	10
<i>Input</i>	10
Interface	10
Processamento do Sistema	12
<i>Output</i>	12
Constrangimentos	12
Manipulação de Dados	12
REQ 2: <i>Input</i> do Funcionário (<i>Check-in</i>)	13
Descrição	13
<i>Input</i>	13
Interface	13
Processamento do Sistema	15
<i>Output</i>	15
Constrangimentos	15
Manipulação de Dados	15
REQ 3: Autenticação de Utilizadores	17
Descrição	17
<i>Input</i>	17
Interface	17
Processamento do Sistema	19
<i>Output</i>	19
Constrangimentos	19
Manipulação de Dados	19
REQ 4: Consulta Financeira	21
Descrição	21
<i>Input</i>	21
Interface	21
Processamento do Sistema	23

<i>Output</i>	23
Constrangimentos	23
Manipulação de Dados	23
REQ 5: Reserva de Almoço/Jantar	25
Descrição	25
<i>Input</i>	25
Interface	25
Processamento do Sistema	27
<i>Output</i>	27
Constrangimentos	27
Manipulação de Dados	27
REQ 6: Armazenamento de Informação	29
Descrição	29
<i>Input</i>	29
Interface	29
Processamento do Sistema	29
<i>Output</i>	29
Constrangimentos	29
Manipulação de Dados	30
BACKLOG	31
QUALIDADES	35
CRONOGRAMA	37
POTENCIAIS RISCOS	39
PREMISSAS	41
MUDANÇAS FUTURAS	43
GIT VS GITHUB VS GITBUCKET	45
<i>Git</i>	45
<i>Github</i>	45
<i>Gitbucket</i>	45
GLOSSÁRIO	47
BIBLIOGRAFIA	49

Lista de Figuras

Figura 1 - Diagrama de Casos de Uso	5
Figura 2 - Diagrama de Fluxo de Dados	5
Figura 3 - Rede de Petri	6
Figura 4 - Reserva de Quarto	11
Figura 5 - Reserva de Quarto (<i>smartphone</i>)	11
Figura 6 - <i>Input</i> do Funcionário (Reservas)	14

Figura 7 - <i>Input</i> do Funcionário (Check-in)	14
Figura 8 - Autenticação de Utilizadores	18
Figura 9 - Autenticação de Utilizadores (<i>smartphone</i>)	18
Figura 10 - Consulta Financeira	22
Figura 11 - Consulta Financeira (<i>smartphone</i>)	22
Figura 12 - Reserva de Almoço/Jantar	26
Figura 13 - Reserva de Almoço/Jantar (<i>smartphone</i>)	26
Figura 14 - Diagrama de Gantt	33

Lista de Tabelas

Tabela 1 - Backlog	31
--------------------	----

Introdução

Neste documento pretendemos explicar e relatar toda a análise feita para que se torne possível desenvolver o sistema para o JACH Resort Hotel.

A ideia surgiu devido ao facto de os funcionários demorarem muito tempo a realizar o *check-out* dos hóspedes porque tinham de encontrar todas as suas despesas.

O documento é constituído pelo sumário do projeto, o seu contexto e uma lista de requisitos, tanto ambientais como funcionais. Também são mencionadas as qualidades que o sistema deverá ter, um cronograma com os objetivos a serem alcançados, os potenciais riscos do projeto, as suas premissas e mudanças futuras.

Sumário Executivo

Esta ferramenta informática visa a melhorar o bem-estar dos hóspedes de um hotel (sendo estes os principais beneficiários deste sistema) assim como facilitar o trabalho dos funcionários e a gestão das diferentes “áreas” do hotel.

Objetivos Principais

- Reserva de quartos;
- Gestão de hóspedes;
- Gestão financeira;
- Reserva de refeições do restaurante.

Principais Funcionalidades

- Login feito por quarto onde a palavra-passe é regenerada a cada *check-in* (durante a estadia esse hóspede fica associado a esse quarto);
- A reserva pode ser efetuada pelo hóspede antes de fazer o *check-in*;
- A Gestão de hóspedes serve principalmente para fazer o *check-in* dos mesmos, mas também para guardar as suas informações necessárias ao funcionamento do sistema;
- A Gestão financeira serve principalmente para cada hóspede ver quanto terá de pagar quando for efetuar o seu *check-out*;
- No Restaurante podem ser reservadas mesas e ementas, mas também é possível pedir para ser servido no quarto.

Principais Riscos e Obstáculos

- Possível falta de compreensão do sistema por parte do hóspede que pode levar algum tempo a perceber toda a potencialidade e importância do sistema;
- Possível falha momentânea do sistema que fará com que seja necessário os funcionários fazerem todo o trabalho que este primeiro está programado para fazer, o que pode criar alguma demora no normal funcionamento do hotel.

Contexto da Aplicação

Este sistema será usado maioritariamente pelos hóspedes do hotel enquanto estão hospedados no mesmo com vista a melhorar o seu conforto em todo o tipo de atividades.

Para melhor entender o projeto, seguem os seguintes diagramas:

