

Cinema

Estudantes:

- + André Costa - up201905916
- + David Ferreira - up202006302
- + Guilherme Carvalho - up202005369

Professores:

- + Michel Celestino Paiva Ferreira
- + Carla Alexandra Teixeira Lopes
- + Lázaro Gabriel Barros da Costa

Tema do trabalho:

- + Cinema

Índice

1. Capa	01
2. Índice	02
3. Descrição	03
4. Diagrama Conceptual - UML	04
5. Diagrama Conceptual - UML revisto	05
6. Modelo Relacional	06
7. Dependências Funcionais e Formas Normais	07-08
8. Restrições	09-11
9. Queries	12
10. Triggers	13
11. Percentagens na Autoavaliação	14

Descrição

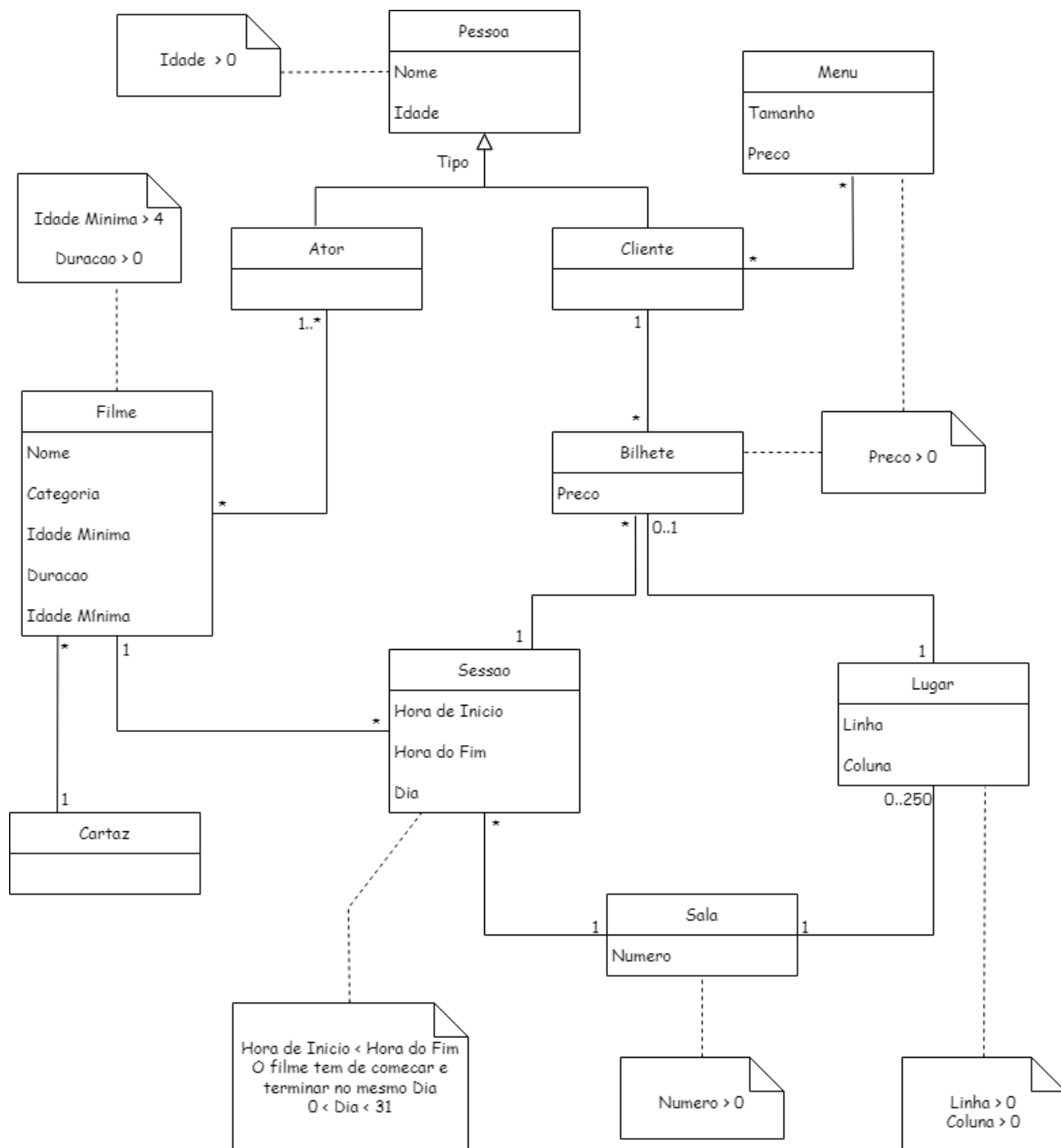
Pretende-se guardar informação relativa ao funcionamento dos Cinema UCI. O Cinema tem a seu cargo a manutenção do cartaz, com a CapaUCI e um IdCartaz, os filmes disponíveis e a respetiva informação de cada filme, e os clientes.

