

# Gestão de uma Loja de Aplicações (*App Store*)



*Relatório Final da Base de Dados*

**Grupo 17 / P4**

Ana Simão 42978

Francisco Magalhães 42745

Pedro Vieira 42610

# Índice

1. Tema e objectivos da Base de Dados .....	3
2. Modelo de Dados .....	6
2.1. Alterações Adotadas em Relação à Versão Inicial .....	6
2.2. Discussão das Opções Tomadas .....	7
2.3. Modelo ER .....	9
2.4. Passagem do Modelo ER para o Modelo Relacional .....	10
2.5. Modelo Relacional .....	12
2.6. Limitações/opções tomadas para a implementação da BD .....	13
2.7. Criação da Base de Dados .....	17
2.7.1. Criação das Tabelas .....	17
2.7.2. Criação de Sequências .....	17
2.7.3. Criação de Funções .....	22
2.7.4. Criação de Vistas .....	26
2.7.5. Criação de Triggers .....	27
2.8. Descrição da Interface .....	34
3. Manual de Utilizador .....	37
4. Considerações Finais .....	62

# 1. Tema e objectivos da Base de Dados

O tema do nosso trabalho é a **Gestão de uma loja de Aplicações** (quer sejam para dispositivos móveis ou não), geralmente denominada como **App Store**. O objetivo deste sistema é gerir os **utilizadores** e **programadores** da loja, bem como as **aplicações** submetidas, que os utilizadores podem **transferir** e **avaliar**.

Os **utilizadores** poderão guardar, na nossa base de dados, informações como: o seu nome, email, número de telefone, cartão de crédito e o seu país (sendo que esta informação é obtida de uma tabela exterior ao dos utilizadores, os **países**, que contém todos os nomes e códigos dos países permitidos na *App Store*).

Para além disto, cada **utilizador** é identificado unicamente por um número gerado automaticamente pelo próprio sistema aquando o seu registo.

Todos os dados são obrigatórios no registo, excetuando o número de telefone e o cartão de crédito (permitindo a possibilidade de fazer downloads de aplicações gratuitas sem a introdução de um cartão de crédito na base de dados).

Os **utilizadores** registados no nosso sistema poderão tornar-se **programadores** (*developers*) sendo que terão todos os dados que os utilizadores "normais", como também o código da licença de programador e o seu website pessoal.

As contas de programador poderão ser usadas por indivíduos ou por empresas. Os *developers* poderão acrescentar **aplicações** (quer sejam gratuitas ou pagas) no sistema para serem transferidas por todos utilizadores elegíveis.

Por outro lado, as contas de **programadores** poderão especializar-se em **administradores**, sendo que este tipo é atribuído unicamente a programadores que sejam trabalhadores da própria empresa da App Store, recebendo assim um estatuto especial relativamente a outros programadores "regulares".

Este estatuto poderá facilitar a publicação de novas aplicações, tendo estas uma maior prioridade de entrada na loja do que todas as outras. Por isso, esta conta terá um novo dado que é o **número do empregado**.

As **aplicações** presentes no sistema terão dados como: o nome, a classificação, o website da aplicação, a descrição, o SKU (este atributo é uma cadeia de caracteres única que é utilizada para identificar uma aplicação, por ex., "BD\_APP\_PROJETO"), o nome e a versão mínima da **plataforma** que o utilizador necessita de ter para correr a aplicação, e o nome da categoria (sendo que a entidade **categoria** também tem um outro atributo para além deste, a descrição da categoria). Para além disto, elas próprias também terão acesso a informações relativas às versões da aplicação, incluindo as versões antigas e a atual, bem como a mensagem do update e a data de lançamento delas.

Com os dados da plataforma mínima necessária para correr uma determinada aplicação, podemos utilizá-la para mostrar uma mensagem informativa (num website ou até mesmo na aplicação da *App Store*) do género: "*Necessita de ter pelo menos o **Android 4.0** para correr a aplicação no seu dispositivo*".

No nosso sistema consideramos que uma aplicação apenas é suportada por um único Sistema Operativo (por exemplo, Android) e se o programador desejar colocar uma nova, nem que seja igual, para suportar outro S.O. (como o iOS), necessita de adicioná-la com um novo SKU, ou seja, em termos práticos, uma aplicação totalmente nova.

Cada aplicação terá associado um **tier**, que será identificado por um número único gerado pelo sistema, e guardará a informação referente ao preço base da aplicação, em dólares. A utilização dos tiers impossibilita que cada programador escolha um determinado preço "aleatório" para a venda das suas *apps*. Por isso, todas as aplicações que utilizem, por exemplo, o *tier 1* terão todas o mesmo preço (que poderá ser convertido para as diferentes **moedas**).

De seguida encontram-se alguns exemplos de *tiers* que são utilizados em sistemas de *App Store* atuais:

- **Tier 0** —> \$0 (grátis)
- **Tier 1** —> \$0,99
- **Tier 2** —> \$1,99
- e assim sucessivamente...

No nosso sistema guardamos também os dados referentes à **moeda** que cada país possui, dentro desses dados consta: o código da moeda (representado por três caracteres obrigatórios), o próprio símbolo, a taxa de conversão (sempre em relação ao dólar) e o nome.

Por último, os **utilizadores** poderão dar uma classificação (fazer uma *review*), dando uma classificação de **1** a **5**, às versões das aplicações de que fizeram **download** ou **update**, podendo inclusivamente deixar um título e um comentário de forma a explicar o porquê de atribuir tal classificação à versão da *app*. As **reviews** para além destes dados, também guardarão a data da submissão.

## 2. Modelo de Dados

### 2.1. Alterações Adotadas em Relação à Versão Inicial

Em relação ao diagrama de relações, alterámos a participação do conjunto de entidades applications, de total para parcial, na relação app\_versions do lado das aplicações.

Alterámos a cardinalidade da relação review\_download de muitos (nas **reviews**) para um (nos **downloads**) para de **um para um** pois faz sentido que apenas uma review diga respeito ao download de uma aplicação por parte de um user. Esta alteração reflete-se também na simplificação das tabelas (na passagem do modelo ER para o modelo relacional). Adicionámos um novo atributo, cat\_description ao conjunto de entidades **categories** de forma a permitir adicionar um pequeno texto descritivo sobre a própria categoria.

Eliminámos o atributo **app\_rating** do conjunto de entidades applications pois concluímos que seria mais eficiente tornar este atributo como um atributo calculado, que será apresentado mais à frente.

No código da criação da Base de Dados, em várias chaves externas eliminámos a linha de código “ON DELETE CASCADE” para garantir que entidades que estejam dependentes do conjunto de entidades referido na chave externa não sejam eliminadas.

Alterámos também o tamanho total (do *varchar*) em alguns atributos de tabelas (como os websites da app e do developer q passaram a ser *varchar*(255), app\_description e release\_notes (que são agora *varchar*(1000)), o email do utilizador (que também passou agora a *varchar*(255), e que agora utiliza uma restrição REGEX que garante que o texto inserido é realmente um email), a title da review passou agora de *varchar*(30) para *varchar*(50)). Muitas destas alterações foram feitas para refletir as necessidades do mundo real.

## 2.2. Discussão das Opções Tomadas

Como descrito anteriormente a nossa base de dados, a *App Store*, modela essencialmente aplicações, utilizadores e as interações entre ambas (como os downloads e as reviews).

Para representar as aplicações utilizámos cinco atributos (o nome, a descrição, SKU, a classificação e o website da aplicação) que nos permitem representar as informações básicas de cada aplicação de forma eficaz.

Para as categorias, as plataformas e os tiers das aplicações decidimos representá-las através de um conjunto de entidades e não como atributos das próprias aplicações, de forma a evitar a repetição de dados.

O conjunto de entidades **applications** tem participação total nas relações com a categoria, com a plataforma e com o tier, pois estas são características obrigatórias nas aplicações.

Cada aplicação possui apenas uma categoria, uma plataforma mínima para utilização e um tier.

Para representar as várias versões que uma aplicação pode ter, optámos por criar o conjunto de entidades fracas **versions**, que guarda as informações referentes à própria versão, a data e as notas de lançamento da versão. As versões são um conjunto de entidades fraco pois só é possível identificá-las e distingui-las se estas estiverem relacionadas com uma determinada aplicação.

Em relação ao conjunto de entidades **tiers**, a sua informação é representada por dois atributos, o número do tier e o preço base do mesmo, em dólares. Tendo em conta que pretendemos simular uma base de dados coerente com o mundo real, sentimos necessidade de representar as diferentes moedas nas quais as transações de aplicações podem ser feitas. Como tal, e novamente para evitar repetições de dados, criámos o conjunto de entidades **currencies**, que guarda várias informações relativamente às moedas dos diferentes países, como o código, o símbolo, a taxa de conversão (em relação ao dólar) e o nome da moeda.

O conjunto de entidades **tiers** tem uma participação total na relação de muitos para muitos com as moedas, pois cada tier é obrigado a ter pelo menos uma moeda associada.

Para representar os utilizadores criámos o conjunto de entidades **users** que tem com atributos o id, o nome, o email, o número de telefone e o número do cartão de crédito do utilizador. Para representar a informação referente ao país associado a

cada conta de utilizador, decidimos criar o conjunto de entidades **countries** que contém o código do país e o seu nome. Como cada país tem a sua moeda oficial, na qual as transações dos utilizadores desse país se realizam, o conjunto de entidades **countries** tem uma relação com o conjunto de entidades **currencies**. A participação dos países é total e a relação é de muitos para um, pois é obrigatório um país ter apenas uma moeda associada.

Desta forma, os atributos referidos anteriormente, assim como a relação com uma moeda, são comuns a todos os tipos de **users**, sejam eles utilizadores comuns ou programadores de aplicações. Este subgrupo de utilizadores, os **developers**, são uma especialização parcial dos **users** (parcial pois podem existir utilizadores da *App Store* que não são programadores) que têm como atributos adicionais o código da licença de programador e o seu website pessoal.

Por sua vez, os **developers** possuem uma especialização parcial em administradores, os **administrators**, que são basicamente trabalhadores da própria empresa da *App Store*, e que possuem um código de empregado.

É fundamental não esquecer as relações que podem ocorrer entre as aplicações e os utilizadores, pois são elas que estão na base do funcionamento de uma *App Store*.

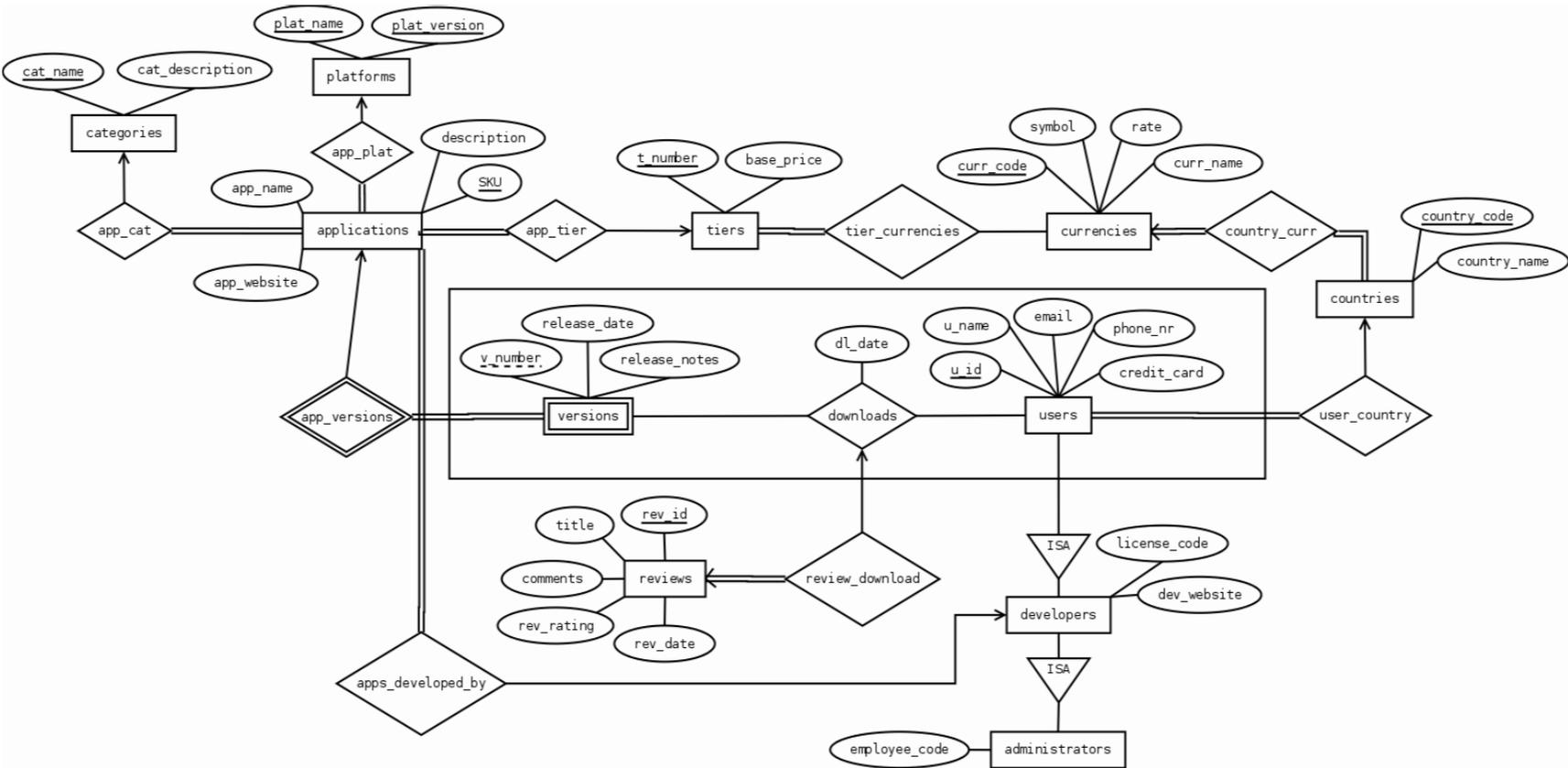
No nosso sistema, cada aplicação é obrigatoriamente desenvolvida por um, e um só, programador (**developer**).