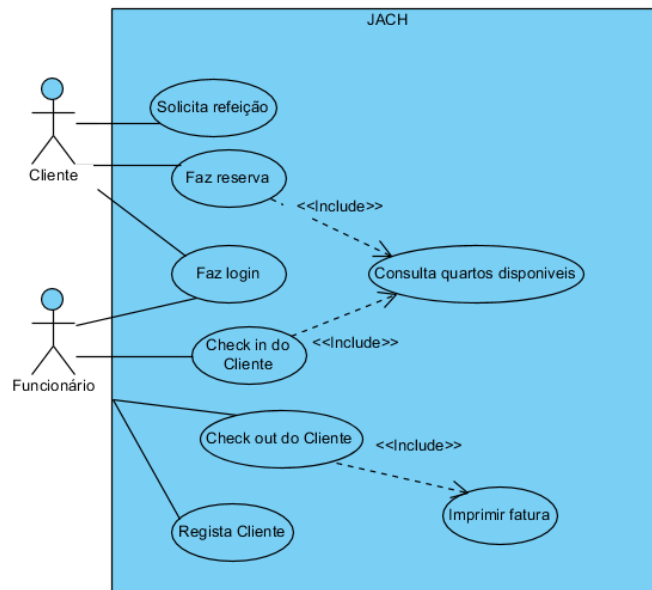


Figura 1 - Diagrama de Casos de Uso

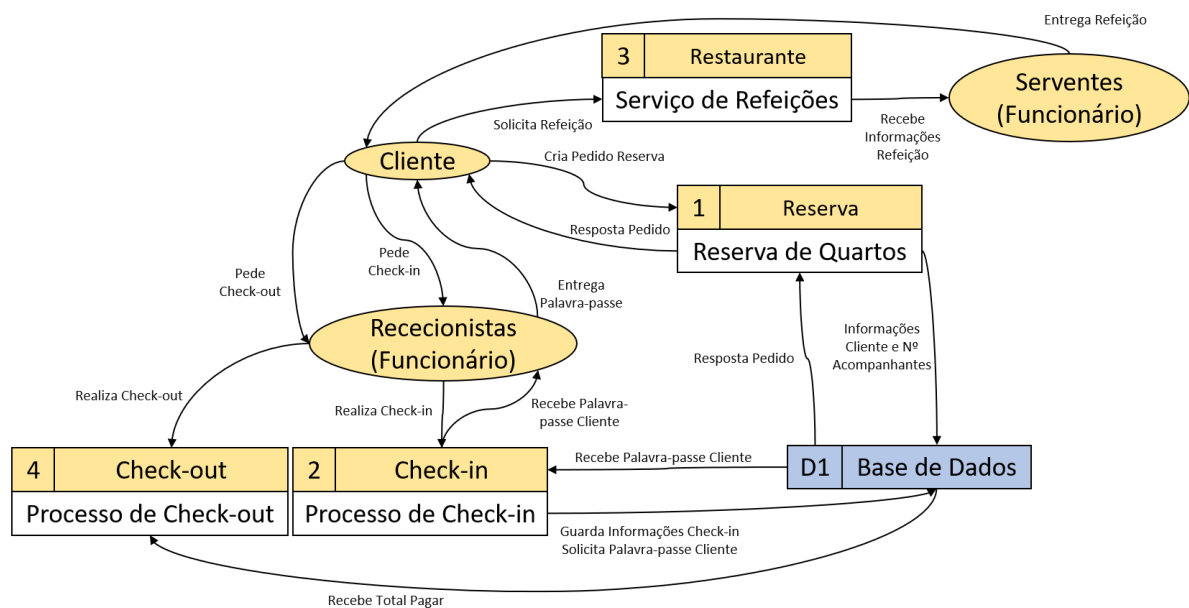


Figura 2 - Diagrama de Fluxo de Dados

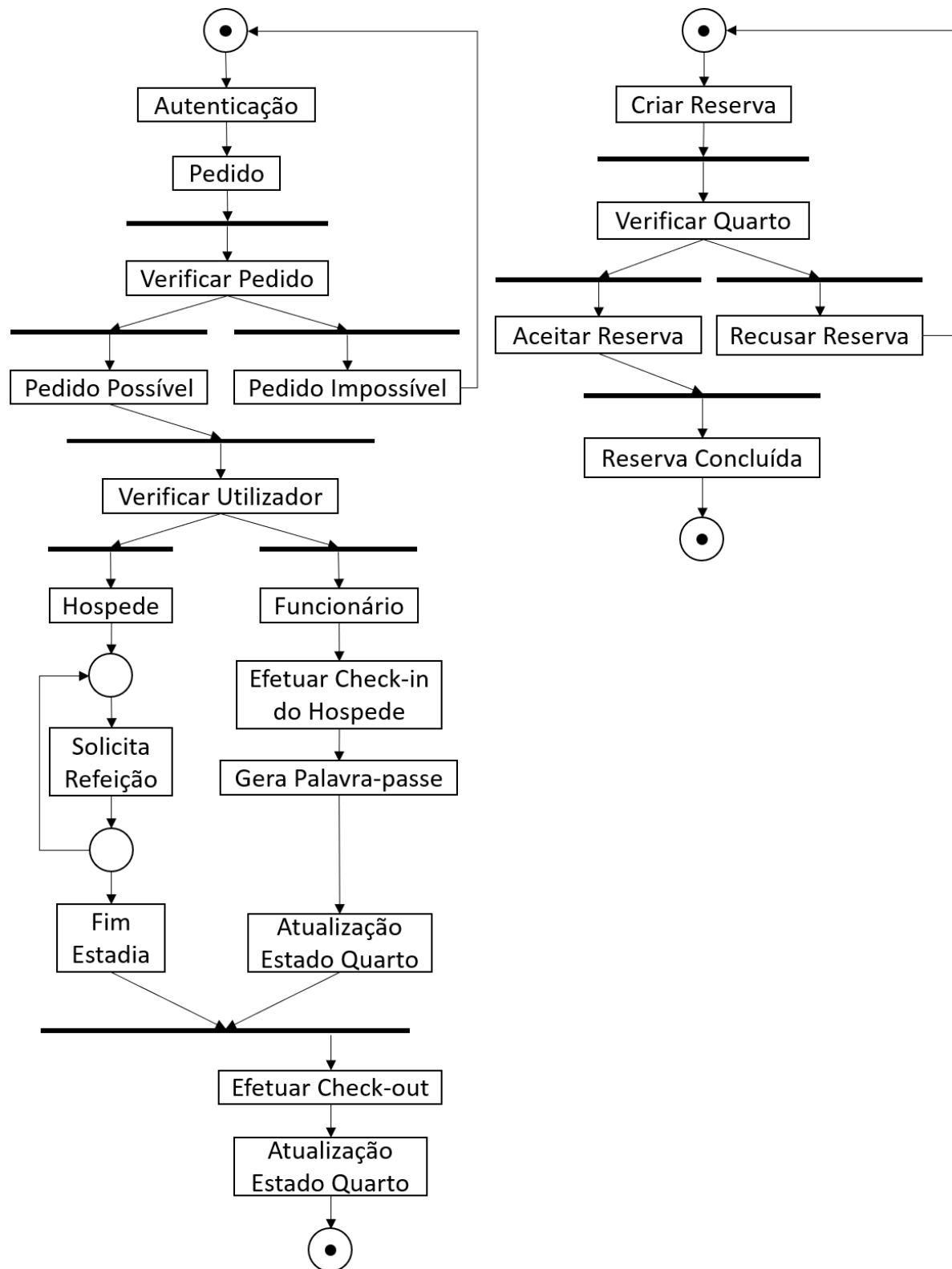


Figura 3 - Rede de Petri

Requisitos Ambientais

Plataforma

Hardware

As máquinas dos funcionários/hospedes não necessitam mais do que um processador *Dual Core*, pelo menos 2 GB de memória *RAM*, um disco rígido de pelo menos 50 GB e terão de estar equipadas com *Windows 7* ou superior, mas também poderá ser utilizada qualquer distribuição *Linux*. Os hospedes poderão também utilizar o seu *smartphone* para aceder ao sistema.