De cada cliente interessa saber que filme gostariam de ver e a sessão correspondente (IdSessao, hora-início, hora-final e o dia). Estes ainda têm à sua disposição um menu consumível, que podem ter diversos tamanhos, variando assim os seus preços.

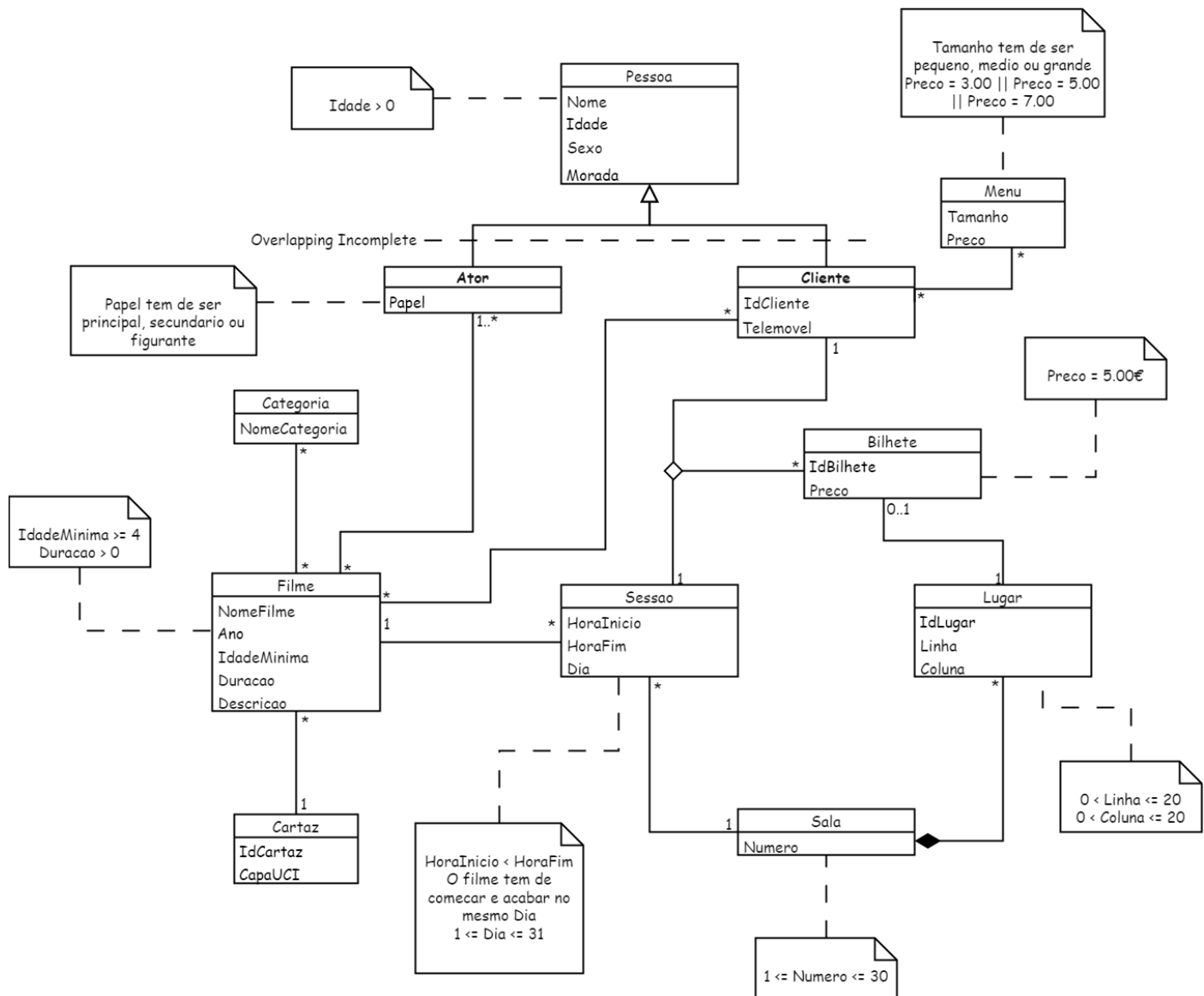
Para ajudar os clientes na escolha dos filmes, estes para além do nome e da sua descrição, têm ainda as várias categorias, sessões disponíveis (com o respetivo número da sala atribuída), a duração, a idade mínima e os atores envolvidos. Os atores e os clientes, são ambos pessoas, descritos pelo nome, idade, sexo e morada. Dos atores ainda se sabe se fazem o papel principal, secundário ou de figurantes.

