**Estrutura do relatório:**

1. Delineamento do projeto;
2. Fases do projeto;
3. Descrição de cada fase;
4. Conclusão;
5. Melhoramentos possíveis;

**Delineamento do projeto:**

O principal objetivo deste projeto seria criar uma base de dados que apresentasse um largo conjunto de informação relativamente a um determinado número de genes associados a uma query de pesquisa escolhida pelo utilizador.

Partindo do princípio que o objetivo é concluído, a realização desta base de dados poderia de certa forma otimizar diversos aspetos de pesquisa de genes, tais como, pesquisar uma determinada percentagem de um nucleótido, ou genes que estejam associados a artigos onde os autores têm como affiliation a Universidade do Minho, ou, a utilização de informação associada a outra base de dados como a Uniprot em simbiose com a informação descrita no NCBI, etc. Em suma, a nossa base de dados irá permitir retirar informação de uma forma mais facil quando comparada ao NCBI.

**Fases do projeto:**

1. Tendo em conta toda a possível informação referente a cada query, realizar o delineamento da Base de dados com aplicação de possíveis entidades, atributos, relacionamentos e respetiva cardinalidade.
2. Análise e construção de código para extração da informação pretendida indo ao encontro da Base de dados criada anteriormente.
3. Análise e construção de código para a povoação das tabelas criadas na fase 1 com a informação obtida na fase 2