Quanto ao servidor, este tem de estar equipado com qualquer distribuição *Linux* para servidor, pelo menos 1 GB de *RAM*, um disco rígido de pelo menos 100 GB e com os demais variados serviços necessários para o alojamento e funcionamento da aplicação.

Software

O sistema é uma Aplicação *Web* desenvolvida com recurso a programas gratuitos de código aberto como *Apache*, *PHP*, *MySQL*.

Será necessário um navegador para utilizar todo o sistema.

Linguagens de Programação

As linguagens de programação utilizadas serão o *HTML*, *CSS*, *JavaScript*, *PHP* e *SQL*.

Requisitos Funcionais

Esta secção contém a lista de todos os requisitos do sistema identificados com número e nome individual e acompanhados por uma breve descrição dos mesmos. As subsecções seguintes descrevem estes requisitos detalhadamente.

REQ 1: Reserva de Quarto – O sistema deve permitir aos potenciais hóspedes ver informação sobre os quartos como também efetuar a sua reserva;

REQ 2: *Input* do Funcionário – O sistema deve permitir aos funcionários introduzirem a informação necessária sobre o hóspede (fazer *check-in*);

REQ 3: Autenticação de Utilizadores – O sistema deve permitir aos seus vários utilizadores efetuarem a sua autenticação independentemente do seu tipo;

REQ 4: Consulta Financeira – O sistema deve permitir aos hóspedes ver quanto vão ter de pagar no *check-out*;

REQ 5: Reserva de Almoço/Jantar – O sistema deve permitir aos hóspedes reservar o almoço/jantar no restaurante ou pedir para serem servidos no quarto;

REQ 6: Armazenamento de Informação – Uma base de dados deve armazenar toda a informação para assegurar o/a acesso rápido e segurança à/da mesma.

REQ 1: Reserva de Quarto

Descrição

O sistema deve permitir aos potenciais hóspedes ver informação sobre os quartos (capacidade de alojamento, fotografias, preço, equipamentos, etc.) como também efetuar a sua reserva. Este processo não é obrigatório para a realização do *check-in*, mas é uma ajuda porque facilita e adianta o trabalho do funcionário.

Input

O futuro hóspede que estiver a reservar o quarto deve introduzir a informação pedida. O sistema deve ter 6 campos nos quais o hóspede deve introduzir os dados corretos para poder fazer a reserva daquele quarto.

Os primeiros campos são designados “Nome do Hóspede”. Devem ser campos separados para o primeiro e último nome. Uma *string* é o tipo de dados a ser introduzido nestes campos que são de preenchimento obrigatório do tipo “*text*”.

O campo “Número de Telemóvel” é um campo de preenchimento obrigatório do tipo “*number*” onde o futuro hóspede apenas pode introduzir um conjunto de 9 números.

O campo “Número de Acompanhantes” é um campo de preenchimento obrigatório do tipo “*select*” onde deve ser selecionado o número de acompanhantes de quarto que o futuro hóspede vai levar para aquele quarto.

Os campos “Data de *Check-in*” e “Data de *Check-out*” servem para inserir a data inicial e data final da estadia com o formato AAAA-MM-DD, onde A é referente ao ano, M ao mês e D ao dia sendo que cada letra representa um número de 0 a 9. Um “*date*” é o tipo de dados a ser introduzido nestes campos que são de preenchimento opcional do tipo “*date*”.

Perto do fim do ecrã está localizado um botão para processar a informação. O futuro hóspede deve pressionar este botão quando finalizar a introdução de toda a informação necessária.

Interface

Os 6 campos devem estar posicionados em forma de grelha, em grupos de 2 ou 3 por linha, de forma a evitar que o futuro hóspede tenha de fazer *scroll* na página para os preencher. O botão de “Reservar” deve estar centrado abaixo de todos os campos.

JACH Resort Hotel

Quarto Médio

Nome:

Apelido:

Número de telemóvel:

Número de acompanhantes que vai levar (não conte você):

Data de check-in:

Data de check-out:

Reservar

Figura 4 - Reserva de Quarto

JACH Resort Hotel

Quarto Médio

Nome:

Apelido:

Número de telemóvel:

Número de acompanhantes que vai levar (não conte você):

Data de check-in:

Data de check-out:

Reservar

Figura 5 - Reserva de Quarto (smartphone)

Processamento do Sistema

O sistema está essencialmente inativo quando o futuro hospede está a introduzir os dados relativos à reserva do quarto. Quando o botão de “Reservar” for pressionado o sistema verifica se o futuro hospede introduziu toda a informação necessária assim como se existe algum quarto do tipo que escolheu livre. Caso o futuro hospede não tenha preenchido todos os campos o sistema deve-o alertar para o fazer. O sistema deve mostrar quais os campos que não foram preenchidos para que o futuro hospede seja capaz de os preencher. Caso não existam quartos livres, o sistema deve alertar o futuro hospede para o mesmo. Quando a informação for aceite pelo sistema, a mesma é enviada para o servidor. O servidor coloca a informação nas respetivas tabelas na base de dados.

Output

O sistema deve mostrar uma mensagem de erro caso alguma da informação introduzida esteja diferente da pretendida pelos campos. Nesta página de “Reserva de Quarto”, onde o futuro hospede insere os dados, o sistema deve alertar o mesmo sobre a informação incorreta mostrando um balão junto ao respetivo campo. Deve também mostrar uma mensagem de erro caso não existam quartos daquele tipo livres.

Quando o futuro hospede pressionar o botão de “Reservar” e a informação que foi inserida estiver correta e existirem quartos daquele tipo livres, o sistema deve avançar para a página onde aparecerá um resumo de toda a informação inserida anteriormente junto com uma mensagem de sucesso, o que fará com que o futuro hospede saiba que a reserva do quarto foi registada com sucesso. Toda a informação introduzida pelo futuro hospede deve ficar armazenada na base de dados do servidor.