Em relação à obtenção de uma aplicação por parte de um utilizador, decidimos criar uma relação, **download**, entre o conjunto de entidades **users** e o conjunto de entidades fracas **versions** (pois através desta temos acesso à aplicação pretendida). Esta relação possui um atributo próprio que corresponde à data inicial do download.

Optámos por criar uma agregação entre as versões e os utilizadores, que se relaciona com as reviews, pois cada **review** diz respeito a uma versão de uma determinada aplicação por parte de um utilizador específico.

O conjunto de entidades **reviews** tem cinco atributos: o id, o título, o comentário, a classificação atribuída nessa review e a data em que a review foi feita.

2.3. Modelo ER



## 2.4. Passagem do Modelo ER para o Modelo Relacional

### Versão 1

#### Entidades

categories ( { cat\_name, cat\_description } )  
platforms ( { plat\_name, plat\_version } )  
tiers ( { t\_number, base\_price } )  
countries ( { country\_code, country\_name } )  
users ( { u\_id, u\_name, email, phone\_nr, credit\_car } )  
developers ( { u\_d, license\_code, dev\_website } )  
administrators ( { u\_id, employee\_code } )  
applications ( { SKU, app\_rating, app\_name, description, app\_website } )  
versions ( { v\_number, SKU, release\_date, release\_notes } )  
downloads ( { v\_number, SKU, u\_id, dl\_date } )  
reviews ( { rev\_id, title, comments, rev\_rating, rev\_date } )  
currencies ( { curr\_code, symbol, rate, curr\_name } )

#### Relações

app\_cat ( { SKU, name } )  
app\_plat ( { SKU, plat\_name, plat\_version } )  
app\_tier ( { SKU, t\_number } )  
app\_versions ( { v\_number, SKU } )  
app\_developed\_by ( { SKU, license\_code, u\_id } )  
user\_country ( { u\_id, country\_code } )  
review\_download ( { rev\_id, v\_number, SKU, u\_id } )  
country\_curr ( { country\_code, curr\_code } )  
tiers\_currencies ( { t\_number, curr\_code } )

## Versão 2

Nesta versão eliminámos as tabelas simplificáveis usando chaves-externas, que se encontram a negrito.

### Entidades

categories ({ cat\_name, cat\_description })

platforms ({ plat\_name, plat\_version })

tiers ({ t\_number, base\_price })

countries ({ country\_code, name, **curr\_code** })

users ({ u\_id, u\_name, email, phone\_nr, credit\_car, **country\_code** })

developers ({ u\_id, license\_code, dev\_website })

administrators ({ u\_id, employee\_code })

applications ({ SKU, app\_rating, app\_name, description, app\_website, **cat\_name**, **plat\_name**, **plat\_version**, **t\_number**, **u\_id** })

versions ({ v\_number, **SKU**, release\_date, release\_notes })

downloads ({ v\_number, **SKU**, u\_id, dl\_date })

reviews ({ rev\_id, title, comments, rev\_rating, rev\_date })

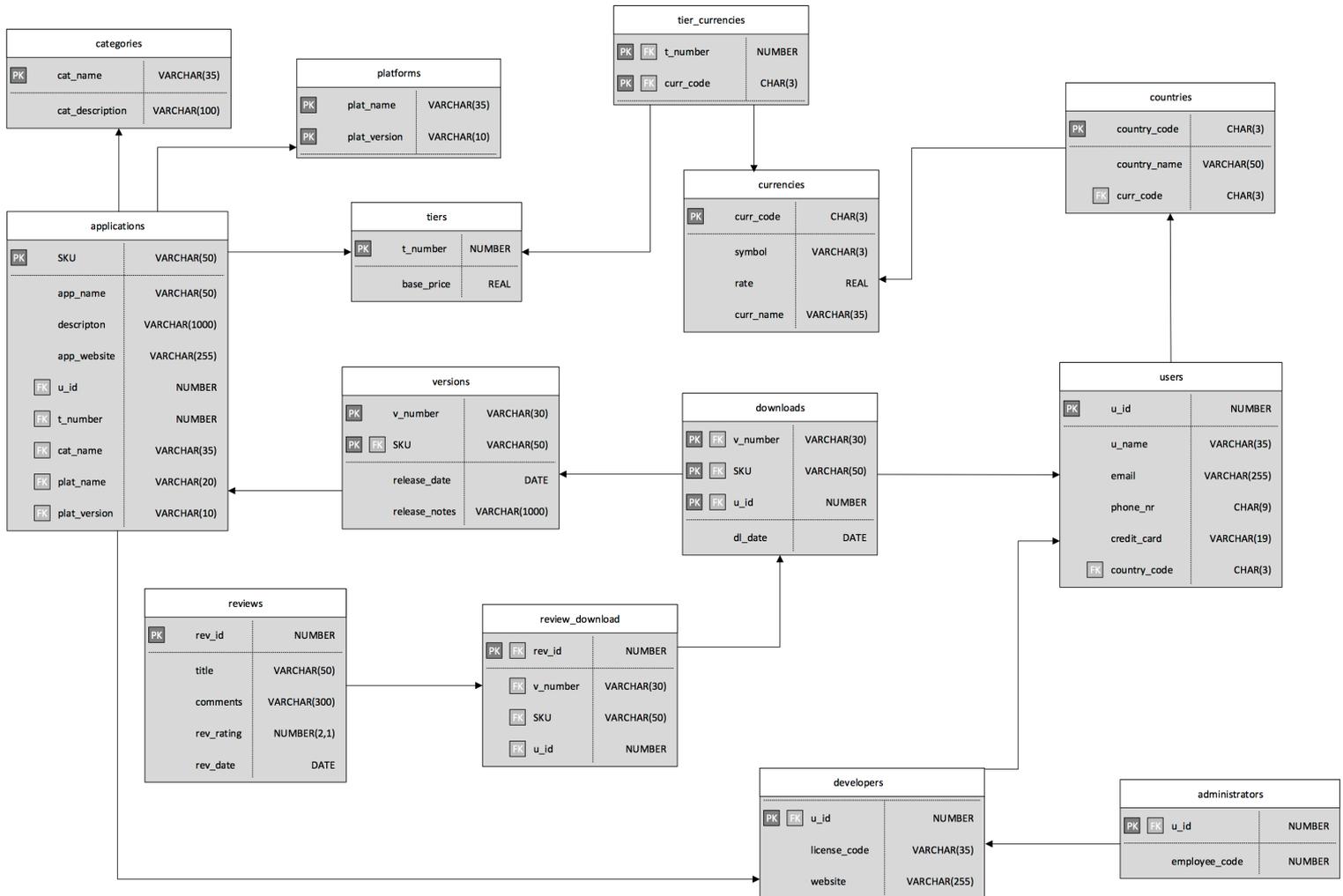
currencies ({ curr\_code, symbol, rate, curr\_name })

### Relações

tier\_currencies ({ t\_number, curr\_code })

reviews\_downloads ({ rev\_id, v\_number, **SKU**, u\_id })

## 2.5. Modelo Relacional



## 2.6. Limitações/opções tomadas para a implementação da BD

Foi necessário criar algumas restrições para cada uma das tabelas representadas na implementação da Base de Dados para garantir o seu bom funcionamento e integridade.

Na tabela **categories** declaramos os atributos *cat\_name* e *cat\_description* como **NOT NULL**, pois são campos obrigatórios na definição de uma categoria de aplicações. A chave primária é *cat\_name*.

A tabela **platforms** contém os atributos *plat\_version* e *plat\_name* ambos declarados como **NOT NULL**, pois são campos obrigatórios na definição de uma plataforma de aplicações. O atributo *plat\_verrsion* possui uma restrição **REGEX** para garantir que é inserida uma plataforma de aplicação com uma estrutura válida (por exemplo: 1.1.0). A chave primária é *plat\_version* e *plat\_name*.

Em relação à tabela **tiers**, são declarados os atributos *t\_number* e *base\_price* como **NOT NULL**, pois são campos obrigatórios na definição de uma tier para aplicações. É garantido que o *base\_price* tem sempre um valor superior a 0. A chave primária é *t\_number*.

Na tabela **currencies** todos os atributos(*curr\_code*, *symbol*, *rate* e *curr\_name*) são declarados com **NOT NULL**, pois são campos obrigatórios na definição de uma moeda de um país. É também garantido que o *rate* tem sempre um valor maior que 0, e que não podem existir moedas com o mesmo nome, declarando o atributo *curr\_name* como **UNIQUE**. A chave primária é *curr\_code*.

Na tabela **tier\_currencies** ambos os atributos, *t\_number* e *curr\_code*, são declarados como **NOT NULL**, pois são campos obrigatórios na definição de uma relação entre uma moeda e uma tier. É garantido que o número da tier é sempre superior a 0. Os atributos *t\_number* e *curr\_code* são chaves externas e são também a chave primária.

A tabela **countries** possui todos os atributos(*country\_code*, *country\_name* e *curr\_code*) declarados como **NOT NULL**, pois são campos obrigatórios na definição de um país. É garantido que não existem países, através de uma restrição de

unicidade, **UNIQUE**. O atributo *curr\_code* é uma chave externa e *country\_code* é chave primária

Em relação à tabela **users** todos os atributos(*u\_id*, *u\_name*, *email*, *country\_code*) exceto *credit\_card* e *phone\_nr* são **NOT NULL**, pois são campos obrigatórios na definição de um utilizador. É garantido que o número do cartão de crédito tem sempre 16 ou mais dígitos, este atributo não é declarado como **UNIQUE**, pois considerámos que várias pessoas podem partilhar o mesmo cartão de crédito(uma família por exemplo) e não é declarado como not null pois os utilizadores podem fazer downloads de aplicações grátis sem a necessidade de inserir um número de cartão de crédito. O email é também definido como único(**UNIQUE**) para cada utilizador e possui uma restrição **REGEX** para garantir que a estrutura do email inserido é válida. O atributo *country\_code* é chave externa e *u\_id* é chave primária.

A tabela **developers** possui os atributos *u\_id*, *license\_code* e *website*, dos quais o *u\_id* e *license\_code* estão definidos como **NOT NULL**, pois são campos obrigatórios na definição de um programador. É garantido que o código de licença é único para cada programador(está definido como **UNIQUE**) e que o website tem uma estrutura válida, através de um restrição **REGEX**. O atributo *u\_id* é chave externa e é também chave primária.