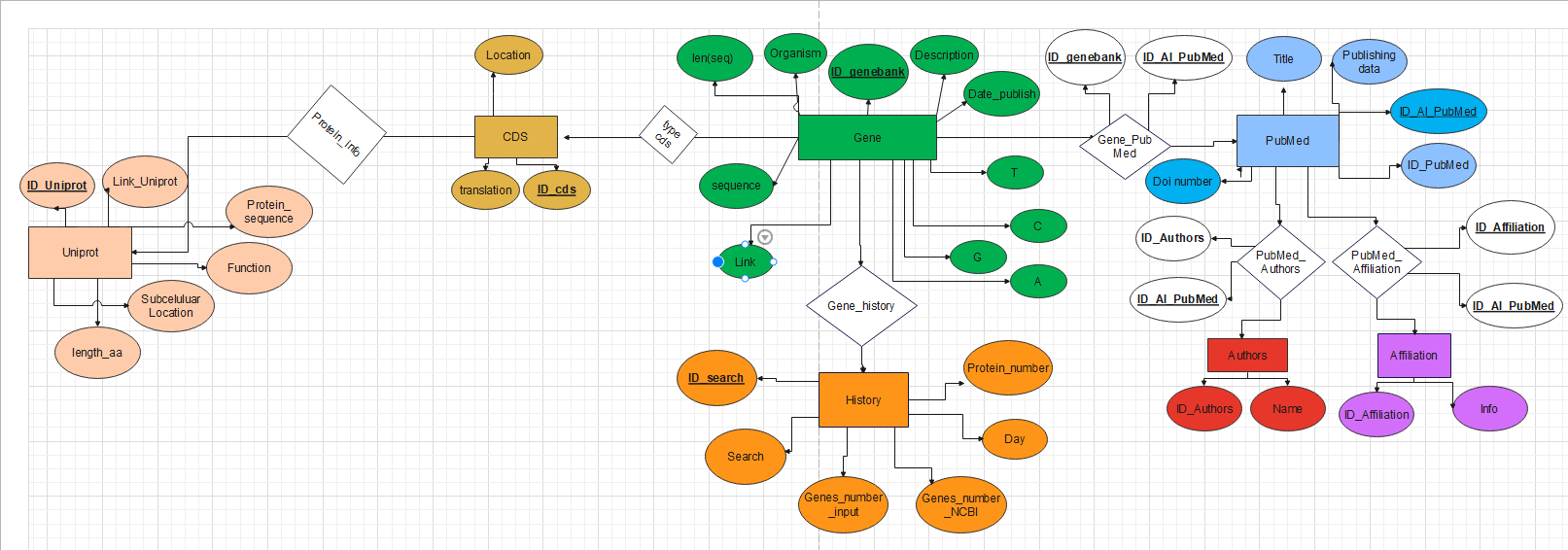
**Descrição de cada fase:**

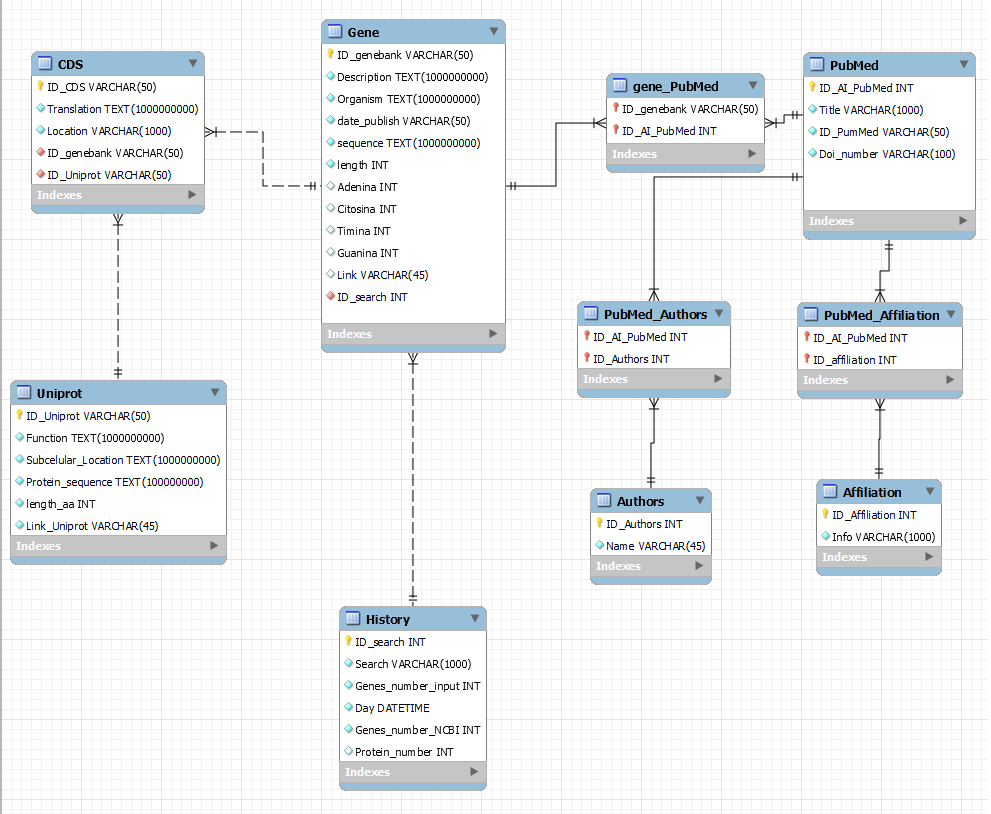
Segundo a informação pretendida, realizou-se a organização desta em entidades, representadas por diversos atributos com diferentes tipos de relacionamentos entre si., como é demonstrado nas tabelas abaixo. Tabelas que serviram de orientação para a realização das restantes fases do trabalho.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identity** | **Name** | **Tipo e dominio** | **Multivalue** | **Zero** | **Keys** |
| Gene | **ID\_genbank** | VARCHAR | N | N | S- |
| Description | TEXT | N | N | N |
| Organism | TEXT | N | N | N |
| Date\_publish | VARCHAR | N | N | N |
| sequence | Text | N | N | N |
| length | INT | N | N | N |
| Adenina | INT | N | S | N |
| Citosina | INT | N | S | N |
| Guanina | INT | N | S | N |
| Timina | INT | N | S | N |
| Link | VARCHAR | N | S | N |
| PubMed | **ID\_AI\_PubMed** | INT | N | N | S |
| Title | VARCHAR | N | N | N |
| ID\_PubMed | VARCHAR | N | N | N |
| Doi\_number | VARCHAR | N | N | N |
| Authors | **ID\_Authors** | INT | N | N | S |
| Name | VARCHAR | N | N | N |
| Affiliation | **ID\_Affiliation** | INT | N | N | S |
| Info | VARCHAR | N | N | N |
| History | **ID\_search** | INT | N | N | S |
| Search | VARCHAR | N | N | N |
| Genes\_number\_input | INT | N | N | N |
| Day | DATETIME | N | N | N |
| Genes\_number\_NCBI | INT | N | N | N |
| Protein\_number | INT | N | N | N |
| CDS | **ID\_CDS** | VARCHAR | N | N | S |
| Translation | TEXT | N | N | N |
| Location | INT | N | N | N |
| Uniprot | **ID\_Uniprot** | VARCHAR | N | N | S |
| Function | TEXT | N | N | N |
| Subcellular\_Location | TEXT | N | N | N |
| Protein sequence | TEXT | N | N | N |
| Length (a.a) | INT | N | N | N |
| Link\_Uniprot | VARCHAR | N | N | N |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identity** | **#** | **Relationship** | **Identity** | **#** |
| Gene | 1…N | Gene\_PubMed | PubMed | 1…N |
| 1…1 | Type\_CDS | CDS | 1…N |
| 1…N | Gene\_History | History | 1…1 |
| PubMed | 1…N | PubMed\_Authors | Authors | 1…N |
| 1…N | PubMed\_Affiliation | Affiliation | 1…N |
| CDS | 1…N | Protein\_info | Uniprot | 1…1 |