Constrangimentos

É possível que não existam quartos do tipo escolhido livres para a data que o futuro hospede indicou na sua reserva, o que fará com que este processo não possa ser concluído (pelo menos para aquele tipo de quarto).

Manipulação de Dados

A base de dados referida no “REQ 6: Armazenamento de Informação” é necessária para guardar tudo o que o futuro hospede inseriu.

REQ 2: *Input* do Funcionário (*Check-in*)

Descrição

O sistema deve permitir aos funcionários introduzirem a informação necessária sobre o hospede efetuando o *check-in* do mesmo. Este é o processo inicial que permitirá a esse hospede ter acesso ao sistema do hotel.

Input

O funcionário que estiver a recolher a informação do hospede para fazer o seu *check-in* deve introduzi-la na interface do sistema. O sistema deve ter 7 campos nos quais o funcionário deve introduzir dados para poder fazer o *check-in* do hospede.

Os primeiros campos são designados “Nome do Hospede”. Devem ser campos separados para o primeiro e último nome. Uma *string* é o tipo de dados a ser introduzido nestes campos que são de preenchimento obrigatório do tipo “*text*”.

O campo “Número de Contribuinte” e “Número de Telemóvel” são campos de preenchimento obrigatório do tipo “*number*” onde o funcionário apenas pode introduzir um conjunto de 9 números.

O campo “Número do Quarto” é um campo de preenchimento obrigatório do tipo “*select*” onde o funcionário deve escolher o quarto onde o hospede vai ficar alojado (neste campo só aparecerão os quartos livres).

O campo “Número de Acompanhantes” é um campo de preenchimento obrigatório do tipo “*select*” onde deve ser selecionado o número de acompanhantes de quarto daquele hospede.

O campo “Data de *Check-out*” é onde o funcionário deve inserir a data final da estadia com o formato AAAA-MM-DD, onde A é referente ao ano, M ao mês e D ao dia sendo que cada letra representa um número de 0 a 9. Um “*date*” é o tipo de dados a ser introduzido nestes campos que são de preenchimento opcional do tipo “*date*”.

Perto do fim do ecrã está localizado um botão para processar a informação. O funcionário deve pressionar este botão quando finalizar a introdução de toda a informação necessária.

Interface

Este requisito é constituído por duas interfaces onde a primeira é uma lista das reservas criadas (caso o funcionário esteja a fazer a reserva de algum hospede que já reservou o quarto) e a segunda é um formulário de dados para o “*Check-in*”.

A primeira é composta por uma tabela constituída por todas as reservas em que cada linha da tabela terá um botão para realizar o “Check-in” daquele hospede (que já reservou). Esse botão redireciona o funcionário para a segunda página onde o formulário já estará todo preenchido com os dados do hospede.

A segunda é constituída por 7 campos que devem estar posicionados em forma de grelha, em grupos de 2 ou 3 por linha, de forma a evitar que o funcionário tenha de fazer *scroll* na página para os preencher. O botão de “Check-In” deve estar centrado abaixo de todos os campos. Deve também existir um botão de “Lista de Reservas” para que o funcionário possa ser redirecionado para a primeira página descrita em cima.

JACH Resort Hotel

Reservas

Nome: Apelido:

Nome	Apelido	Quarto	+
João	Carvalho	Pequeno	<input type="button" value="Check-in"/>
António	Fernandes	Médio	<input type="button" value="Check-in"/>
Carlos	Simões	Médio	<input type="button" value="Check-in"/>
Hélder	Carvalho	Grande	<input type="button" value="Check-in"/>

Início Quartos Contactos Sair

Figura 6 - Input do Funcionário (Reservas)

JACH Resort Hotel

Check-in

Nome: Apelido:

Número de contribuinte: Número de telemóvel:

Número do quarto: Número de acompanhantes (não contar o que registou):

Data de check-out:

Início Quartos Contactos Sair

Figura 7 - Input do Funcionário (Check-in)

Processamento do Sistema

O sistema está essencialmente inativo quando o funcionário está a introduzir os dados relativos ao *check-in* do hospede. Quando o botão de “*Check-In*” for pressionado o sistema verifica se o funcionário introduziu toda a informação necessária. Caso o funcionário não tenha preenchido todos os campos o sistema deve-o alertar para o fazer. O sistema deve mostrar quais os campos que não foram preenchidos para que o funcionário seja capaz de os preencher.

Caso o “*Check-in*” esteja a ser efetuado a partir de uma reserva, quando o funcionário pressionar o botão de “*Check-in*” da primeira página, deve ser redirecionado para a segunda já com o formulário de *check-in* preenchido, para que apenas seja necessário o funcionário confirmar os dados.

Quando a informação for aceite pelo sistema, a mesma é enviada para o servidor. O servidor coloca a informação nas respetivas tabelas na base de dados.

Output

O sistema deve mostrar uma mensagem de erro caso alguma da informação introduzida esteja diferente da pretendida pelos campos. Nesta página de “*Check-In*”, onde o funcionário insere os dados, o sistema deve alertar o mesmo sobre a informação incorreta mostrando um balão junto ao respetivo campo.

Quando o funcionário pressionar o botão de “*Check-In*” e a informação que foi inserida estiver correta, o sistema deve avançar para a página onde aparecerá o número de aluguer a palavra-passe que esse hospede utilizará para entrar no sistema a ele destinado, o que fará com que o funcionário saiba que o *check-in* do hospede foi registado com sucesso. Toda a informação introduzida pelo funcionário deve ficar armazenada na base de dados do servidor.

Constrangimentos

O tempo de *check-in* depende da velocidade de escrita do funcionário assim como a velocidade de fornecimento da informação pedida ao hospede.

Manipulação de Dados

A base de dados referida no “REQ 6: Armazenamento de Informação” é necessária para guardar tudo o que o funcionário inseriu.

REQ 3: Autenticação de Utilizadores

Descrição

O sistema deve permitir aos seus vários utilizadores efetuarem a sua autenticação independentemente do seu tipo. Este processo é o único que permite entrar na área reservada a utilizadores.

Input

Os hóspedes que já tiverem o *check-in* realizado, ou qualquer funcionário, podem iniciar sessão no sistema. O sistema deve ter 2 campos nos quais o utilizador deve introduzir dados para poder entrar na sua área reservada.