Na tabela **administrators** os atributos *u\_id* e *employee\_code* são definidos como **NOT NULL**, pois são campos obrigatórios na definição de um administrador. O código de empregado é definido como **UNIQUE** para cada administrador, para garantir que não existem dois empregados com o mesmo código. O atributo *u\_id* é chave externa e é também chave primária.

A tabela **applications** possui os atributos *SKU*, *app\_name*, *description*, *u\_id*, *t\_number*, *cat\_name*, *plat\_name* e *plat\_version* declarados como **NOT NULL**, pois são campos obrigatórios na definição de uma aplicação. O website da aplicação possui uma restrição **REGEX** para garantir que a estrutura do website inserido é válida. É garantido que o nome da aplicação é único pois o atributo *app\_name* está definido como **UNIQUE**. Os atributos *cat\_name*, *plat\_name*, *plat\_version*, *t\_number* e *u\_id* são chave externa e *SKU* é chave primária.

A tabela **versions** contém os atributos *v\_number*, *SKU*, *release\_date* e *release\_notes* todos declarados como **NOT NULL**, pois são campos obrigatórios na definição de uma versão de aplicação. O atributo *v\_number* possui uma restrição **REGEX** para garantir que é inserida uma versão de aplicação com uma estrutura válida (por exemplo: 1.1.0 ou 1.1). O atributo *SKU* é chave externa. A chave primária é *SKU* e *v\_number*.

Na tabela **downloads** declaramos os atributos *v\_number*, *SKU*, *u\_id* e *dl\_date* como **NOT NULL**, pois são campos obrigatórios na definição de um download. Os atributos *v\_number*, *SKU* e *u\_id* são chaves externas e são também chave primária. Para além disto, consideramos que um utilizador não fará download da mesma versão de uma aplicação que transferiu anteriormente: achamos que este caso seria processado diretamente por uma interface web, ou até mesmo aplicação, da própria *App Store* e que, caso isto acontecesse, simplesmente o deixaria transferir a aplicação sem acrescentar/alterar nada na base de dados.

A tabela **reviews** possui os atributos *rev\_id*, *rev\_rating* e *rev\_date* são declarados com **NOT NULL**, pois são campos obrigatórios na definição de um download. É também garantido que o *rev\_rating* tem sempre um valor entre 1 e 5. A chave primária é *rev\_id*.

Na tabela **review\_download** os atributos *rev\_id*, *v\_number*, *SKU* e *u\_id* definidos como **NOT NULL**, pois são campos obrigatórios na definição de uma relação entre reviews e downloads. Todos os atributos são definidos como **UNIQUE** para garantir que todos os atributos sejam únicos. Todos os atributos desta relação são chaves externas. O atributo *rev\_id* é chave primária.

Foram também criadas **views** para efectuarmos determinadas consultas, nomeadamente:

- **v\_new\_apps** que mostra todas as aplicações inseridas no dia da pesquisa.
  
- **v\_reviews** que permite ver todas as reviews feitas por utilizadores a versões de aplicações
  
- **v\_all\_apps** que mostra todas as aplicações na Base de Dados.

Desta forma é garantido um certo nível de abstracção relativamente ao sistema da base de dados aos utilizadores, garantindo a segurança dos dados.

Uma consideração que achamos importante referir é o facto de no nosso sistema não estarmos a considerar erros por parte do utilizador ao inserir chaves primárias como o nome da plataforma ou o código do país.

## 2.7. Criação da Base de Dados

### 2.7.1. Criação das Tabelas

```
DROP TABLE categories CASCADE CONSTRAINTS;  
CREATE TABLE categories (  
  cat_name varchar(35) NOT NULL,  
  cat_description varchar(100) NOT NULL,  
  PRIMARY KEY (cat_name)  
);
```

```
DROP TABLE platforms CASCADE CONSTRAINTS;  
CREATE TABLE platforms (  
  plat_name varchar(35) NOT NULL,  
  plat_version varchar(10) NOT NULL CHECK (  
    REGEXP_LIKE (  
      plat_version,  
      '^(([0-9])+\.([0-9]))+\.[0-9]+)?$'  
    )  
  ),  
  PRIMARY KEY (plat_name, plat_version)  
);
```

```
DROP TABLE tiers CASCADE CONSTRAINTS;  
CREATE TABLE tiers (  
  t_number NUMBER NOT NULL,  
  base_price real NOT NULL CHECK (base_price >= 0),  
  UNIQUE (base_price),  
  PRIMARY KEY (t_number)  
);
```

```
DROP TABLE currencies CASCADE CONSTRAINTS;  
CREATE TABLE currencies (  
  curr_code char(3) NOT NULL,  
  symbol varchar(3) NOT NULL,  
  rate real NOT NULL CHECK (rate >= 0),  
  curr_name varchar(35) NOT NULL,  
  UNIQUE (curr_name),  
  PRIMARY KEY (curr_code)  
);
```

```
DROP TABLE tier_currencies CASCADE CONSTRAINTS;  
CREATE TABLE tier_currencies (  
  t_number NUMBER NOT NULL CHECK (t_number >= 0),  
  curr_code char(3) NOT NULL,  
  PRIMARY KEY (t_number, curr_code),  
  FOREIGN KEY (t_number) REFERENCES tiers ON DELETE CASCADE,  
  FOREIGN KEY (curr_code) REFERENCES currencies ON DELETE CASCADE  
);
```

```

DROP TABLE countries CASCADE CONSTRAINTS;
CREATE TABLE countries (
  country_code char(3) NOT NULL,
  country_name varchar(50) NOT NULL,
  curr_code char(3) NOT NULL,
  UNIQUE (country_name),
  PRIMARY KEY (country_code),
  FOREIGN KEY (curr_code) REFERENCES currencies
);

```

```

DROP TABLE users CASCADE CONSTRAINTS;
CREATE TABLE users (
  u_id NUMBER NOT NULL,
  u_name varchar(35) NOT NULL,
  email varchar(255) NOT NULL CHECK (
    REGEXP_LIKE (
      email,
      '^([A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4})$'
    )
  ),
  phone_nr char(9),
  credit_card varchar(19) CHECK (LENGTH (credit_card) >= 16),
  country_code char(3) NOT NULL,
  UNIQUE (email),
  PRIMARY KEY (u_id),
  FOREIGN KEY (country_code) REFERENCES countries
);

```

```

DROP TABLE developers CASCADE CONSTRAINTS;
CREATE TABLE developers (
  u_id NUMBER NOT NULL,
  license_code varchar(35) NOT NULL,
  website varchar(255) CHECK (
    REGEXP_LIKE (
      website,
      '^(https?:\W)?(www.)?([a-zA-Z0-9]+).[a-zA-Z0-9]*.[a-z]{3}.?([a-z]+)?$'
    )
  ),
  UNIQUE (license_code),
  PRIMARY KEY (u_id),
  FOREIGN KEY (u_id) REFERENCES users ON DELETE CASCADE
);

```

```

DROP TABLE administrators CASCADE CONSTRAINTS;
CREATE TABLE administrators (
  u_id NUMBER NOT NULL,
  employee_code NUMBER NOT NULL,
  UNIQUE (employee_code),
  PRIMARY KEY (u_id),
  FOREIGN KEY (u_id) REFERENCES developers ON DELETE CASCADE
);

```

```

DROP TABLE applications CASCADE CONSTRAINTS;
CREATE TABLE applications (
  SKU varchar(50) NOT NULL,
  app_name varchar(50) NOT NULL,
  description varchar(1000) NOT NULL,
  app_website varchar(255) CHECK (
    REGEXP_LIKE (
      app_website,
      '^((https?:\W)?(www.)?([a-zA-Z0-9]+).[a-zA-Z0-9]*.[a-z]{3}.?([a-z]+)?$'
    )
  ),
  u_id NUMBER NOT NULL,
  t_number NUMBER NOT NULL,
  cat_name varchar(35) NOT NULL,
  plat_name varchar(20) NOT NULL,
  plat_version varchar(10) NOT NULL,
  UNIQUE (app_name),
  PRIMARY KEY (SKU),
  FOREIGN KEY (cat_name) REFERENCES categories,
  FOREIGN KEY (plat_name, plat_version) REFERENCES platforms,
  FOREIGN KEY (t_number) REFERENCES tiers,
  FOREIGN KEY (u_id) REFERENCES developers ON DELETE CASCADE
);

```

```

DROP TABLE versions CASCADE CONSTRAINTS;
CREATE TABLE versions (
  v_number varchar(30) NOT NULL CHECK (
    REGEXP_LIKE (
      v_number,
      '^(([0-9])+.\.([0-9]+)\.([0-9]+)?$'
    )
  ),
  SKU varchar(50) NOT NULL,
  release_date date NOT NULL,
  release_notes varchar(1000) NOT NULL,
  PRIMARY KEY (v_number, SKU),
  FOREIGN KEY (SKU) REFERENCES applications ON DELETE CASCADE
);

```

```

DROP TABLE downloads CASCADE CONSTRAINTS;
CREATE TABLE downloads (
  v_number varchar(10) NOT NULL,
  SKU varchar(50) NOT NULL,
  u_id NUMBER NOT NULL,
  dl_date date NOT NULL,
  PRIMARY KEY (v_number, SKU, u_id),
  FOREIGN KEY (v_number, SKU) REFERENCES versions ON DELETE CASCADE,
  FOREIGN KEY (u_id) REFERENCES users ON DELETE CASCADE
);

```

```
DROP TABLE reviews CASCADE CONSTRAINTS;  
CREATE TABLE reviews (  
  rev_id int NOT NULL,  
  title varchar(50),  
  comments varchar(300),  
  rev_rating number(2,1) NOT NULL CHECK (rev_rating BETWEEN 1 AND 5),  
  rev_date date NOT NULL,  
  PRIMARY KEY (rev_id)  
);
```

```
DROP TABLE review_download CASCADE CONSTRAINTS;  
CREATE TABLE review_download (  
  rev_id NUMBER NOT NULL,  
  v_number varchar(30) NOT NULL,  
  SKU varchar(50) NOT NULL,  
  u_id NUMBER NOT NULL,  
  PRIMARY KEY (rev_id),  
  FOREIGN KEY (rev_id) REFERENCES reviews ON DELETE CASCADE,  
  FOREIGN KEY (v_number, SKU, u_id) REFERENCES downloads ON DELETE CASCADE  
);
```

## 2.7.2. Criação de Sequências

As seguintes *Sequences* são utilizadas para dar valores únicos e incrementais às Chaves Primárias das tabelas **users**, **reviews** e **tiers** (nos atributos u\_id, rev\_id, t\_number respetivamente):

```
DROP SEQUENCE seq_users;  
CREATE SEQUENCE seq_users INCREMENT BY 1 START WITH 1;
```

```
DROP SEQUENCE seq_reviews;  
CREATE SEQUENCE seq_reviews INCREMENT BY 1 START WITH 1;
```

```
DROP SEQUENCE seq_tiers;  
CREATE SEQUENCE seq_tiers INCREMENT BY 1 START WITH 1;
```

Estas sequências serão depois utilizadas em *Triggers*, que serão detalhados mais à frente.