As tabelas apresentadas serviram como guia para a realização do modelo conceptual no software *Wondershare EdrawMax*.



Após o desenho do modelo conceptual e, tendo em particular atenção a cardinalidade entre cada entidade/relacionamento, realizou-se o modelo lógico utilizando o software *MySQL Workbench.*

Após a realização do modelo lógico, seguiu-se para o modelo físico usando o mesmo software referido anteriormente.

**2.**

Em primeiro lugar, criou-se um código com o prepósito de através de uma função (*url get*), obter os links associados ao NCBI a partir de uma determinada query e, destes links com auxílio de expressões regulares extrair os respetivos IDS genebank. Referir, que criamos a possibilidade ao utilizador de escolher um determinado número de IDs tendo sempre como limite máximo o valor 20, o que está associado à própria paginação do NCBI. Como os IDs poderiam aparecer repetidos a nossa estratégia passou por criar uma lista(n\_genes) onde os ids apenas apareciam uma única vez, sem repetições.

Uma imagem com mesa

Descrição gerada automaticamente.

Dos IDs obtidos por *web scraping*, alguns já foram removidos do genebank e, como tal, a sua associação dava erro. Desta forma, a solução encontrada foi o recurso ao biopython, nomeadamente o modulo Entrez, para extrair apenas os que não foram removidos.



Tendo em atenção que um dos atributos pretendidos para a entidade Gene é o Link\_Gene fizemos uma associação entre este atributo (variavel no código) e os links obtidos com a utilização de uma query.

A partir da estratégia referida anteriormente e com principal recurso a Biopython foi possível obter descrições, organismos, sequência, tamanho de sequência, percentagem dos nucleótidos, data de publicação do gene. Posto isto, a informação necessária para preencher os atributos da entidade “Gene” foi extraída.



O próximo aspeto focou-se na obtenção da informação relativa à entidade PubMed, onde através dos links dos IDs já “processados” referentes ao NCBI extrair o seu conteúdo utilizando o módulo *Beautifulsoup* do *bs4*. Todavia, para ser mais fácil trabalhar com este conteúdo, criamos uma n lista(listas) onde o output estava em formato utf-8.

Uma imagem com texto

Descrição gerada automaticamente

De seguida, a estratégia utilizada passa pela utilização de expressões regulares extrair da lista referida anteriormente os IDs do NCBI e os Ids da PubMed. A partir daqui, criou-se um dicionário onde os IDs dos genes seriam as keys, e os IDs da PubMed (referentes a cada gene) os valores. No caso do IDs do gene não ter nenhum ID PubMed associado, o valor desde seria uma lista vazia.