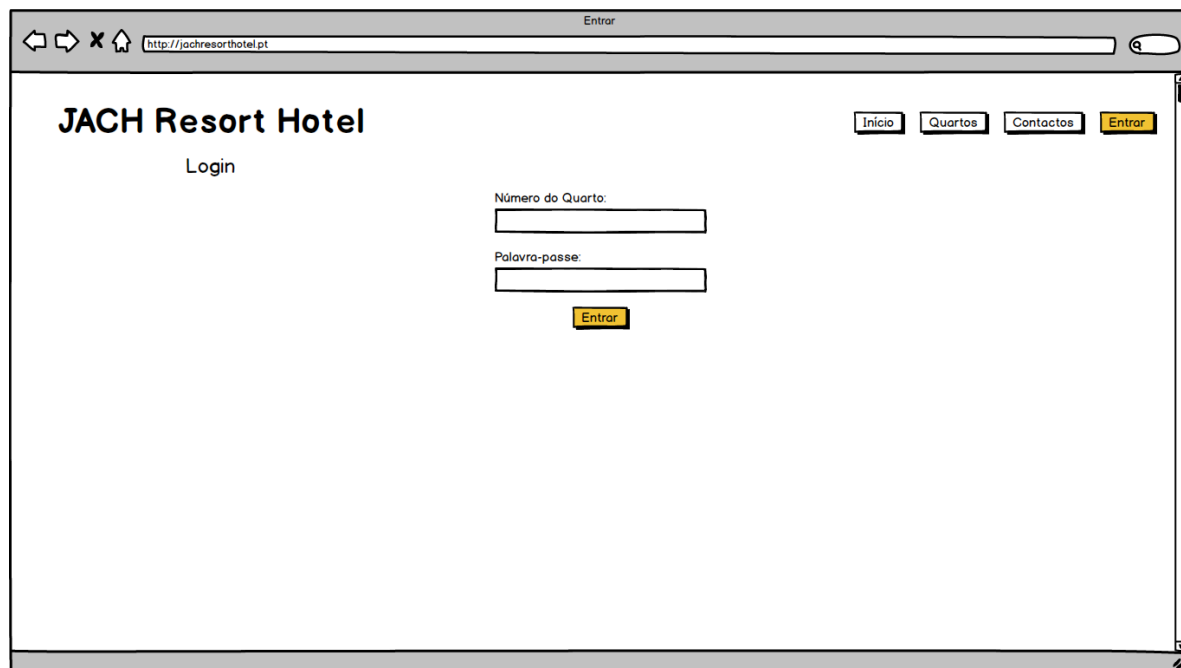
O primeiro campo é designado “Número de Aluguer”. Este de preenchimento obrigatório do tipo “*number*”. Os hóspedes introduzirão o número de aluguer (obtido no momento do *check-in*) enquanto que os funcionários introduzirão o seu código identificativo.

O campo “Palavra-passe” é um campo de preenchimento obrigatório do tipo “*password*” onde o utilizador deve introduzir um conjunto de caracteres. No caso do hóspede a sua palavra-passe é gerada no momento do *check-in* enquanto que no caso do funcionário é definida quando a sua conta é criada.

Por baixo destes campos está localizado um botão para processar a informação. O utilizador deve pressionar este botão quando finalizar a introdução de toda a informação necessária.

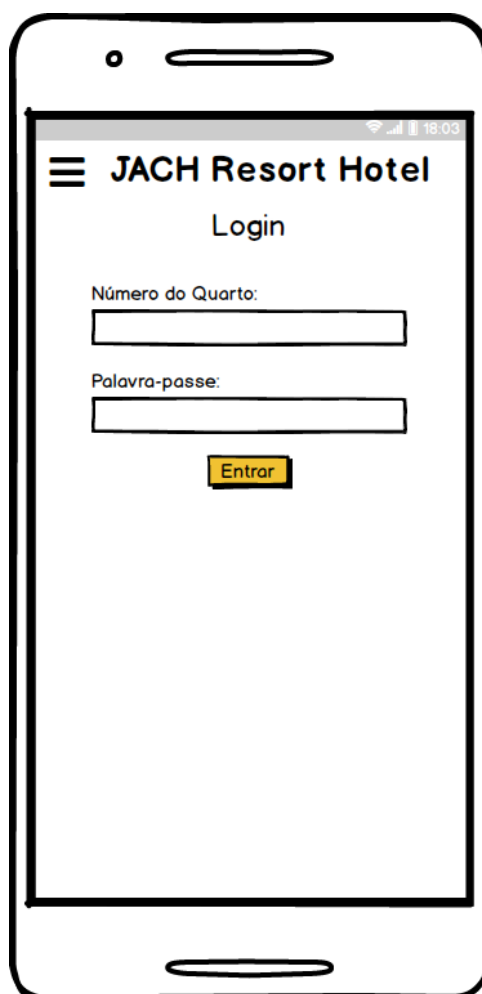
Interface

Os 2 campos devem estar posicionados um abaixo do outro de forma a serem fáceis de preencher. O botão de “Entrar” deve estar centrado abaixo dos campos.



The image shows a web browser window with the address bar displaying "http://jachresorthotel.pt". The page title is "JACH Resort Hotel". Below the title, the word "Login" is centered. On the right side of the page, there are four navigation buttons: "Início", "Quartos", "Contactos", and "Entrar". The "Entrar" button is highlighted in yellow. In the center of the page, there are two input fields: "Número do Quarto:" and "Palavra-passe:". Below these fields is a yellow "Entrar" button.

Figura 8 - Autenticação de Utilizadores



The image shows a smartphone screen displaying the JACH Resort Hotel login page. The status bar at the top shows the time as 18:03. The page title is "JACH Resort Hotel" with a hamburger menu icon to the left. Below the title, the word "Login" is centered. On the right side of the page, there are four navigation buttons: "Início", "Quartos", "Contactos", and "Entrar". The "Entrar" button is highlighted in yellow. In the center of the page, there are two input fields: "Número do Quarto:" and "Palavra-passe:". Below these fields is a yellow "Entrar" button.