### 2.7.3. Criação de Funções

Neste trabalho criámos uma função, `f_n_downloads`, que recebe como parâmetro o **SKU** de uma aplicação e retorna o número total de *downloads* que ela recebeu (de todas as versões). Esta função não considera *updates* da aplicação como *downloads* (considera-se um *update* quando o utilizador transfere uma versão de uma aplicação, tendo já previamente baixado uma outra da mesma *app*).

```
CREATE OR REPLACE FUNCTION f_n_downloads(a_sku IN VARCHAR)
RETURN NUMBER IS
  n_down NUMBER;
BEGIN
  SELECT COUNT(DISTINCT downloads.u_id)
  INTO n_down
  FROM downloads
  WHERE a_sku LIKE downloads.sku;

  RETURN n_down;
END;
/
```

Criámos também a função `f_rating` que recebe como parâmetro o **SKU** de uma aplicação e retorna a classificação média (*rating*) da aplicação, considerando tanto *downloads* como *updates*. Esta função é também referente a todas as versões de uma determinada aplicação.

```
CREATE OR REPLACE FUNCTION f_rating(a_sku IN VARCHAR)
RETURN NUMBER IS
  rating NUMBER;
BEGIN
  SELECT AVG(rev_rating)
  INTO rating
  FROM reviews NATURAL INNER JOIN review_download
  WHERE a_sku LIKE SKU;

  RETURN NVL(rating, 0);
END;
/
```

A função `f_version_release_date` recebe como parâmetros o **SKU** de uma aplicação e uma **versão** (por ex, '1.0.0') e retorna a data de lançamento dessa mesma versão, em formato **date**.

```
CREATE OR REPLACE FUNCTION f_version_release_date(a_sku IN VARCHAR,  
a_v_number IN VARCHAR)  
RETURN DATE IS  
  r_date DATE := NULL;  
  myCount NUMBER;  
BEGIN  
  
  SELECT COUNT(1) INTO myCount  
  FROM versions  
  WHERE SKU LIKE a_sku AND v_number LIKE a_v_number;  
  
  IF myCount > 0 THEN  
  
    SELECT release_date INTO r_date  
    FROM versions  
    WHERE SKU LIKE a_sku AND v_number LIKE a_v_number;  
  
  END IF;  
  
  RETURN r_date;  
END;  
/
```

Para além das funções anteriores criámos também a função f\_latest\_version que recebe como parâmetro o **SKU** e devolve a versão mais recente da aplicação que seja identificada por esse **SKU**.

```
CREATE OR REPLACE FUNCTION f_latest_version(a_sku IN VARCHAR)
RETURN VARCHAR IS
v VARCHAR(30) := NULL;
myCount NUMBER;
BEGIN

SELECT COUNT(1) INTO myCount
FROM versions
WHERE SKU LIKE a_sku;

IF myCount > 0 THEN

SELECT v_number INTO v
FROM (SELECT *
      FROM versions
      WHERE SKU LIKE a_sku
      ORDER BY release_date desc)
WHERE ROWNUM = 1;

END IF;

RETURN v;
END;
/
```

Criámos uma função `f_user_latest_version` que recebe como parâmetros o `u_id` de um utilizador e o `SKU` de uma aplicação e devolve a versão mais recente que o utilizador transferiu dessa aplicação.

```

CREATE OR REPLACE FUNCTION f_user_latest_version(a_u_id IN NUMBER, a_sku IN
VARCHAR)
RETURN VARCHAR IS
  v VARCHAR(30) := NULL;
  myCount NUMBER := 0;
BEGIN

  SELECT COUNT(1) INTO myCount
  FROM versions NATURAL INNER JOIN downloads
  WHERE SKU LIKE a_sku AND u_id = a_u_id;

  IF myCount > 0 THEN

    SELECT v_number INTO v
    FROM (SELECT v_number
          FROM versions NATURAL INNER JOIN downloads
          WHERE SKU LIKE a_sku AND u_id = a_u_id
          ORDER BY release_date desc)
    WHERE ROWNUM = 1;

  END IF;

  RETURN v;
END;
/

```

## 2.7.4. Criação de Vistas

A *view v\_new\_apps* é utilizada para mostrar todas as aplicações que foram lançadas (primeira versão) no dia corrente. No *SELECT* utilizamos a *keyword DISTINCT* para garantir que não aparecem aplicações repetidas no resultado, pois existe a possibilidade de uma dada aplicação receber vários updates num único dia.

```
CREATE OR REPLACE VIEW v_new_apps (SKU, app_name, developer) AS
SELECT SKU, app_name, u_id || '-' || u_name
FROM applications NATURAL INNER JOIN
  (SELECT SKU, MAX(release_date) maxDate, COUNT(*) nrOfVersions
   FROM versions
   GROUP BY SKU)
NATURAL INNER JOIN developers NATURAL INNER JOIN users
WHERE nrOfVersions = 1 AND trunc(maxDate) = trunc(SYSDATE)
ORDER BY app_name ASC;
```

A *view v\_all\_apps* tem como objetivo mostrar uma listagem de todas as aplicações, bem como o seu número total de downloads (que é calculado através da função *f\_n\_downloads*) e a classificação da aplicação dada pelos utilizadores através das *reviews* (tal como o anterior, este campo também é calculado utilizando a função *f\_rating*). Esta view será posteriormente utilizada no **Oracle Application Express** para mostrar o **Top X** (sendo X: 10, 20, 50, 100, 150, 200).

```
CREATE OR REPLACE VIEW v_all_apps (SKU, app_name, developer, price, downloads, rating) AS
SELECT SKU, app_name, u_id || '-' || u_name, base_price, f_n_downloads(SKU),
  TO_CHAR(f_rating(SKU), '9.99')
FROM users NATURAL INNER JOIN developers NATURAL INNER JOIN applications
  NATURAL INNER JOIN (SELECT DISTINCT SKU FROM versions)
  NATURAL INNER JOIN tiers;
```

A *view v\_reviews* é uma junção dos dados da tabela *reviews* e da tabela *review\_download*. À semelhança da anterior, esta *view* também será utilizada no **Oracle Application Express** que irá permitir a inserção de *reviews* que serão automaticamente propagadas para a *review\_download* (através de um trigger que será detalhado mais à frente).

```
CREATE OR REPLACE VIEW v_reviews AS
SELECT *
FROM reviews NATURAL INNER JOIN review_download;
```

## 2.7.5. Criação de Triggers

O *Trigger* `t_payment` vai garantir que um utilizador só é obrigado a ter um cartão de crédito associado a ele (não **NULL**) quando estiver prestes a fazer *download* de uma aplicação que seja paga, ou seja, não há impedimentos aquando do *download* de uma aplicação grátis ou aquando do *update* de uma aplicação anteriormente adquirida.

```

CREATE OR REPLACE TRIGGER t_payment
BEFORE INSERT ON downloads
FOR EACH ROW
DECLARE
  app_price REAL := 0;
  c_card VARCHAR(19);
BEGIN

  IF f_user_latest_version(:NEW.u_id, :NEW.SKU) IS NULL THEN
    SELECT credit_card INTO c_card
    FROM users
    WHERE :NEW.u_id = u_id;

    IF c_card IS NULL THEN

      SELECT DISTINCT base_price INTO app_price
      FROM versions NATURAL INNER JOIN applications NATURAL INNER JOIN tiers
      WHERE SKU LIKE :NEW.SKU;

      IF app_price <> 0 THEN
        RAISE_APPLICATION_ERROR(-20001,
          'Erro: e necessario um cartao de credito para comprar uma app. ');
      END IF;

    END IF;
  END IF;
END;
/

```

O *Trigger t\_download\_version* vai garantir que um utilizador apenas pode transferir a versão mais recente de uma aplicação e que a data do *download* não pode ser mais antiga que a data de lançamento dessa mesma versão.

```

CREATE OR REPLACE TRIGGER t_download_version
BEFORE INSERT ON downloads
FOR EACH ROW
DECLARE
  ver VARCHAR(30);
BEGIN
  ver := f_latest_version(:NEW.SKU);

  IF ver IS NULL THEN
    RAISE_APPLICATION_ERROR(-20001,
      'Erro: nao foi possivel encontrar uma versao da aplicacao.');
```

**END IF;**

```

  IF :NEW.v_number NOT LIKE ver THEN
    RAISE_APPLICATION_ERROR(-20001,
      'Erro: so pode transferir a ultima versao da aplicacao.');
```

**END IF;**

```

  IF trunc(:NEW.dl_date) < trunc(f_version_release_date(:NEW.SKU, :NEW.v_number))
  THEN
    RAISE_APPLICATION_ERROR(-20001,
      'Erro: a data do download nao pode ser inferior a data de lancamento da propria versao.');
```

**END IF;**

```

END;
/
```

O *Trigger t\_new\_review* vai garantir que um utilizador apenas faz uma *review* por cada versão de uma aplicação que já transferiu e que a data da *review* seja igual ou superior à data do download da versão que pretende classificar.

```

CREATE OR REPLACE TRIGGER t_new_review
BEFORE INSERT ON review_download
FOR EACH ROW
DECLARE
    already_reviewed NUMBER := 0;
    download_date DATE;
    review_date DATE;
BEGIN

    IF f_user_latest_version(:NEW.u_id, :NEW.SKU) NOT LIKE :NEW.v_number THEN
        RAISE_APPLICATION_ERROR(-20001,
            'Erro: so e possivel classificar a ultima versao da aplicacao que o utilizador transferiu.');
```

**END IF;**

```

    SELECT COUNT(*) INTO already_reviewed
    FROM reviews NATURAL INNER JOIN review_download NATURAL INNER JOIN
downloads
    WHERE u_id = :NEW.u_id AND v_number LIKE :NEW.v_number AND SKU
    LIKE :NEW.SKU;

    IF already_reviewed <> 0 THEN
        RAISE_APPLICATION_ERROR(-20001,
            'Erro: o utilizador ja classificou essa versao da aplicacao. ');
    END IF;

    SELECT dl_date INTO download_date
    FROM downloads
    WHERE v_number LIKE :NEW.v_number AND u_id = :NEW.u_id AND SKU
    LIKE :NEW.SKU;

    SELECT rev_date INTO review_date
    FROM reviews
    WHERE rev_id = :NEW.rev_id;

    IF trunc(review_date) < trunc(download_date) THEN
        RAISE_APPLICATION_ERROR(-20001,
            'Erro: a data da classificacao tem de ser igual ou superior a data do download da
aplicacao.');
```

**END IF;**

```

    END;
/
```

O *Trigger t\_ins\_v\_reviews* é utilizado para propagar as inserções na view *v\_reviews* até às tabelas **reviews** e **review\_download**.

```

CREATE OR REPLACE TRIGGER t_ins_v_reviews
INSTEAD OF INSERT ON v_reviews
FOR EACH ROW
DECLARE
  r_id NUMBER;
BEGIN
  INSERT INTO reviews VALUES
    (:NEW.rev_id, :NEW.title, :NEW.comments, :NEW.rev_rating, :NEW.rev_date);

  select seq_reviews.currval INTO r_id from DUAL;

  INSERT INTO review_download VALUES (r_id, :NEW.v_number, :NEW.SKU, :NEW.u_id);
END;
/

```

O *Trigger t\_new\_version* serve para garantir que a data de lançamento de uma versão de uma aplicação é válida, ou seja, a própria versão tem de ser superior às que a aplicação tem (por ex, 1.0.1 é superior a 1.0.0), a *release\_date* inserida tem de ser anterior ou igual à data atual e as datas de lançamento de todas as outras versões já lançadas têm de ser anteriores à nova *release\_date*.

```

CREATE OR REPLACE TRIGGER t_new_version
BEFORE INSERT ON versions
FOR EACH ROW
DECLARE
  latest_release DATE;
BEGIN

  IF :NEW.v_number <= f_latest_version(:NEW.SKU) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Erro: a nova versao deve ser maior que a ultima. ');
  END IF;

  IF trunc(:NEW.release_date) > trunc(SYSDATE) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Erro: data invalida. ');
  END IF;

  latest_release := f_version_release_date(:NEW.SKU, f_latest_version(:NEW.SKU));

  IF latest_release IS NOT NULL AND trunc(:NEW.release_date) < trunc(latest_release)
  THEN
    RAISE_APPLICATION_ERROR(-20001, 'Erro: a release date da nova versao '
      || 'da aplicacao tem de ser mais recente que a data de todas as '
      || 'suas outras versoes');
  END IF;
END;
/

```

O *Trigger t\_review\_date* serve para garantir que a data de uma nova *review* de uma versão de aplicação é anterior ou igual à data atual.

```
CREATE OR REPLACE TRIGGER t_review_date
BEFORE INSERT ON reviews
FOR EACH ROW
BEGIN
  IF trunc(:NEW.rev_date) > trunc(SYSDATE) THEN
    RAISE_APPLICATION_ERROR(-20001, 'Erro: data invalida. ');
  END IF;
END;
/
```

O *Trigger t\_ins\_developers* vai garantir que sempre que inserimos um programador em **developers**, este tem um cartão de crédito associado (não **NULL**).

```
CREATE OR REPLACE TRIGGER t_ins_developers
BEFORE INSERT ON developers
FOR EACH ROW
DECLARE
  c_card VARCHAR(19);
  user_count NUMBER;
BEGIN
  SELECT credit_card, u_id INTO c_card, user_count
  FROM users
  WHERE :NEW.u_id = u_id;

  IF user_count = 0 THEN
    RAISE_APPLICATION_ERROR(-20001,
      'Erro: para se tornar um programador e necessario possuir uma conta'
      || 'de utilizador. ');
  END IF;

  IF c_card IS NULL THEN
    RAISE_APPLICATION_ERROR(-20001, 'Erro: para se tornar um programador e'
      || 'necessario possuir um cartao de credito. ');
  END IF;
END;
/
```