Posto isto, criaram-se duas listas diferentes, uma lista para armazenar os IDs dos genes(id\_ncbii) e outra lista para armazenar os IDs da pubmed(ID\_PUB), no caso de não existir IDs de PubMed associado, será adicionado à lista a string “N/A”. Como existe a possibilidade de termos IDs PubMed repetidos, utilizamos um código para criar uma lista(*new\_list\_)* só com ids não repetidos.

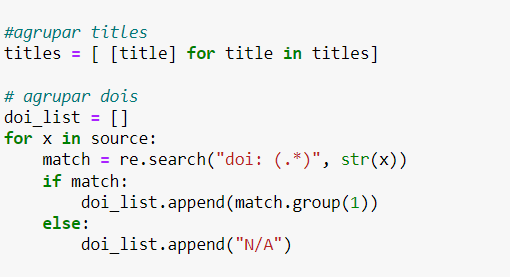
Uma imagem com texto

Descrição gerada automaticamente

Para extrair informação relativamente aos artigos na PubMed, usamos o modulo Entrez de biopython, este módulo permite através dos IDs da PubMed contidos na lista (*new\_list\_)* extrair informações como os títulos, os autores, a fonte, *affiliation* e o *doi.*

Uma imagem com texto

Descrição gerada automaticamente

O *doi* estava contido na informação proveniente da lista *source*, e para ser extraído apenas o doi, utilizou-se expressões regulares, no caso de não existir nenhum *doi*, adicionou-se a *string* N/A. Posto isto, a informação necessária para preencher os atributos referentes à entidade “PubMed” foi recolhida.

No que concerne à estratégia para a entidade autores e respetivos atributos realizou-se um dicionário que através dos autores extraídos com recurso ao *biopython*, tem como chaves, os IDs PubMed e como valor o nome dos autores. Referenciar, que como alguns artigos possuíam muitos autores, criamos uma lista onde continha apenas os 5 primeiros autores extraídos de cada artigo. A partir deste ponto, criaram-se duas listas diferentes, uma lista para armazenar os IDs da PubMed(*pubmed\_list*) e outra lista para armazenar os nomes dos autores(*authors\_list*). Como existe a possibilidade de termos nomes de autores repetidos, utilizamos um código para criar uma lista(*new\_list\_authors*) só com nomes de autores não repetidos. Assim, a informação necessária para preencher os atributos referentes à tabela “*Authors*” está totalmente recolhida.

Uma imagem com texto

Descrição gerada automaticamente

Em relação à entidade “*affiliation*” raciocínio utilizado é idêntico ao descrito anteriormente para a entidade “*Authors*”, com a particularidade de que caso um id pubmed não tenha associado nenhuma affiliation, será acrescentado à *lista affi\_list* a string N/A. Desta forma, a informação para esta entidade, está recolhida.

Uma imagem com texto

Descrição gerada automaticamente

Relativamente à entidade CDS, a estratégia principal baseia-se na utilização de biopython. A função get\_*CDS*\_info, cria uma lista de listas em que cada uma é composta por: ID\_CDS, Location e Translation (atributos pretendidos). Tendo em atenção, que existe uma relacionamento especifico entre esta entidade e a entidade gene, a estratégia utilizada foi a antecipada criação de um dicionário(*result\_dict*) que permite associar cada atributo desejado a um determinado *ID\_genebank* .



Para a entidade Uniprot o primeiro passo, foi a obtenção dos links associados a uma determinada query de pesquisa,com recurso à mesma estratégia utilizada anteriormente para obtenção dos links ncbi, com a particularidade da criação de uma nova função *url\_get\_id\_p*. Desta forma, o atributo-*Link\_Uniprot* foi conseguido.

Recorremos à *webAPI* do Uniprot e atribuindo um field a cada atributo pretendido para esta entidade, conseguiu-se adquiri toda a informação desejada. Todavia esta informação aparecia de uma forma pouca organizada e , como tal, com recurso a expressões regulares retirou-se a informação pretendida com a apresentação desejada e correto conteúdo. De salientar, que cada um destes atributos da entidade Uniprot está associado a um determinado *ID\_CDS*.