Figura 9 - Autenticação de Utilizadores (smartphone)

Processamento do Sistema

O sistema está essencialmente inativo quando o utilizador está a introduzir os dados de acesso. Quando o botão de “Entrar” for pressionado o sistema verifica se o utilizador introduziu toda a informação necessária. Caso o utilizador não tenha preenchido todos os campos o sistema deve-o alertar para o fazer. O sistema deve mostrar quais os campos que não foram preenchidos para que o utilizador seja capaz de os preencher. Quando a informação for aceite pelo sistema, a mesma é enviada para o servidor que verificará se estão corretas.

Output

O sistema deve mostrar uma mensagem de erro caso alguma da informação introduzida esteja diferente da pretendida pelos campos. Nesta página de “Entrar”, onde o utilizador insere os dados, o sistema deve alertar o mesmo sobre a informação incorreta mostrando um balão junto ao respetivo campo.

Quando o utilizador pressionar o botão de “Entrar” e a informação que foi inserida estiver correta, o sistema avançará para a página de resumo que disponibilizará o acesso ao processo referido nos “REQ 4: Consulta Financeira” e “REQ 5: Reserva de Almoço/Jantar”. Caso os dados introduzidos estejam errados, o sistema deve mostrar uma mensagem de erro e voltar para a página principal deste processo.

Constrangimentos

O utilizador pode eventualmente esquecer-se da sua palavra-passe o que fará com que tenha de pedir a um funcionário para a regenerar.

Manipulação de Dados

A base de dados referida no “REQ 6: Armazenamento de Informação” é necessária para consultar as credenciais de acesso.

REQ 4: Consulta Financeira

Descrição

O sistema deve permitir aos hóspedes ver quanto vão ter de pagar quando realizarem o *check-out*. Este processo ver uma lista de despesas, ou seja, permite ver tudo pormenorizado sobre o que o hóspede vai pagar.

Input

O hóspede poderá aplicar filtros para poder ver apenas as despesas que lhe interessarem em particular. O sistema deve ter 3 campos nos quais o hóspede pode introduzir dados para poder filtrar a sua pesquisa.

Os primeiros campos são designados “Intervalo de Tempo”. Devem ser campos separados para a data inicial e data final com o formato AAAA-MM-DD, onde A é referente ao ano, M ao mês e D ao dia sendo que cada letra representa um número de 0 a 9. Um “*date*” é o tipo de dados a ser introduzido nestes campos que são de preenchimento opcional do tipo “*date*”.

O campo “Tipo de Despesa” é um campo de preenchimento opcional do tipo “*select*” onde o hóspede deve selecionar qual o tipo de despesa que quer ver.

Por baixo destes campos está localizado um botão para processar a informação. O hóspede pode pressionar este botão se finalizar a introdução de toda a informação necessária.

Interface

Os 3 campos devem estar posicionados em forma de grelha, juntando o máximo possível por linha, de forma a evitar que ocupem muito espaço na página, visto que são todos de preenchimento opcional. O botão de “Pesquisar” deve estar alinhado à esquerda abaixo de todos os campos (caso existam duas ou mais linhas de campos) ou alinhado à direita na mesma linha dos campos (caso só exista essa linha).

JACH Resort Hotel

Consulta Financeira

Data inicial: Data final: Tipo de despesa:

Nome	Tipo de despesa	Dia	Valor
Dia	Quarto	27/10/2018	50€
Almoço	Restaurante	27/10/2018	20€
Jantar	Restaurante	27/10/2018	20€
Dia	Quarto	28/10/2018	50€
Almoço	Restaurante	28/10/2018	20€
Jantar	Restaurante	28/10/2018	20€

Total a pagar: 180€

Figura 10 - Consulta Financeira

JACH Resort Hotel

Consulta Financeira

Data inicial: Data final: Tipo de despesa:

Nome	Tipo de despesa	Dia	Valor
Dia	Quarto	27/10/2018	50€
Almoço	Restaurante	27/10/2018	20€
Jantar	Restaurante	27/10/2018	20€
Dia	Quarto	28/10/2018	50€
Almoço	Restaurante	28/10/2018	20€
Jantar	Restaurante	28/10/2018	20€

Total a pagar: 180€

Figura 11 - Consulta Financeira (smartphone)

Processamento do Sistema

O sistema está essencialmente inativo quando o hospede está a introduzir os dados pesquisa. Se o botão de “Pesquisar” for pressionado o sistema verifica se o hospede introduziu a informação corretamente. Caso o hospede não tenha preenchido algum campo corretamente o sistema deve-o alertar para o fazer. Quando a informação for aceite pelo sistema, a mesma é enviada para o servidor que verificará se estão corretas, executará o respetivo comando de pesquisa na Base de Dados e enviará de volta apenas a informação pedida pelo hospede.

Output

O sistema deve apresentar uma mensagem de erro caso alguma da informação introduzida esteja diferente da pretendida pelos campos mostrando um balão junto ao respetivo campo.

Quando o utilizador pressionar o botão de “Pesquisar” e a informação que foi inserida estiver válida, o sistema atualizará a página mostrando agora apenas a informação que o hospede pediu.

Constrangimentos

O hospede pode não ter nenhuma despesa registada relativamente aos parâmetros que introduziu na pesquisa.

Manipulação de Dados

A base de dados referida no “REQ 6: Armazenamento de Informação” é necessária para devolver a informação que o hospede pediu.

REQ 5: Reserva de Almoço/Jantar

Descrição

O sistema deve permitir aos hóspedes reservar o almoço/jantar no restaurante ou pedir para serem servidos no quarto. Este processo permite ver uma ementa onde o hóspede poderá escolher entre os diversos pratos, bebidas, sopas e sobremesas disponíveis juntamente com o preço total da refeição.

Input

O sistema deve ter 9 campos (ou mais, dependendo do número de produtos que o hóspede quiser pedir) nos quais o hóspede deve seleccionar o prato/bebida/sopa/sobremesa que quer e a respetiva quantidade.

Deve existir uma secção de campos para cada parte da refeição (prato, bebida, sopa, sobremesa). Em cada secção deve existir um campo do tipo “*select*” de preenchimento obrigatório onde o hóspede deverá seleccionar o produto que quer (relativo a essa secção) e outro campo “Quantidade” de preenchimento obrigatório do tipo “*number*” onde o hóspede deverá introduzir a quantidade que quer do anterior produto.

O campo “Local a Servir” é um campo de preenchimento obrigatório do tipo “*select*” onde o hóspede deverá seleccionar onde quer que lhe seja servida a refeição.