O *Trigger* `t_upd_users` garante que quando se pretende actualizar a tabela **users** de forma a tornar um utilizador regular num programador, este tem de ter um cartão de crédito associado (não **NULL**).

```
CREATE OR REPLACE TRIGGER t_upd_users
BEFORE UPDATE ON users
FOR EACH ROW
DECLARE
    dev_count NUMBER;
BEGIN
    IF :NEW.credit_card IS NULL THEN

        SELECT COUNT(*) INTO dev_count
        FROM developers
        WHERE developers.u_id = :NEW.u_id;

        IF dev_count <> 0 THEN
            RAISE_APPLICATION_ERROR(-20001,
                'Erro: um programador necessita de um cartao de credito. ');
        END IF;

    END IF;
END;
/
```

Os seguintes 3 *Triggers* têm como objetivo o preenchimento automático das Chaves Primárias através da utilização de *Sequences* que foram anteriormente apresentadas:

```
CREATE OR REPLACE TRIGGER t_user_id
BEFORE INSERT ON users
FOR EACH ROW
DECLARE
    user_num NUMBER;
BEGIN
    SELECT seq_users.nextval INTO user_num
    FROM dual;
    :NEW.u_id := user_num;
END;
/
```

```
CREATE OR REPLACE TRIGGER t_review_id
BEFORE INSERT ON reviews
FOR EACH ROW
DECLARE
    id_rev NUMBER;
BEGIN
    SELECT seq_reviews.nextval INTO id_rev
    FROM dual;
    :NEW.rev_id := id_rev;
END;
/
```

Neste *Trigger* é necessário subtrair 1 ao número da tier pois é pretendido que exista a tier 0, que corresponde às aplicações grátis, mas as *Sequences* iniciam sempre com o valor 1.

```
CREATE OR REPLACE TRIGGER t_tier_number
BEFORE INSERT ON tiers
FOR EACH ROW
DECLARE
    tier_num NUMBER;
BEGIN
    SELECT seq_tiers.nextval INTO tier_num
    FROM dual;
    :NEW.t_number := tier_num - 1;
END;
/
```

## 2.8. Descrição da Interface

Na nossa interface está presente uma página que corresponde ao menu inicial onde é possível aceder às páginas subseqüentes: Administradores, Desenvolvedores, Países, Downloads, Moedas, Tiers, Utilizadores e Categorias.

Na página Utilizadores podemos consultar todos os utilizadores presentes na base de dados, sejam eles utilizadores regulares, programadores ou administradores. Podemos ver o ID, o nome, o email, o número de telefone, o número de cartão de crédito e o país do utilizador. No topo da página, à direita, encontra-se o botão de criação de um novo utilizador. Na criação de um utilizador é obrigatório o preenchimento de todos os atributos exceto o número de telefone e o número de cartão de crédito.

Ao carregarmos no botão de editar um utilizador somos levados para uma página semelhante à da criação de um novo utilizador.

Na página Programadores é possível consultar os utilizadores que são programadores da App Store. Podemos ver o ID, o nome, o código de licença e o website do programador. No topo da página, à direita, encontra-se o botão de criação de um novo programador. Na criação de um programador é obrigatório colocar o código de licença. Para seleccionar o utilizador que se irá tornar num programador basta escolher um da select list que se encontra no início do formulário, de salientar que esse utilizador tem de possuir um cartão de crédito.

Ao carregarmos no botão de editar um programador somos levados para uma página semelhante à da criação de um novo programador.

Na página Administradores é possível consultar os utilizadores que são administradores da App Store. Podemos ver o ID, o nome e o código de empregado. No topo da página, à direita, encontra-se o botão de criação de um novo administrador. Na criação de um administrador é obrigatório colocar o código de empregado. Para seleccionar o programador que se irá tornar num administrador basta escolher um da select list que se encontra no início do formulário.

Ao carregarmos no botão de editar um administrador somos levados para uma página semelhante à da criação de um novo administrador.

A página Aplicações é um Master detail onde no master podemos ver as aplicações e no detail as versões do respetivo master. No master podemos ver o SKU, o nome, a descrição, a classificação, o programador, o website, a categoria, a tier, a plataforma e a versão da plataforma da aplicação. No topo da página encontra-se um select list, que permite filtrar as aplicações por categoria, e o número de aplicações existentes para a categoria seleccionada. No canto superior

direito da página, encontra-se o botão de criação de uma nova aplicação. Na criação de uma aplicação é obrigatório colocar todos os atributos exceto o website da aplicação, para além disso é importante referir que se não for escolhida nenhuma plataforma na select list Plataforma, não aparecerá nenhuma opção na select list Versão da Plataforma. Ao carregarmos no botão de editar uma aplicação somos levados para uma página semelhante à da criação de uma nova aplicação.

Ao carregarmos no botão de editar uma aplicação somos levados para uma página de edição onde vemos o detail referente às versões da aplicação. No detail podemos ver o número, a data de lançamento e as novidades da versão e temos ainda a hipótese de criar uma nova versão, ou editar uma já existente. Em ambas é obrigatório o preenchimento de todos os atributos.

Na coluna das transferências está um ícone para cada aplicação, que na verdade é um link que permite um drill-down para a página Downloads. Esta página mostra o ID e o nome do utilizador que fez a transferência, a versão da aplicação transferida e a data da transferência. Para além disso possui também um botão para criar um novo download. Na criação é obrigatório colocar o utilizador que vai fazer a transferência e a data em que esta ocorre.

Na coluna do SKU podemos carregar diretamente no SKU de uma determinada aplicação, pois este é um link que permite um drill-down para a página das Reviews. Esta página mostra o título, o comentário, a classificação e a data da review, assim como a versão da aplicação a que a review diz respeito e o utilizador que fez a review. Para além disso possui também um botão para criar uma nova review. Na criação é obrigatório colocar o utilizador que vai fazer a review, a classificação e a data em que esta ocorre.

Na página Categorias são listadas todas as categorias presentes na base de dados. Esta página mostra o nome e a descrição das categorias e possui um botão, no canto superior direito, para inserir uma nova categoria. Na criação é obrigatório inserir o nome da categoria e a descrição. Se carregarmos numa determinada categoria outra página é aberta onde são mostradas todas as aplicações referentes à categoria selecionada. No cimo vemos o nome da categoria selecionada anteriormente, bem como o número de aplicações com essa mesma categoria e em baixo vemos a informação referente às aplicações.

Na página Países é possível consultar todos os países registados na App Store. Podemos ver o código do país, o nome e a moeda que ele utiliza. No topo da página, à direita, encontra-se o botão de criação de um novo país. Na criação de um país é obrigatório o preenchimento de todos os atributos. Para seleccionar a moeda desse mesmo país basta escolher uma da select list que se encontra no fim do formulário.

Ao carregarmos no botão de editar um país somos levados para uma página semelhante à da criação de um novo país.

Na página Moedas é possível consultar todas as moedas utilizadas na App Store. Podemos ver o código da moeda, o símbolo, o nome e o câmbio (taxa de conversão relativamente ao dólar). No topo da página, à direita, encontra-se o botão de criação de uma nova moeda. Na criação de uma moeda é obrigatório o preenchimento de todos os atributos.

Ao carregarmos no botão de editar uma moeda somos levados para uma página semelhante à da criação de uma nova moeda.

Na página Tiers é possível consultar todos os tiers da App Store. Podemos ver o ID do tier e o preço base (em dólares). No topo da página, à direita, encontra-se o botão de criação de um novo tier. Na criação de um tier é obrigatório colocar unicamente o preço base, já que o ID é automaticamente gerado pela base de dados.

Ao carregarmos no botão de editar um tier somos levados para uma página semelhante à da criação de um novo tier.

Na página Top Aplicações é possível consultar todas as aplicações da App Store, ordenadas pelo tipo de listagem (por download ou por classificação), pelo tipo de aplicações (gratuitas, pagas, ou até ambas). Por fim, escolhe-se o Top a mostrar: 10, 20, 50, 100, 150, 200. Estas três últimas funcionalidades poderão ser acedidas através das Select Lists presentes no topo desta página. Como esta página serve unicamente para informar ao utilizador de como está constituído o top, não faria sentido permitir a inserção/remoção/alteração de dados. Podemos ver o SKU da aplicação, o seu nome, o seu programador, o seu preço (em dólares), o seu número total de downloads recebidos e qual a sua classificação média (sendo que ambos são atributos calculados).

Na página Novas Aplicações é possível consultar todas as aplicações da App Store, que foram lançadas no dia corrente (excetuando updates). Como esta página serve unicamente para informar ao utilizador de quais são as aplicações que foram lançadas no dia atual, não faria sentido permitir a inserção/remoção/alteração de dados. Caso realmente existam aplicações novas, podemos ver o SKU da aplicação, o seu nome e o seu programador.

Em todas as páginas de listagem existe, na parte superior, uma Search Bar para pesquisas que se pretendam realizar.

Em todas as páginas onde se procede à edição de valores o Help Text tem uma pequena descrição acerca do conteúdo esperado para o campo selecionado.

### 3. Manual de Utilizador

Página inicial onde se encontram todas as ligações para as páginas subsequentes da base de dados:

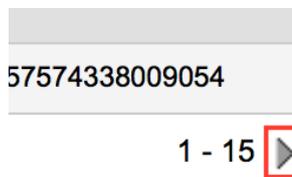


Ao escolhermos o *link* **Utilizadores** podemos observar a seguinte página que contém todos os dados de todos os utilizadores registados no sistema:

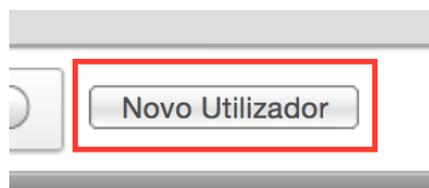
ID	Nome	Email	Número de Telefone	Cartão de Crédito	País
1	Joao	joa@a.pt	-	1235463745745745757	Portugal
2	Manel	manel@fct.pt	919999999	1235463745745745757	Estados Unidos da America
3	Maria Joao	maria_joao@gmail.com	923352999	1235463745745745757	Estados Unidos da America
4	Miguel Mendes	mmendes@gmail.com	-	-	Brasil
5	Ka Wai Sio	ka_w_sio@gmail.com	-	-	Republica Popular da China
6	Valeria Sousa	vals23@hotmail.com	934880249	1112136758350978	Brasil
7	John Smith	jjj@gmail.com	923352999	12284673880956384	Canada
8	Joe	joe@gmail.com	923352999	2314158603940571	Australia
9	Marta Correia	martuxa.fofa@gmail.com	923432679	-	Portugal
10	Joana Silva	joana_silva@fct.pt	-	-	Portugal
11	Carla Pires	carla_p@gmail.com	923352999	1235463745745745757	Portugal
12	Antonio	to.14@hotmail.com	-	-	Portugal
13	Jean	jean_1990@gmail.com	913555629	1332654788903405	Suica
14	Ivan	ivan@gmail.com	-	-	Federacao Russa
15	Maria Luisa	maria_Luisa@hotmail.com	923352978	1457574338009054	Portugal

1 - 15 ▶

É possível transitar entre as várias páginas de utilizadores existentes na *App Store* através da seta no fundo da página:



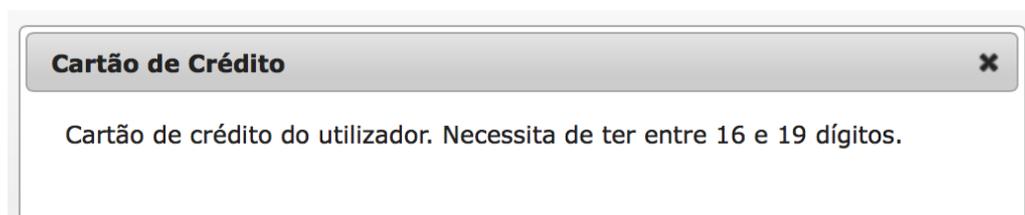
Ainda na página dos utilizadores, é também possível criar uma nova entrada na tabela clicando no botão “Novo Utilizador” que se encontra no topo da página:



De seguida é apresentado um formulário para o utilizador preencher, em que os dados essenciais à nova entrada do utilizador, ou seja, de preenchimento obrigatório estão assinalados com um \*. Segue um exemplo teste já preenchido com todas as informações:

Criar/Editar Utilizadores	
*Nome	Conta Teste
*Email	teste@mail.com
<b>Número de Telefone</b>	919999999
<b>Cartão de Crédito</b>	12345678912345678
*País	Portugal

Ao preencher o formulário, se o utilizador tiver dúvidas em relação ao dados a inserir em cada um dos campos, pode carregar no nome do atributo, que aparecerá um *Help Text* como o seguinte:



No caso do país, de forma a facilitar a inserção de dados, em vez de introduzir diretamente em texto o país (o que poderia levar a alguns erros, de escrita por exemplo) o preenchimento é feito escolhendo uma das opções presentes na *Select List* apresentada:



Depois de ter preenchido o formulário com todas as informações necessárias, o utilizador deve então confirmar a criação do novo utilizador, clicando no botão que se encontra no canto superior direito.