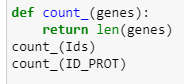
Uma imagem com texto

Descrição gerada automaticamente

Para a entidade *History*, fez-se o import do módulo *datetime* para extrair a hora e data do momento em que se fez a pesquisa, ou seja que se atribui uma query (atributo search). Associado a isso elaborou-se uma função que conta o número de proteínas, o número de IDs NCBI e o numero de IDs requeridos pelo o utilizador completando todos os atributos associados à entidade em questão.

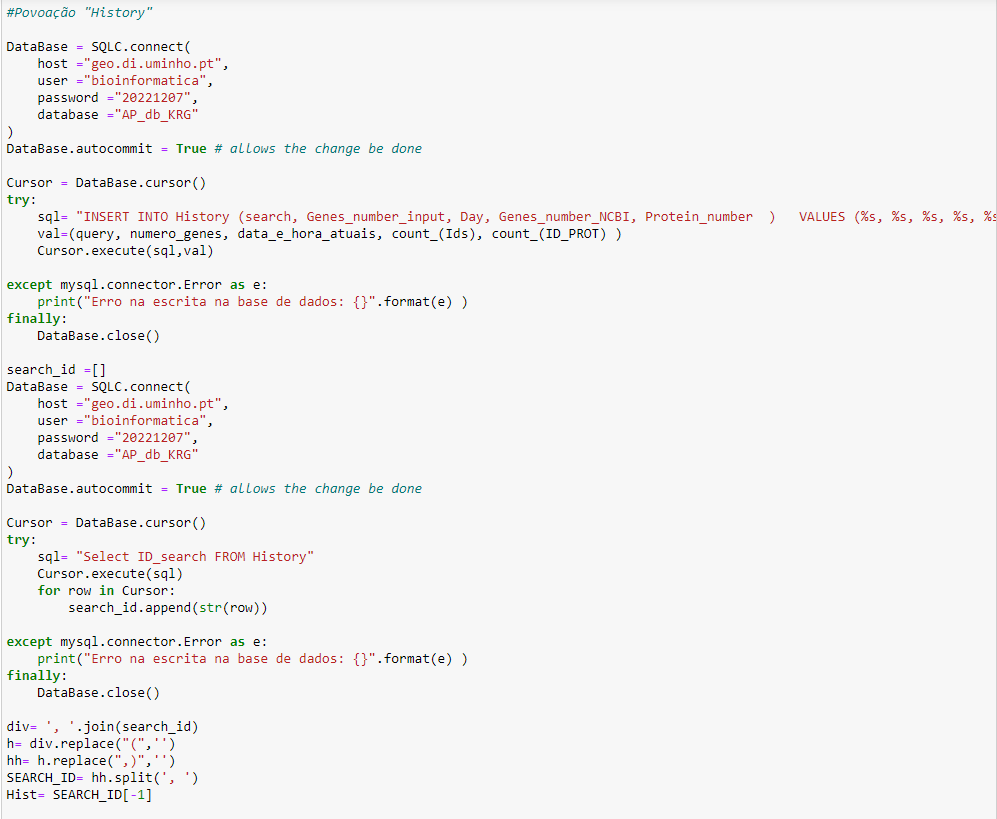






**3.**

Para povoar os atributos das entidades, é necessário fazer a conexão com a base de dados criada anteriormente (*SQLC.connect*), depois é necessário inserir a “ação” que pretendemos executar (*Insert*) indicando a entidade onde pretendemos executar essa ação selecionando os atributos que vão ser alterados e indicando os dados (*Values*) que se pretende adicionar. Evidenciar que os *values* referidos foram determinados na 2 fase deste projeto.

Em relação à entidade *Histor*y, definiu-se que a chave primária (ID\_*search*) seria autoincrementada e, por isso, a “ação” explicada anteriormente para associar um valor a este atributo primário é ligeiramente diferente. Além disso, como a entidade em questão está relacionada (1 para N) com a entidade *Gene* foi necessário extrair os IDs autoincrementados, sendo que, apenas interessa o último *ID\_searc*h obtido pois, como é o mais atual, é esse que vai estar relacionado com a entidade *Gene* que se vai povoar de seguida.