Por baixo destes campos está localizado um botão para processar a informação. O hóspede deve pressionar este botão quando finalizar a introdução de toda a informação necessária.

Interface

Cada linha tem 2 campos (o da seleção do produto e à frente o da quantidade) que por sua vez está inserida na respetiva secção (parte da refeição). No fim de cada secção existirá um botão de “Adicionar Produto” que criará uma nova linha com os mesmos campos onde o hóspede poderá escolher outro produto e a sua quantidade. As secções estarão uma abaixo de outra e o campo “Local a Servir” deve estar abaixo de todas as secções (alinhado à esquerda) na mesma linha com o botão de “Encomendar” (alinhado à direita).

The desktop view of the reservation form for JACH Resort Hotel. The browser address bar shows 'http://jachresorthotel.pt'. The page title is 'Reserva de Almoço/Jantar'. Navigation links include 'Início', 'Quartos', 'Contactos', and 'Sair'. The form is organized into four sections: 'Prato' (Main Course), 'Sopa' (Soup), 'Bebida' (Beverage), and 'Sobremesa' (Dessert). Each section contains a 'Produto:' dropdown menu, a 'Quantidade:' input field with a vertical spinner, and a yellow '+' button. At the bottom right, it shows 'Total a pagar: 0€', a 'Local a Servir:' dropdown menu set to 'Restaurante', and a yellow 'Encomendar' button.

Figura 12 - Reserva de Almoço/Jantar

The mobile view of the reservation form for JACH Resort Hotel, shown on two smartphone screens. The left screen displays the 'Prato' and 'Sopa' sections. The right screen displays the 'Bebida' and 'Sobremesa' sections, with a 'scroll' arrow indicating the transition between the two views. Each section contains a 'Produto:' dropdown menu, a 'Quantidade:' input field with a vertical spinner, and a yellow '+' button. At the bottom right, it shows 'Total a pagar: 0€', a 'Local a Servir:' dropdown menu set to 'Restaurante', and a yellow 'Encomendar' button.

Figura 13 - Reserva de Almoço/Jantar (smartphone)

Processamento do Sistema

O sistema está essencialmente inativo quando o hospede está a introduzir os dados pesquisa. Quando o botão “Adicionar Produto” for pressionado será executada uma função na máquina do hospede que fará a linha mencionada anteriormente aparecer. Relativamente ao botão “Encomendar”, quando este for pressionado o sistema vai verificar se o hospede introduziu a informação corretamente. Caso o hospede não tenha preenchido algum campo corretamente o sistema deve-o alertar para o fazer assim como o deve avisar caso não tenha preenchido algum. Toda a informação introduzida pelo hospede deve ficar armazenada na base de dados do servidor. O servidor coloca a informação nas respetivas tabelas na base de dados.

Output

O sistema deve apresentar uma mensagem de erro caso alguma da informação introduzida esteja diferente da pretendida pelos campos mostrando um balão junto ao respetivo campo.

Quando o utilizador pressionar o botão de “Encomendar” e a informação que foi inserida estiver válida, o sistema avançará para a página que mostrará o resumo da refeição acabada de encomendar. Toda a informação introduzida pelo hospede deve ficar armazenada na base de dados do servidor.

Constrangimentos

O hospede pode eventualmente, por engano, encomendar quantidades de produtos maiores do que as que queria ter encomendado.

Manipulação de Dados

A base de dados referida no “REQ 6: Armazenamento de Informação” é necessária para guardar tudo o que o funcionário inseriu.

REQ 6: Armazenamento de Informação

Descrição

O sistema deve disponibilizar uma base de dados *MySQL* para permitir acesso rápido à informação e assegurar que a mesma está protegida.

Input

Os utilizadores devem introduzir informação nos campos das interfaces dos requisitos mencionados anteriormente. Essa informação é então guardada na respetiva tabela na base de dados.

Interface

A interface do sistema deve mostrar informação proveniente da base de dados.

Processamento do Sistema

Sempre que algum utilizador pressionar algum botão que envie informação para a base de dados, os dados são enviados através da rede para o servidor. Esse servidor coloca os dados nas respetivas tabelas da base de dados. Os utilizadores podem também pedir informação à base de dados e a base de dados deve responder a esse pedido enviando os dados pedidos para o computador do utilizador que processará a informação recebida e a mostrará na sua interface.

Output

Nada a apontar.

Constrangimentos

A quantidade de tráfego que a base de dados pode tratar é determinada pela velocidade de processamento do servidor e pela velocidade da rede. Um servidor dedicado será necessário caso o número de conexões recebidas avolumado.

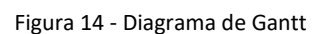
Manipulação de Dados

Todos os dados são manipulados pela base de dados como descrito acima.

Backlog

Tabela 1 - Backlog

Lista de itens	Estimativa (pontos)
Análise	50
Desenho	25
Permitir aos utilizadores fazerem reservas	10
Permitir ao funcionário fazer check-in dos hóspedes	20
Permitir aos hóspedes e funcionários a sua autenticação	15
Permitir aos hóspedes consultarem quanto têm de pagar no check-out	10
Permitir aos hóspedes reservarem o almoço/jantar	15
Testes	40
Total:	185



Qualidades

O sistema tem de ter:

- Exatidão – tudo o que for introduzido no sistema tem de ser utilizado para devolver informação correta (não poderá haver erros no cálculo do total a pagar pelo hospede no momento do *check out*);
- Confiabilidade – o sistema não deve ter falhas;
- Eficiência – os dados que forem necessários ser introduzidos no sistema devem poder ser introduzidos rapidamente, o sistema deve operar sem lentidão provocada por *hardware*, deve poder operar aceitavelmente quando, eventualmente, estiver em processamento máximo, deve também operar em *hardware* comum como também em qualquer sistema operativo de servidor;
- Integridade – o sistema deve preservar corretamente os dados, não criando duplicados indesejados;
- Usabilidade – o sistema não deve ter problemas de utilização da interface de utilizador;
- Manutenção – o sistema deve ser desenvolvido de maneira a facilitar a introdução de novas funcionalidades assim como a resolução de problemas;
- Flexibilidade – o sistema deve poder receber novas funcionalidades;
- Portabilidade – o sistema deve ser extremamente portátil visto que deverá existir a possibilidade de ser utilizado em *smartphones*;
- Robustez – o sistema deve ser capaz de lidar com erros/falhas durante a execução e com entradas de dados erradas/inválidas/inesperadas;
- Segurança – o sistema deve ser capaz de proteger toda a informação de ataques maliciosos mantendo, em simultâneo, a sua perfeita execução.