Para o cliente assistir a um filme, tem de comprar o bilhete com um IdBilhete e um preço associado, cujo tem a informação do lugar (IdLugar, linha, coluna), do número da sala, o IdCliente e da sessão correspondente. Tem que deixar também o seu número de telemóvel, para caso seja necessário ser contactado.

Modelo Conceptual - UML



Modelo Conceptual – UML revisto



Modelo Relacional

- Cartaz(IdCartaz, CapaUCI)
- Filme(NomeFilme, Ano, IdadeMinima, Duracao, Descricao, IdCartaz→Cartaz)
- Categoria(NomeCategoria)
- Sala(Numero)
- Sessao(HoraInicio, HoraFim, Dia, Numero→Sala, NomeFilme→Filme, Ano→Filme)
- Lugar(IdLugar, Linha, Coluna, Numero→Sala)
- Pessoa(Nome, Idade, Morada, Sexo)
- Ator(Nome→Pessoa, Idade→Pessoa, Morada→Pessoa, Papel)
- Cliente(IdCliente, Nome→Pessoa, Idade→Pessoa, Morada→Pessoa, Telemovel)
- Bilhete(IdBilhete, Preco, {HoraInicio, HoraFim, Dia, Numero}→Sessao, IdLugar→Lugar, IdCliente→Cliente)
- Menu(Tamanho, Preco)

Many-to-Many Associations:

- FilmeCategoria(NomeFilme→Filme, Ano→Filme, NomeCategoria→Categoria)
- FilmeAtor(NomeFilme→Filme, Ano→Filme, {Nome, Idade, Morada}→Ator)
- FilmeCliente(NomeFilme→Filme, Ano→Filme, IdCliente→Cliente)
- ClienteMenu(IdCliente→Cliente, Tamanho→Menu)

Dependências Funcionais e Formas Normais

Cartaz(IdCartaz, CapaUCI)

IdCartaz → CapaUCI

Filme(NomeFilme, Ano, IdadeMinima, Duracao, Descricao, IdCartaz → Cartaz)

{NomeFilme, Ano} → {IdadeMinima, Descricao, Duracao, Id}

Sessao(HoraInicio, HoraFim, Dia, Numero → Sala, NomeFilme → Filme, Ano → Filme)

{HoraInicio, HoraFim, Dia, Numero} → {NomeFilme, Ano}

Lugar(IdLugar, Linha, Coluna, Numero → Sala)

IdLugar → {Linha, Coluna, Numero}

Pessoa(Nome, Idade, Morada, Sexo)

{Nome, Idade, Morada} → Sexo

Ator(Nome → Pessoa, Idade → Pessoa, Morada → Pessoa, Papel)

{Nome, Idade, Morada} → Papel

Cliente(IdCliente, Nome → Pessoa, Idade → Pessoa, Morada → Pessoa, Telemovel)

IdCliente → {Nome, Idade, Morada, Telemovel}

Bilhete(IdBilhete, Preço, {HoraInicio, HoraFim, Dia, Numero}->Sessao, IdLugar->Lugar, IdCliente->Cliente)

IdBilhete->{Preço, {HoraInicio, HoraFim, Dia, Numero}, IdLugar, IdCliente}

Menu(Tamanho, Preço)

Tamanho->Preço

FilmeCategoria(NomeFilme->Filme, Ano->Filme, NomeCategoria->Categoria)

Não tem;

FilmeAtor(NomeFilme->Filme, Ano->Filme, {Nome, Idade, Morada}->Ator)

Não tem;

FilmeCliente(NomeFilme->Filme, Ano->Filme, IdCliente->Cliente)

Não tem;

ClienteMenu(IdCliente->Cliente, Tamanho->Menu)

Não tem;

As nossas relações estão todas na BCNF, pois ou as dependências funcionais são triviais, apesar de estas não estarem representadas, tal como Numero->Numero da relação Sala, ou os atributos do lado esquerdo são chaves/superchaves.