De modo a comprovar o correto funcionamento do código, utilizou-se uma query de pesquisa escolhida aleatoriamente (Diabetes) e obteve-se a povoação da tabela History.

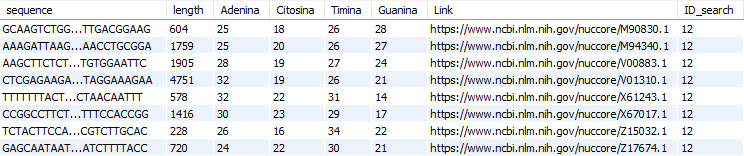


No que concerne à povoação da entidade *Gene* o processo é ligeiramente distinto, uma vez que, como estamos a povoar vário genes ao mesmo tempo, foi necessário usar um *ciclo for*, com recurso ao *enumerate*. Esta estratégia, permite trabalhar com todas as listas criadas na fase 2, separadamente, e a partir daqui associar a cada atributo necessário para povoar esta tabela.

Uma imagem com texto

Descrição gerada automaticamente

O resultado obtido para a povoação da entidade em questão com a query já referida está demonstrado na seguinte tabela, constituída pelos respetivos atributos.

Uma imagem com texto

Descrição gerada automaticamente

Para a entidade *PubMed*, foi utilizado o mesmo raciocínio que na entidade Gene, notando que *a primary key*, é um ID autoincrementado (*ID\_AI\_PubMed*).

Uma imagem com texto

Descrição gerada automaticamente

A povoação resultante está demonstrada na tabela abaixo. Referir que há determinados genes que podem não estar associados a artigos *PubMed* e portanto aparece a *string* N/A.

Uma imagem com texto

Descrição gerada automaticamente

É necessário ter em atenção que, como a entidade Gene está relacionada com a entidade *Pubmed* com cardinalidade de n para n, isto é, um gene pode ter vários artigos associados e vice-versa, foi necessário criar uma lista (*SEARCH\_ID*) com todos os IDs auto incrementados. A partir daqui foi possível saber o número do primeiro ID autoincrementado para depois a cada ID PubMed presentes na lista *ID\_PUB*, pela ordem certa, atribuir o mesmo ID autoincrementado e, assim povoar o relacionamento Uma imagem com texto

Descrição gerada automaticamente*gene\_pubmed*.

Na tabela seguinta, mostra a correta associação entre um ID\_genebank e um ID\_AI\_PubMed, ou seja, uma correta povoação do relecionamente gene\_PubMed.

Uma imagem com mesa

Descrição gerada automaticamente

Para a povoação da entidade *Authors* e *Afilliation* e, respetivos relacionamentos com a entidade PubMed de n para n, foi utilizado o mesmo raciocínio descrito anteriormente, alterando claro, apenas os respetivos atributos.

Os resultados obtidos para a povoação da entidade *authors* e *affiliation* estão representados nas 2 primeiras tabelas seguintes. Sendo as restantes, correspondentes ao relacionado gene\_PubMed com Authors/ Affiliation.

Uma imagem com mesa

Descrição gerada automaticamenteUma imagem com mesa

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

Uma imagem com mesa

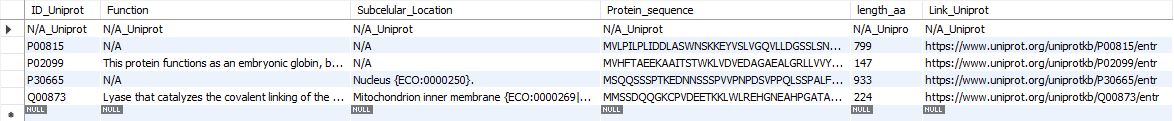
Descrição gerada automaticamente

A estratégia de povoação utilizada para a entidade Uniprot é bastante semelhante às restantes, todavia o caso particular é que poderiam existe genes que não tinham associado nenhum ID\_Uniprot. Posto isso, e com o intuito de não ter a repetição no valor da chave primária (ter diversos ID\_Uniprot = NA), a solução passou por associar todos os genes que não têm referencia na Uniprot ao ID\_Uniprot=N/A\_Uniprot.