O utilizador pode então constatar que o novo utilizador foi de facto criado, e que tem agora uma entrada na tabela dos **Utilizadores**:

Action Processed. ✕

ID	Nome	Email	Número de Telefone	Cartão de Crédito	País
16	Sam Faraday	sam.far@gmail.com	-	-	Australia
17	Anna	anna_34@hotmail.com	-	-	Reino Unido
18	Kuroko	kuroko.san@gmail.com	923352229	-	Japao
19	Diana Lopes	dianinha@fct.pt	921012999	123548774570000057	Portugal
20	Raquel Vieira	raquel_vieira@gmail.com	922222999	1235487745745780057	Portugal
21	Facebook Inc.	facebook@gmail.com	-	1232217745745780057	Estados Unidos da America
22	Chone Startup	pv.chone@gmail.com	-	1232217745745754057	Portugal
23	Google Inc.	google@gmail.com	-	1232217745740054057	Estados Unidos da America
24	RTP	rtp@hotmail.com	-	1232217215745664057	Portugal
25	Joguex	jogos.ex@gmail.com	-	1232217745745751157	Portugal
42	Conta Teste	teste@mail.com	919999999	12345678912345678	Portugal

Caso o utilizador se tenha enganado a preencher algum dado, ou simplesmente queira atualizar alguma informação referente a algum utilizador tem de carregar no botão com um lápis que aparece no lado esquerdo do nome do utilizador:

	25	Joguex
	41	Conta Teste

É então de novo apresentado o formulário com todos os dados anteriormente inseridos, e que podem agora ser alterados. Caso o utilizador tenha efetuado todas as alterações pretendidas e pretenda submeter o novo formulário, deve carregar no botão “Guardar Alterações”, caso tenha mudado de ideias, ou tenha escolhido o utilizador errado para editar, pode carregar no botão “Cancelar” e nenhuma alterações feita será guardada. Também há a hipótese de apagar definitivamente o utilizador da página de **Utilizadores**, para isso basta carregar no botão “Apagar” (atenção, esta operação não pode ser desfeita).

Criar/Editar Utilizadores

Cancelar Apagar Guardar Alterações

\*Nome

\*Email

Número de Telefone

Cartão de Crédito

\*País

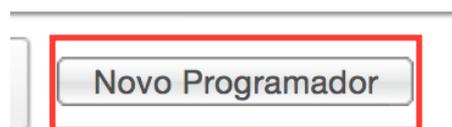
Voltando à página inicial, se escolhermos o *link Programadores* ser-nos-á apresentada uma página com a lista de todos os utilizadores que têm o estatuto de programador na *App Store*:

Início > Programadores

ID	Nome	Código da Licença	Website
1	Joao	AOC8C1MN9	www.joao.pt
7	John Smith	AGT3C1CN9	www.john.com
11	Carla Pires	COC8C4ZN9	www.carla.pt
14	Ivan	CIC2C4ZN2	www.ivan.pt
19	Diana Lopes	BBC8C1MM3	www.dianalopes.com
20	Raquel Vieira	ADC5C1MN3	www.raquel.pt
21	Facebook Inc.	ADA5C2MO4	www.facebook.com
22	Chone Startup	ADB5C3MN5	www.chonestup.pt
23	Google Inc.	ADC5C4MN6	www.google.com
24	RTP	ADD5C5MN7	www.rtp.pt
25	Joguex	ADE5C6MN8	www.joguex.pt

1 - 11

O utilizador pode escolher criar um novo programador carregando no botão que se encontra no canto superior direito:



De seguida é apresentado um novo formulário referente às informações do novo programador.

**Criar/Editar Programador**

**Utilizador**

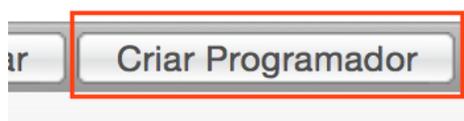
**\*Código da Licença**

**Website**

Como um programador necessita de ter uma conta na *App Store*, é obrigatório escolher, da *Popup LOV*, o utilizador que se quer tornar num programador. Para isso, clicamos no botão da seta, como está assinalado na seguinte imagem. Após esta ação, uma janela irá aparecer, com todos os utilizadores registados na base de dados e, para escolher o pretendido, simplesmente clica-se nele.

- [1 - Joao](#)
- [2 - Manel](#)
- [3 - Maria Joao](#)
- [4 - Miguel Mendes](#)
- [5 - Ka Wai Sio](#)
- [6 - Valeria Sousa](#)
- [7 - John Smith](#)
- [8 - Joe](#)
- [9 - Marta Correia](#)
- [10 - Joana Silva](#)
- [11 - Carla Pires](#)
- [12 - Antonio](#)
- [13 - Jean](#)
- [14 - Ivan](#)
- [15 - Maria Luisa](#)
- [16 - Sam Faraday](#)
- [17 - Anna](#)
- [18 - Kuroko](#)
- [19 - Diana Lopes](#)
- [20 - Raquel Vieira](#)
- [21 - Facebook Inc.](#)
- [22 - Chone Startup](#)
- [23 - Google Inc.](#)
- [24 - RTP](#)
- [25 - Joguex](#)
- [41 - Conta Teste](#)

Após o preenchimento dos dados referentes ao novo programador, o utilizador deve premir o botão para submeter a ação: “Criar Programador”:



Tal como foi referido no ponto anterior, relativamente à criação de utilizadores, se a inserção terminar com sucesso, a nova entrada ficará visível na tabela dos programadores.

Na página principal, se carregarmos no *link Administradores*, a seguinte página será apresentada:

ID	Nome	Código de Empregado
1	Joao	43
20	Raquel Vieira	20

Nesta página podemos todos os utilizadores que são também administradores da *App Store*. É possível criar um novo administrador carregando no botão “Novo Administrador”. Ao carregar nesse botão, um formulário de criação irá aparecer.

**Criar/Editar Administradores**

**Programador**

**\*Código de Empregado**

À semelhança do formulário de criação de um novo programador, é necessário escolher um utilizador presente no *Popup LOV*, mas neste caso todos os utilizadores apresentados possuem uma conta de programador.

**Criar/Editar Administradores**

**Programador**

**\*Código de Empregado**

11 - [Carla Pires](#)  
 22 - [Chone Startup](#)  
 42 - [Conta Teste](#)  
 19 - [Diana Lopes](#)  
 21 - [Facebook Inc.](#)  
 23 - [Google Inc.](#)  
 14 - [Ivan](#)  
 1 - [Joao](#)  
 25 - [Joguex](#)  
 7 - [John Smith](#)  
 24 - [RTP](#)  
 20 - [Raquel Vieira](#)

Para confirmar realmente confirmar e finalizar a inserção, carrega-se no botão “Criar Administrador”.

**Criar Administrador**

Na página principal, se o utilizador carregar no **link Aplicações** terá acesso à seguinte página:

Início > Aplicações

**Categoria:** - Todas -

Número De Aplicações: 30

Aplicações											Nova Aplicação
Transferir	SKU	Nome da Aplicação	Descrição	Classificação	Programador	Website da Aplicação	Categoria	Tier	Plataforma	Versão da Plataforma	
	APP_JOGOS_2014	Jogo do Quadrado	Diverte-te com o quadrado magico	.00	1 - Joao		Jogos	0	Android	1.0.0	
	APP_RTP_2015_NEWSIOS	iOS_RTPjornal	As noticias da atualidade em qualquer lugar	.00	24 - RTP		Noticias	0	iOS	7.2	
	APP_RTP_2015_NEWS	RTPjornal	As noticias da atualidade em qualquer lugar	3.00	24 - RTP		Noticias	0	Android	3.0.0	
	APP_RTP_2015_TVA	RTPtv	Veja a RTP em qualquer lugar	.00	24 - RTP		Videos	0	Android	1.0.0	
	APP_RTP_2015_TVIX	X_RTPtv	Veja a RTP em qualquer lugar	.00	24 - RTP		Videos	0	OS X	10.0.0	
	APP_RTP_2015_TVW	W_RTPtv	Veja a RTP em qualquer lugar	.00	24 - RTP		Videos	0	Windows	7.2.0	
	APP_GOOGLE_2015_MAP	GoogleMaps	Viage e descubra com os nossos mapas	.00	23 - Google Inc.		Utilitarios	0	Android	1.0.0	
	APP_CHONE_2015_MUZ	Muzzy	O melhor companheiro para o iTunes	.00	22 - Chone Startup		Musica	0	OS X	10.0.0	
	APP_FB_2015_FBM	FacebookMessengerI	Conecta-te com os teus amigos.	.00	21 - Facebook Inc.		Social	0	iOS	6.1.3	

Nesta página é possível visualizar todas as aplicações de todas as categorias existentes na *App Store*. O utilizador pode escolher ver apenas as aplicações referentes a uma categoria, para isso basta selecionar a categoria pretendida na *Select List* no topo da página. No exemplo seguinte são mostradas apenas as aplicações com a categoria Jogos.

**Categoria**  - Todas -  
 Desporto  
 Educacao  
 Jogos  
 Musica  
 Noticias  
 Social  
 Utilitarios  
 Videos

Número De Aplicações: 30

**Categoria**

Número De Aplicações: 6

1 - 1

Aplicações											Nova Aplicação
Transferir	SKU	Nome da Aplicação	Descrição	Classificação	Programador	Website da Aplicação	Categoria	Tier	Plataforma	Versão da Plataforma	
	APP_JOGOS2_2014	Jogo do Quadrado	Diverte-te com o quadrado magico	.00	1 - Joao		Jogos	0	iOS	7.2	
	APP_JOGOS_2014	Jogo do Quadrado	Diverte-te com o quadrado magico	.00	1 - Joao		Jogos	0	Android	1.0.0	
	APP_IVAN_2015_CB	CortaBanana	Consegues cortar todas as bananas? Experimenta!	.00	14 - Ivan		Jogos	0	Windows Phone	7.1	
	APP_JOQUEX_2015_FF	Flappy Fish	Como o Flappy Bird, mas com Fish	2.80	25 - Jogueux		Jogos	0	iOS	7.0.1	
	APP_JOQUEX_2015_BICHO	PUU	Descobre o teu novo amigo	.00	25 - Jogueux		Jogos	1	iOS	7.0.1	
	APP_JOQUEX_2015_TIROS	Shoot the thing	Tiros, tiros e mais tiros!	.00	25 - Jogueux		Jogos	2	Android	1.0.0	

1 - 6

O utilizador pode inserir uma nova aplicação carregando no botão no canto superior direito. O botão vai carregar um formulário referente às informações sobre a nova aplicação, que deve ser preenchido corretamente, quando assim for é necessário que o utilizador carregue no botão “Criar Aplicação” para submeter a nova aplicação.



**Criar/Editar Aplicações**

\***Plataforma** Windows

\***Versão da Plataforma** 7.2.0

**SKU** APP\_TESTE\_WINDOWS

\***Nome da Aplicação** Minha primeira app

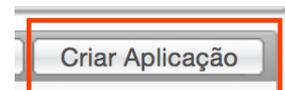
**Descrição** Divirtam-se!