As nossas relações estão todas na 3NF, pois para cada dependência funcional não-trivial, os atributos do lado esquerdo são chaves/superchaves ou os atributos do lado direito são todos primos.

Restrições

Cartaz(IdCartaz, CapaUCI)

O Id auto-incrementa-se.

IdCartaz NOT NULL

PRIMARY KEY(IdCartaz)

CapaUCI DEFAULT 'UCI'

Filme(NomeFilme, Ano, IdadeMinima, Duracao, Descricao, IdCartaz->Cartaz)

Uma pessoa para ir assistir a um filme tem de ter no mínimo 4 anos, apesar que pode variar de filme para filme.

Um filme tem de ter mais do que 0 mins;

PRIMARY KEY(NomeFilme, Ano)

NomeFilme NOT NULL

Ano NOT NULL

CHECK(Duracao > 0)

Duracao NOT NULL

Descricao NOT NULL

CHECK(IdadeMinima >= 4)

IdCartaz NOT NULL

IdadeMinima DEFAULT 4

Categoria(NomeCategoria)

PRIMARY KEY(NomeCategoria)

NomeCategoria NOT NULL

Sala(Numero)

Só existem 30 salas no cinema numeradas a começar no número 1.

PRIMARY KEY(Numero)

Numero NOT NULL

CHECK(Numero >= 1 AND Numero <= 30)

Sessao(HoraInicio, HoraFim, Dia, Numero->Sala, NomeFilme->Filme, Ano->Filme)

A hora do fim de um filme é sempre maior do que a hora do inicio.

O filme tem de começar e acabar no mesmo dia. E o dia tem de pertencer aos dias do mês.

(Generalizamos os meses para meses com 31 dias)

PRIMARY KEY(HoraInicio, HoraFim, Dia, Numero)

CHECK(HoraInicio < HoraFim)

HoraInicio NOT NULL

HoraFim NOT NULL

CHECK(HoraFim <= '23:59:59')

Numero NOT NULL

NomeFilme NOT NULL

CHECK(Dia > 0 AND Dia <= 31)

Ano NOT NULL

Lugar(IdLugar, Linha, Coluna, Numero->Sala)

O Id auto incrementa-se. O número das linhas e das colunas estão numerados de 0 a 20.

PRIMARY KEY(IdLugar)

CHECK(Linha > 0 AND Linha <= 20)

IdLugar NOT NULL

Linha NOT NULL

CHECK(Coluna > 0 AND Coluna <= 20)

Coluna NOT NULL

Numero NOT NULL

Pessoa(Nome, Idade, Morada, Sexo)

Qualquer pessoa tem mais do que 0 anos.

PRIMARY KEY(Nome, Idade, Morada)

CHECK(Idade > 0)

Nome NOT NULL

Idade NOT NULL

CHECK(Sexo = 'M' OR Sexo = 'F')

Morada NOT NULL

Sexo NOT NULL

Ator(Nome->Pessoa, Idade->Pessoa, Morada->Pessoa, Papel)

Um ator ou tem papel principal, papel secundário ou faz de figurante.

Nome NOT NULL

PRIMARY KEY(Nome, Idade, Morada)

Papel DEFAULT 'Figurante'

Idade NOT NULL

CHECK(Papel = 'Papel Principal' OR Papel = 'Papel Secundario' OR Papel = 'Figurante') Morada NOT NULL

Cliente(IdCliente, Nome->Pessoa, Idade->Pessoa, Morada->Pessoa, Telemovel)

O Id auto incrementa-se. O número de telemóvel é único, não existem 2 iguais e têm todos 9 números.

PRIMARY KEY(IdCliente) IdCliente NOT NULL Nome NOT NULL

UNIQUE(Telemovel) Idade NOT NULL Morada NOT NULL Telemovel NOT NULL

Bilhete(IdBilhete, Preco, {HoraInicio, HoraFim, Dia, Numero}-> Sessao, IdLugar->Lugar, IdCliente->Cliente)

O Id auto incrementa-se. Os bilhetes não são grátis. IdBilhete NOT NULL IdLugar NOT NULL

PRIMARY KEY(IdBilhete) HoraInicio NOT NULL HoraFim NOT NULL Numero NOT NULL

CHECK(Preco = 5.00) Preco DEFAULT 5.00 IdCliente NOT NULL

Menu(Tamanho, Preco)

Um cliente tem acesso a menus com 3 tamanhos distintos, pequeno, médio ou grande.

E os menus não são grátis.