Uma imagem com texto

Descrição gerada automaticamente

O resultado da povoação da entidade Uniprot está demonstrado abaixo, com a particularidade referida de poder existir um N/A\_Uniprot que representa que um determinado gene não tem informação associada ao Uniprot.

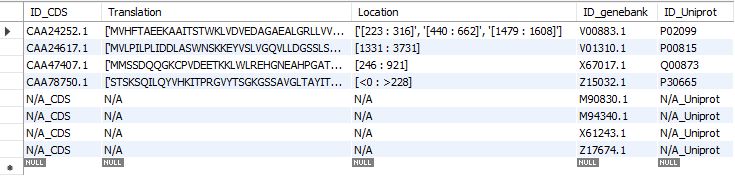


No caso da entidade CDS, como a chave primária é composta por ID\_genebank e ID\_CDS, os valores correspondentes podem aparecer repetidamente. Este aspeto, permite associar a designação N/A\_CDS para diversos ID\_genebank, o que significa que posso ter n genes associados a 1 ID\_CDS=N/A\_CDS.

Uma imagem com texto

Descrição gerada automaticamente

Como demonstrado na tabela seguinte o ID\_CDS apareceu repetidamente com a designação N/A\_CDS associada a diversos genes, o que mostra que a estratégia utilizada para povoar a entidade CDS correu como esperado.



Para a povoação da entidade *History* funcionar corretamente para sucessivas pesquisas, foi necessário limpar todos os dados das restantes entidades e relacionamentos já referidos. Permitindo assim ficar com um histórico de todas as pesquisas feitas, sem ter problemas associados à repetição de chaves primárias repetidas.

Uma imagem com texto

Descrição gerada automaticamente

A povoação da entidade *History* está representada abaixo, com duas *querys* aleatórias inseridas pelo utilizador bem como os restantes atributos, referenciando a data e hora quando esta pesquisa foi realizada.

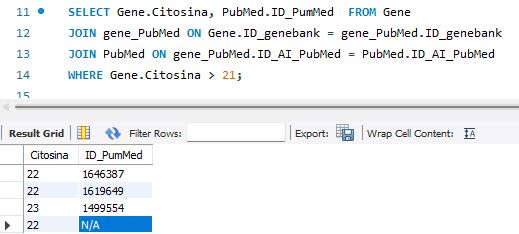
Uma imagem com texto

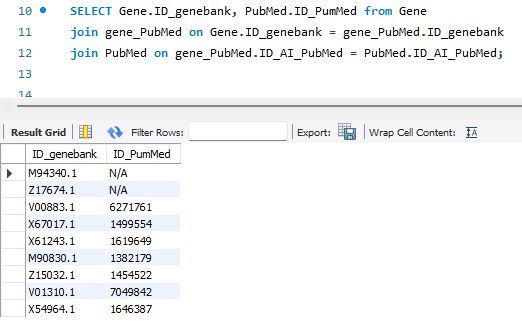
Descrição gerada automaticamente

**Conclusões:**

De modo a verificar a correta povoação da nossa base de dados, tendo em atenção as diversas entidades, atributos e relacionamentos, realizaram-se alguns testes no MySQL. Uma imagem com texto

Descrição gerada automaticamente





Tendo em conta a povoação de cada entidade, verificou-se que os resultados obtidos estão corretos, o que permite concluir que, de forma geral, a nossa base de dados está operacional.

**Melhoramentos Futuros;**

1. Pretende-se aumentar o número de genes possíveis a ser trabalhados, ou seja, passar de 20 para 100/200 (relacionado com a paginação do NCBI).
2. Pretendemos explorar a entidade *History* onde esta passa a ser uma entidade n para n e, por isso, não seja necessário apagar os dados antes de uma nova pesquisa.
3. Otimizar e simplificar o código para funcionar com todas as *querys* que nele se colocam e, ao mesmo tempo, ser ainda mais fácil a sua interpretação.