**Website da Aplicação**

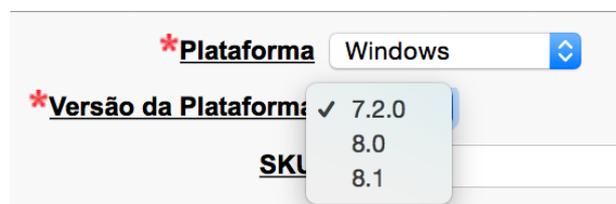
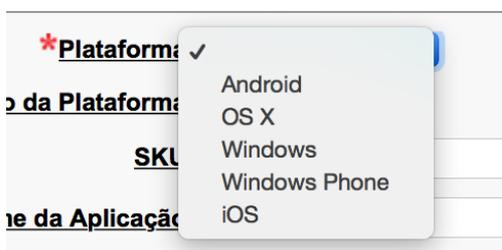
\***Programador** 21 - Facebook Inc.

\***Tier** 3

\***Categoria** Jogos



Em relação ao preenchimento do formulário, é importante referir que é necessário escolher uma plataforma do primeiro *Select List* para que o sistema apresente automaticamente as versões disponíveis para essa dada plataforma, no segundo *Select List*.



Podemos então observar a página **Aplicações** após a inserção:

Editar	Aplicação	Nome	Descrição	Preço	Programador	Website	Categoria	Idioma	Preço	
	APP_CARLA_2015_NOT	AllNews	Esta aplicação permite-te saber todas as notícias do mundo	2.50	11 - Carla Pires	noticias.com	Noticias	2	Windows	8.1
	APP_JOGUEX_2015_TIROS	Shoot the thing	Tiros, tiros e mais tiros!	.00	25 - Jogueux		Jogos	2	Android	1.0.0
	APP_TESTE_WINDOWS	Minha primeira app	Divirtam-se!	.00	21 - Facebook Inc.		Jogos	3	Windows	7.2.0

Previous | row(s) 16 - 30 of 31 | Next

Mais uma vez, caso tenha ocorrido um erro ao inserir os dados, ou caso haja necessidade de atualizar alguma da informação referente a uma aplicação, basta carregar no botão em forma de lápis à esquerda da aplicação em questão. De referir que na página de edição de uma aplicação podemos ver no as versões que a aplicação possui.



**Criar/Editar Aplicações** Cancelar Apagar Guardar Alterações < >

\***Plataforma** Android

\***Versão da Plataforma** 1.0.1

SKU APP\_CARLA\_2015\_TEMP

\***Nome da Aplicação** TempoJa

\***Descrição** Esta aplicação permite-te estar sempre a par do tempo

**Website da Aplicação**

\***Programador** 11 - Carla Pires

\***Tier** 0

\***Categoria** Utilitarios

**Versões** Nova Versão

Versão	Data de Lançamento	Novidades da versão
1.0.0	25-MAY-15	Lançamento inicial

1 - 1

É também a partir desta página que o utilizador pode criar uma nova versão para uma aplicação carregando no botão “Nova Versão”, no fundo da página. Depois disso é novamente apresentado um formulário acerca das informações da nova versão e volta a ser necessário a confirmação do utilizador, carregando no botão “Criar Versão”.

Nova Versão

**Criar/Editar Versão**

Versão

\*Data de Lançamento  

\*Novidades da versão

Criar Versão

Podemos ver então que a ação foi bem-sucedida e que a versão foi criada com sucesso:

Action Processed. ✕

**Criar/Editar Aplicações** Cancelar

\*Plataforma

\*Versão da Plataforma

SKU

\*Nome da Aplicação

\*Descrição

Website da Aplicação

\*Programador

\*Tier

\*Categoria

**Versões**

Versão	Data de Lançamento	Novidades da versão
 1.0.0	25-MAY-15	Lançamento inicial
 1.0.1	25-MAY-15	Erros corrigidos!

1 - 2

Voltando um pouco atrás, é necessário não esquecer que o utilizador deve decidir o que fazer com as alterações feitas ou não à aplicação. Nomeadamente o utilizador pode cancelar a ação e desta forma nenhuma alteração feita nesta página será guardada (botão “Cancelar”, a verde); o utilizador pode decidir apagar a aplicação da *App Store* (botão “Apagar”, a amarelo); o utilizador pode estar satisfeito com as alterações que foram feitas à aplicação e pretende que estas sejam permanentes, desta forma é necessário que o utilizador guarde as alterações (botão “Guardar Alterações”, a vermelho).

Para além destas ações o utilizador pode também mudar de aplicação, carregando para isso nos botões com setas, para a esquerda e para a direita (a azul), sem ter de voltar atrás.



Após o término da edição, a página das **Aplicações** é novamente apresentada. Se o utilizador passar o rato por cima do atributo *SKU* de uma determinada aplicação, aparecerá uma *tooltip* com uma nota a informar que é possível aceder às *reviews* de cada aplicação, clicando no próprio *SKU* (é notável que o ponteiro do rato, em cima do *SKU*, passa do ponteiro normal para a mão).

<a href="#">APP_JOAO_2015_BD</a>	Joapp	Esta aplicacao p media facilment
<a href="#">APP_JOGOS_2014</a>	Quadrado	Diversos jogos

Visualizar e adicionar críticas à aplicação 'Joapp'

Início > Aplicações > Reviews

SKU APP\_JOAO\_2015\_BD

Go Actions Nova Review

Título	Comentários	Classificação	Data	Versão	Utilizador
-	-	1	25-MAY-15	1.0.1	20 - Raquel Veira
Meior app de sempre!	Recomendo vivamente a que todos os estudantes façam download desta aplicação. Facilitou bastante a minha vida no que toca ao cálculo da minha média. Digam adeus ao Excel.	5	25-MAY-15	1.0.1	2 - Manel
-	-	2	25-MAY-15	1.0.1	1 - Joao

Na página das *Reviews* da aplicação, existe uma *label* (**SKU**) no topo (ver imagem acima) que nos informa a que aplicação as *reviews* que estamos a observar pertencem.

É então possível criar uma nova *review* referente à aplicação escolhida, carregando no botão “Nova Review” no topo da página. Feito isto é necessário preencher o formulário apresentado e confirmar a ação carregando no botão “Criar Review”.

**Criar/Editar Review**

\***Utilizador** 15 - Maria Luisa

**Título** Muito bom!

**Comentários**  
Recomendo esta app a todos.

\***Classificação** 5

\***Data** 25-MAY-15

**Criar Review**

Podemos então observar que a criação da nova *review* foi bem-sucedida:

Action Processed.

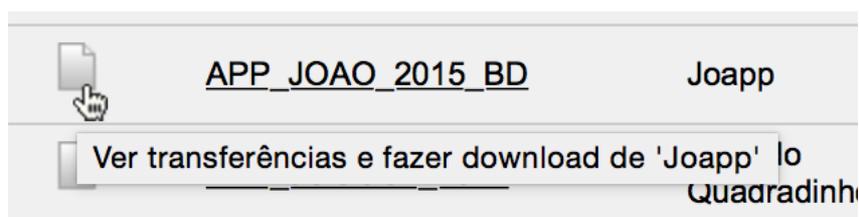
SKU APP\_JOAO\_2015\_BD

Go Actions Nova Review

Ícone	Título	Comentários	Classificação	Data	Versão	Utilizador
✎	-	-	1	25-MAY-15	1.0.1	20 - Raquel Vieira
✎	Melhor app de sempre!	Recomendo vivamente a que todos os estudantes façam download desta aplicação. Facilitou bastante a minha vida no que toca ao cálculo da minha média. Digam adeus ao Excel.	5	25-MAY-15	1.0.1	2 - Manel
✎	-	-	2	25-MAY-15	1.0.1	1 - Joao
✎	Muito bom!	Recomendo esta app a todos.	5	25-MAY-15	1.0.1	15 - Maria Luisa

1 - 4

De regresso à página **Aplicações**, o utilizador pode decidir ainda aceder às transferências de uma aplicação, para isso apenas tem de clicar no ícone em forma de página, na coluna das Transferências, que se encontra à esquerda do *SKU*. Se o utilizador poisar o rato em cima do ícone das transferências aparecerá também uma *tooltip* a informar o utilizador.



Início > Aplicações > Downloads

SKU APP\_JOAO\_2015\_BD

Go Actions Novo Download

Utilizador	Versão	Data
20 - Raquel Vieira	1.0.1	25-MAY-15
15 - Maria Luisa	1.0.1	25-MAY-15
2 - Manel	1.0.0	25-MAY-15
1 - Joao	1.0.1	25-MAY-15
2 - Manel	1.0.1	25-MAY-15
25 - Joguex	1.0.1	25-MAY-15
14 - Ivan	1.0.1	25-MAY-15

1 - 7

É então possível fazer uma nova transferência, através do botão “Novo Download”. É, como sempre, preciso preencher corretamente o formulário apresentado e submeter a criação. Se esta for bem-sucedida podemos observar a inserção do *download* na página das transferências.

**Criar Download**

\*Utilizador 19 - Diana Lopes

\*Data 25-May-15

Criar Download

Action Processed.

SKU APP\_JOAO\_2015\_BD

Go Actions

Utilizador	Versão	Data
20 - Raquel Vieira	1.0.1	25-MAY-15
15 - Maria Luisa	1.0.1	25-MAY-15
19 - Diana Lopes	1.0.1	25-MAY-15
2 - Manel	1.0.0	25-MAY-15
1 - Joao	1.0.1	25-MAY-15
2 - Manel	1.0.1	25-MAY-15
25 - Joguex	1.0.1	25-MAY-15
14 - Ivan	1.0.1	25-MAY-15

1 - 8

Voltando à página principal, se escolhermos o *link* **Categorias**, é-nos apresentada a seguinte página:



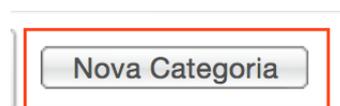
Início > Categorias

Go Actions Nova Categoria

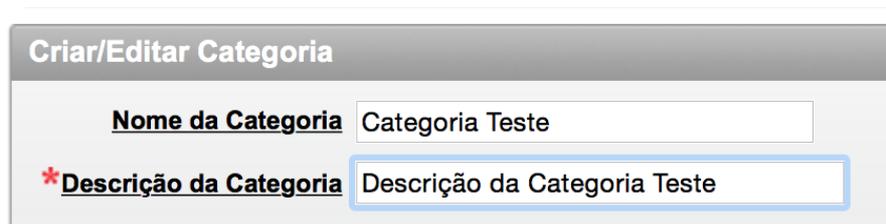
Nome da Categoria	Descrição
 <a href="#">Jogos</a>	Divirta-se!
 <a href="#">Utilitarios</a>	Tudo de util.
 <a href="#">Desporto</a>	Tudo o que precisa para se manter em forma.
 <a href="#">Noticias</a>	Mantenha-se a par da actualidade.
 <a href="#">Social</a>	Conecte-se com a sua rede.
 <a href="#">Musica</a>	Para ouvir as melhores musicas.
 <a href="#">Videos</a>	Os melhores leitores de video da actualidade
 <a href="#">Educacao</a>	Para aprender tudo o que necessita.

1 - 8

Para criar um nova categoria é novamente necessário carregar no botão no canto superior direito, “Nova Categoria”:



Depois disso o utilizador deve preencher o formulário referente à nova categoria.

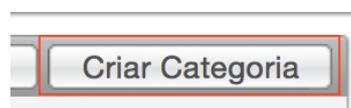


**Criar/Editar Categoria**

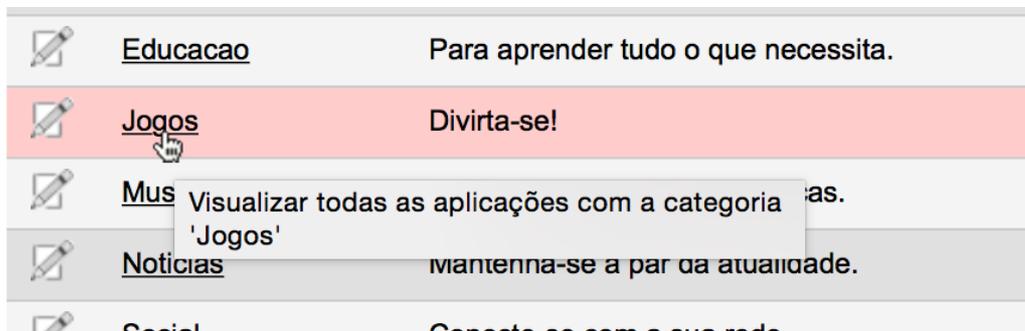
**Nome da Categoria**

**\*Descrição da Categoria**

De seguida é necessário submeter a criação da nova categoria clicando no botão “Criar Categoria”.