PRIMARY KEY(Tamanho) Tamanho DEFAULT 'Pequeno' Preco DEFAULT 3.00

CHECK((Preco = 3.00 AND Tamanho = 'Pequeno') OR (Preco = 5.00 AND Tamanho = 'Medio')

OR (Preco = 7.00 AND Tamanho = 'Grande'))

FilmeCategoria(NomeFilme->Filme, Ano->Filme, NomeCategoria->Categoria)

PRIMARY KEY(NomeFilme, Ano, NomeCategoria) NOT NULL(NomeFilme, Ano, NomeCategoria)

FilmeAtor(NomeFilme->Filme, Ano->Filme, {Nome, Idade, Morada}->Ator)

PRIMARY KEY(NomeFilme, Ano, {Nome, Idade, Morada}) NOT NULL(NomeFilme, Ano, Nome, Idade, Morada)

FilmeCliente(NomeFilme->Filme, Ano->Filme, IdCliente->Cliente)

PRIMARY KEY(NomeFilme, Ano, IdCliente) NOT NULL(NomeFilme, Ano, IdCliente)

ClienteMenu(IdCliente->Cliente, Tamanho->Menu)

PRIMARY KEY(IdCliente, Tamanho) IdCliente NOT NULL

Queries

- I. O Tamanho do Menu e o Nome e a Idade e de todos os clientes cujo nome começa por “Ma” e que tenham comprado um Menu de Tamanho médio.
- II. Qual o Nome, a idade e o Telemovel do cliente mais velho, de todos os clientes que foram ver o filme “EndGame” na última Sessao no dia 14.
- III. Todos os Filmes lançados no ano 2017 e todos os Ators que participaram em cada Filme.
- IV. Um dos Dias em que houve mais Clientes numa sala e quantos foram.
- V. A NomeFilme e o Ano do Filme mais recente e do mais antigo disponíveis no cinema.
- VI. O Identificador e o nome dos Clientes que assistiram tanto ao Filme “EndGame” como ao “Spider-Man: NO WAY HOME”.
- VII. Quanto dinheiro gastou cada Cliente a comprar Bilhetes e Menus (PrecoTotal).
//Deduz-se que os Clientes compram sempre o mesmo menu quando vão ao cinema
- VIII. Todos os pares de filmes que partilhem uma Categoria, ou seja, pertencem ambos a essa Categoria.
- IX. Toda a informação dos Lugares ocupados na Sala que teve mais Clientes no Dia 14.
- X. Quantas Categorias tem cada filme, mesmo que não tenha nenhuma.

Triggers

1) `ChangeIdCartaz`

O objetivo deste Trigger é verificar se o Filme já se enquadra no grupo dos Filmes "antigos" para ser transferido para o Cartaz com a `CapaUCI = 'UCI OLD'`. Para isso é feito um "before update of `IdCartaz` on Filme" com a condição de o Ano ser maior do que '2016-12-31'. Se for, passa a mensagem de erro "O Filme ainda é recente".

2) `ConfirmNumeroSessoes`

Este Trigger verifica se o número de Sessões por Dia de um Filme atinge o valor máximo, ou seja, por Dia só é permitido, para cada Filme, ter 3 Sessões (não interessando a hora destas). Portanto, criamos uma View que conta o número de Sessões por Dia para cada Filme e no Trigger é verificado a condição referida através de um "before insert on". Não passando, deixa uma mensagem de erro que diz "Chegou ao número máximo de Sessões nesse dia."

3) `ConfirmLugarBilhete`

Este Trigger não permite que um Cliente, ao comprar um Bilhete, não calhe no mesmo Lugar que outro Cliente, na mesma Sessão, na mesma Sala, no mesmo Dia. Isso é testado através de um "after insert on" e de um "rollback" caso isso aconteça, ou seja, volta atrás o insert feito na tabela Bilhete, dando a mensagem "Lugar Ocupado."

Percentagens na Autoavaliação

- Opinião do André Costa:

	1ª Entrega	2ª entrega	3ª entrega
André Costa	50%	95%	70%
David Ferreira	25%	5%	22,5%
Guilherme Carvalho	25%	0%	7,5%

- Opinião do David Ferreira:

	1ª Entrega	2ª entrega	3ª entrega
André Costa	33%	70%	60%
David Ferreira	33%	25%	35%
Guilherme Carvalho	33%	5%	5%

- Opinião do Guilherme Carvalho:

	1ª Entrega	2ª entrega	3ª entrega
André Costa	33%	60%	70%
David Ferreira	33%	25%	22,5%
Guilherme Carvalho	33%	15%	7,5%