Cronograma

O sistema deve estar pronto até um certo dia, a definir.

Devem ser alcançados pelo menos estes dois objetivos:

- Implementação concluída;
- Testes completos.

Potenciais Riscos

Este projeto poderá ter os seguintes potenciais riscos:

- Cronograma – devido a escassos recursos e falhas ao identificar funcionalidades complexas pode levar ao cálculo errado da estimativa de tempo necessário para o desenvolvimento do sistema;
- Orçamental – pode existir estimativa errada do orçamento (devido aos riscos de Cronograma, por exemplo) o que por consequência fará com que seja excedido;
- Técnicos – podem existir mudanças frequentes nos requisitos devido à complexidade de implementar o sistema;
- Programáticos – pode existir desenvolvimento no mercado, alteração da legislação governamental e alteração da estratégia de negócio por parte do dono do sistema.

Premissas

Neste projeto iremos assumir como garantidas as seguintes suposições:

- Recursos – os utilizadores finais serão os principais elementos a testar o sistema em contexto real de trabalho;
- Entrega – o servidor chega configurado como esperado e o número correto de dispositivos utilizados pelos funcionários, se ainda não existirem, serão entregues na data prevista sem atrasos;
- Orçamento – os custos do projeto permanecerão iguais aos custos inicialmente orçados;
- Escopo – o escopo do projeto não será alterado a partir do momento em que as partes interessadas concordarem que o mesmo está bem definido;
- Cronograma – os objetivos serão atingidos antes acabarem os respetivos prazos;
- Metodologia – o projeto irá seguir a metodologia em cascata;
- Tecnologia – a equipa irá escrever a solução com recurso a “HTML”, “CSS”, “JavaScript”, “PHP” e “MySQL”;
- Arquitetura e Design – a solução utilizará a arquitetura de “REST API”.

Mudanças Futuras

O sistema deve ser desenvolvido de maneira a poder receber novas funcionalidades facilmente.

Algumas funcionalidades não importantes para o principal objetivo do sistema são:

- Gestão de espaços:
 - Reserva de espreguiçadeiras na piscina;
 - Reserva de equipamento e horas de utilização para Golf, Ténis e Futebol;
 - Reserva de tratamentos no *Spa* com hora marcada.

Git vs Github vs Gitbucket

Git

Git é o sistema distribuído de controlo de versões mais utilizado do mundo. É um projeto de código aberto criado em 2005 por Linus Torvalds.

Este é o sistema base em que muitos outros estão dependentes como os dois abaixo mencionados, *Github* e *Gitbucket*.

É geralmente uma linha de comandos onde podem ser escritos comandos para realizar as nossas tarefas.

O *Git* faz a gestão local de um repositório alojado por exemplo no *Github*, podemos fazer modificações a esse repositório sendo que o *Git* apenas faz essas modificações localmente. Quando estivermos satisfeitos podemos ordenar o *Git* a enviar essas modificações para o repositório alojado *online*.

Github

Github é um local para partilhar código com amigos, colegas de trabalho, colegas de turma e estranhos. Mais de 31 milhões de pessoas usam o *Github* e é por isso que é uma boa forma de partilhar código. Está também integrado em muitas ferramentas de programação que facilitam ainda mais a sua utilização.

É possível criar repositórios públicos e privados, os privados só estão disponíveis em contas “Premium”, é fácil de utilizar, tem uma interface intuitiva, não tem limite de colaboradores associados ao mesmo repositório, tem a possibilidade de sinalizar um problema no código do projeto para que outros colaboradores possam ver, acompanhar e até resolver, tem a possibilidade de criar um quadro de planeamento do projeto para que os colaboradores possam ver as tarefas que estão em aberto, a ser realizadas, etc., tem a possibilidade de criar uma *Wiki* que documenta todo o projeto no repositório e por fim disponibiliza gráficos e estatísticas do repositório.

Gitbucket

Gitbucket é um local para partilhar código semelhante ao *Github*. Surgiu com o apoio de mais de 130 contribuidores. Não se sabe quantos utilizadores existem porque tem de ser alojado por nós, localmente ou num servidor cujo somos proprietários, o que faz com que seja o que menor comunidade tem e por consequência um dos menos utilizados.

Fornece uma interface baseada no *Github* assim como recursos iguais ou semelhantes.

Além de tudo isto é totalmente gratuito.

Glossário

- *String* – conjunto de caracteres;
- *Scroll* – rolar a página para cima ou para baixo;

Bibliografia

Para a realização deste relatório foram consultados os seguintes *websites*:

- *Example requirements specification*
<https://www.slideshare.net/indrisrozas/example-requirements-specification>
- *Software Correctness*
[https://en.wikipedia.org/wiki/Correctness_\(computer_science\)](https://en.wikipedia.org/wiki/Correctness_(computer_science))
- *Software Reliability*
<https://pt.slideshare.net/AnandKumar87/software-reliability-11841804>
- *Software Efficiency*
https://www.axia-consulting.co.uk/html/importance_software_efficiency.html
- *Software Integrity*
<https://www.techopedia.com/definition/10284/integrity>
- *Software Usability*
<https://blog.rocketsoftware.com/2014/11/software-usability/#.W7tmfmhKiUk>
- *Software Maintainability*
<https://software.ac.uk/resources/guides/developing-maintainable-software>
- *Software Flexibility*
<https://informativearchitecture.wordpress.com/2011/10/04/software-flexibility/>
- *Software Portability*
https://en.wikipedia.org/wiki/Software_portability
- *Software Robustness*
[https://en.wikipedia.org/wiki/Robustness_\(computer_science\)](https://en.wikipedia.org/wiki/Robustness_(computer_science))
- *Software Security*
<https://www.techopedia.com/definition/24866/software-security>
- *Software Potential Risks*
<https://www.softwaretestinghelp.com/types-of-risks-in-software-projects/>
- *Software Assumptions*
<https://projectbliss.net/project-assumptions-list/>
- *Git*
<https://git-scm.com/>
- *Git vs Github vs Gitbucket*
<https://stackshare.io/stackups/git-vs-gitbucket-vs-github>