Mais uma vez, uma *tooltip* aparece se o utilizador colocar o rato sobre uma categoria, informando que clicando na mesma é possível mostrar todas as aplicações com essa categoria



Após essa ação, o utilizador será redirecionado para a página das aplicações que, desta maneira, irá mostrar automaticamente todas as aplicações que estão de momento dentro da categoria que foi clicada previamente.

Categoria:

Número De Aplicações: 6

1 - 1

Aplicações											Nova Aplicação
Transferir	SKU	Nome da Aplicação	Descrição	Classificação	Programador	Website da Aplicação	Categoria	Tier	Plataforma	Versão da Plataforma	
	APP_JOGOS2_2014	Jogo do Quadrado	Diverte-te com o quadrado magico	.00	1 - Joao		Jogos	0	iOS	7.2	
	APP_JOGOS_2014	Jogo do Quadrado	Diverte-te com o quadrado magico	.00	1 - Joao		Jogos	0	Android	1.0.0	
	APP_IVAN_2015_CB	CortaBanana	Consegues cortar todas as bananas? Experimental!	.00	14 - Ivan		Jogos	0	Windows Phone	7.1	
	APP_JOQUEX_2015_FF	Flappy Fish	Como o Flappy Bird, mas com Fish	2.80	25 - Joguex		Jogos	0	iOS	7.0.1	
	APP_JOQUEX_2015_BICHO	PUU	Descobre o teu novo amigo	.00	25 - Joguex		Jogos	1	iOS	7.0.1	
	APP_JOQUEX_2015_TIROS	Shoot the thing	Tiros, tiros e mais tiros!	.00	25 - Joguex		Jogos	2	Android	1.0.0	

1 - 6

De regresso à página principal, se carregarmos no *link Países*, podemos ver a seguinte página:

Início > Países

Código do País	País	Moeda
USA	Estados Unidos da America	Dolar dos Estados Unidos (\$)
POR	Portugal	Euro (€)
JPN	Japao	Iene (¥)
GBR	Reino Unido	Libra esterlina (£)
CHE	Suica	Franco suico (CHF)
CA	Canada	Dolar canadense (\$)
AUS	Australia	Dolar australiano (\$)
RUS	Federacao Russa	Rublo (?)
BRA	Brasil	Real (R\$)
CHN	Republica Popular da China	Renminbi (¥)

A partir desta página podemos adicionar um novo país à *App Store*, para isso é necessário carregar no botão “Novo País”. De seguida o utilizador deve proceder ao preenchimento do formulário com as informações necessárias e quando terminar submeter a ação carregando no botão “Criar País”.



**Criar/Editar Países**

**Código do País**

**\*Nome do País**

**\*Moeda**



De salientar que a escolha da moeda para o novo país deve ser feita através de uma *Select List* que apresenta todas as moedas existentes na *App Store*:



Na página principal se escolhermos o *link* **Moedas**, podemos aceder à seguinte página:

Início > Moedas

	<b>Código da Moeda</b>	<b>Símbolo</b>	<b>Câmbio (em relação ao Dólar)</b>	<b>Nome da Moeda</b>
	USD	\$	1	Dolar dos Estados Unidos
	EUR	€	1.1091	Euro
	JPY	¥	.0082	Iene
	GBP	£	1.5529	Libra esterlina
	CHF	CHF	1.0672	Franco suíço
	CAD	\$	.8193	Dolar canadense
	AUD	\$	.7874	Dolar australiano
	RUB	?	.0201	Rublo
	BRL	R\$	.3326	Real
	CNY	¥	.1612	Renminbi

Para inserir uma nova moeda na *App Store* o utilizador deve carregar no botão “Nova Moeda”.



O utilizador tem então de inserir os dados acerca da moeda no formulário apresentado e de seguida submeter a nova moeda carregando no botão “Criar Moeda”.

**Criar/Editar Moedas**

**Código da Moeda**

**\* Nome da Moeda**

**\* Símbolo**

**\* Câmbio (em relação ao dólar)**

Podemos observar a página **Moedas** resultante da inserção da nova moeda:

Action Processed. X

Código da Moeda	Símbolo	Câmbio (em relação ao Dólar)	Nome da Moeda
USD	\$	1	Dolar dos Estados Unidos
EUR	€	1.1091	Euro
JPY	¥	.0082	Iene
GBP	£	1.5529	Libra esterlina
CHF	CHF	1.0672	Franco suíço
CAD	\$	.8193	Dolar canadense
AUD	\$	.7874	Dolar australiano
RUB	?	.0201	Rublo
BRL	R\$	.3326	Real
CNY	¥	.1612	Renminbi
ASD	###	1.02	Moeda Teste

1 - 11

Se seleccionarmos o *link* **Tiers** na página principal acedemos à seguinte página:



Tier	Preço Base (em dólares)
 0	0
 1	.99
 2	1.99
 3	2.99
 4	3.99
 5	4.99
 6	5.99
 7	6.99
 8	7.99
 9	8.99

O procedimento para criar uma nova tier é semelhante às inserções anteriormente descritas, primeiramente o utilizador carrega no botão “Novo Tier”, de seguida preenche o formulário apresentado com as informações referentes ao novo tier e para finalizar seleciona o botão “Criar Tier”.



Novo Tier

**Criar/Editar Tier**

\***Preço Base (em dólares)** 10.99

Criar Tier

Podemos observar a página resultante da inserção da nova tier:

Action Processed. ×

🔍GoActions ▾Novo Tier

Tier	Preço Base (em dólares)
0	0
1	.99
2	1.99
3	2.99
4	3.99
5	4.99
6	5.99
7	6.99
8	7.99
9	8.99
10	10.99

1 - 11

Ao selecionarmos o *link Top Aplicações* temos acesso à seguinte página:

**Top Aplicações**

**Listagem por:** Downloads ▾

**Tipo de Aplicações:** Todas ▾

**Top:** 10 ▾

SKU	Nome da Aplicação	Programador	Preço (em dólares)	Downloads	Classificação
APP_CARLA_2015_TEMP	TempoJa	11 - Carla Pires	0	17	.00
APP_RTP_2015_NEWS	RTPjornal	24 - RTP	0	8	3.00
APP_JOGUEX_2015_FF	Flappy Fish	25 - Jogueux	0	7	2.80
APP_FB_2015_FBMA	FacebookMessengerA	21 - Facebook Inc.	0	6	3.00
APP_RAQUEL_2015_LEARN	Aprenda	19 - Diana Lopes	.99	5	3.00
APP_IVAN_2015_CB	CortaBanana	14 - Ivan	0	5	.00
APP_CARLA_2015_NOT	AllNews	11 - Carla Pires	1.99	2	3.50
APP_JOAO_2015_BD	Joapp	1 - Joao	.99	2	2.00
APP_JOAO_2015_UT	Lanterna	1 - Joao	0	2	.00
APP_GOOGLE_2015_MAP	GoogleMaps	23 - Google Inc.	0	1	.00

1 - 10

Facilmente se pode constatar que neste momento as aplicações estão listadas por ordem decrescente do número de downloads:

**Top Aplicações**

**Listagem por:** Downloads ▾

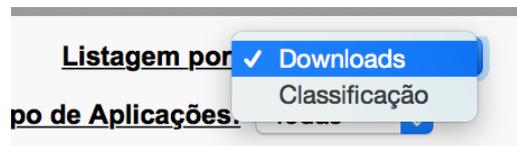
**Tipo de Aplicações:** Todas ▾

**Top:** 10 ▾

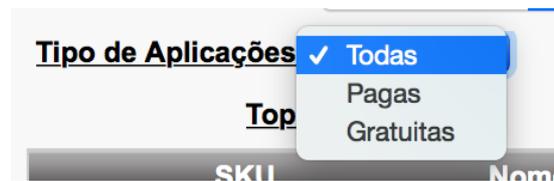
SKU	Nome da Aplicação	Programador	Preço (em dólares)	Downloads	Classificação
APP_CARLA_2015_TEMP	TempoJa	11 - Carla Pires	0	17	.00
APP_RTP_2015_NEWS	RTPjornal	24 - RTP	0	8	3.00
APP_JOGUEX_2015_FF	Flappy Fish	25 - Jogueux	0	7	2.80
APP_FB_2015_FBMA	FacebookMessengerA	21 - Facebook Inc.	0	6	3.00
APP_RAQUEL_2015_LEARN	Aprenda	19 - Diana Lopes	.99	5	3.00
APP_IVAN_2015_CB	CortaBanana	14 - Ivan	0	5	.00
APP_CARLA_2015_NOT	AllNews	11 - Carla Pires	1.99	2	3.50
APP_JOAO_2015_BD	Joapp	1 - Joao	.99	2	2.00
APP_JOAO_2015_UT	Lanterna	1 - Joao	0	2	.00
APP_GOOGLE_2015_MAP	GoogleMaps	23 - Google Inc.	0	1	.00

1 - 10

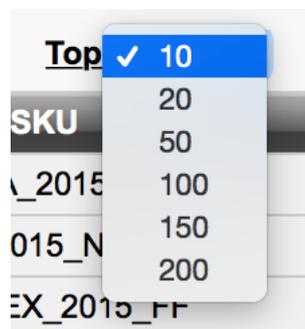
No entanto, é possível alterar os parâmetros de listagem, clicando na *Select List* Listagem por, para listar as aplicações pela sua classificação nas *reviews* dos utilizadores



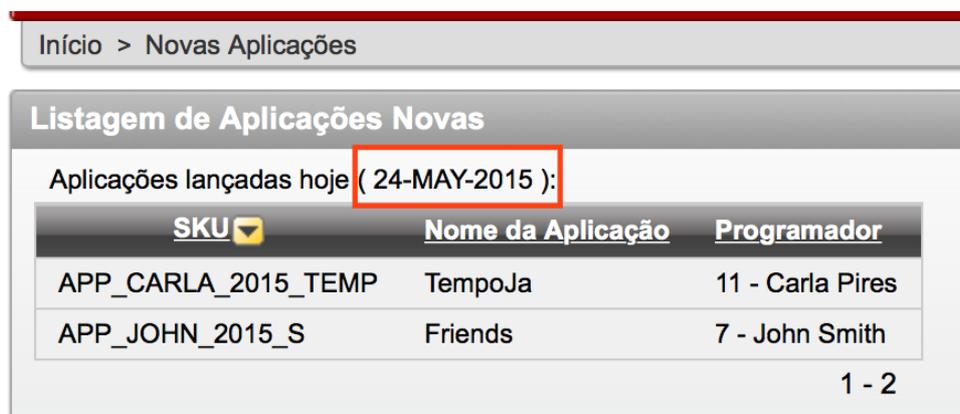
É também possível escolher que tipo de aplicações são listadas:



Assim como definir quantas aplicações (ou seja, o Top) aparecem por página no **Top Aplicações**:



Por último é o utilizador pode escolher o *link* **Novas Aplicações** que apresenta a seguinte página:



Início > Novas Aplicações

### Listagem de Aplicações Novas

Aplicações lançadas hoje ( 24-MAY-2015 ):

<u>SKU</u> ▼	<u>Nome da Aplicação</u>	<u>Programador</u>
APP_CARLA_2015_TEMP	TempoJa	11 - Carla Pires
APP_JOHN_2015_S	Friends	7 - John Smith

1 - 2

Nesta página o utilizador encontra todas as aplicações inseridas na *App Store* no dia atual (explicitado pela própria aplicação), excetuando updates.

## 4. Considerações Finais

Consideramos ter implementado uma base de dados que reflete, a uma escala um pouco mais reduzida, o que se pretende de um sistema *App Store* nos dias atuais.

Este trabalho foi bastante educativo para todos os intervenientes, pois durante todo o processo da estruturação e execução do projeto foi necessário uma extensiva pesquisa em várias Lojas de Aplicações para saber como funcionam e quais os seus pontos mais importantes.

Com isto, pensamos ter cumprido e executado todos os requisitos a que nos propusemos no início do trabalho.

Concluindo, com a resolução deste projeto o grupo aprendeu um pouco mais sobre como os sistemas de *App Stores* funcionam, bem como a criar uma base de dados consistente, simples, eficaz e o menos redundante possível.