

APLICAÇÃO WEB PARA O BANCO DE DADOS RELACIONAL BIOSTRINGDB: ANÁLISE DO PROCESSO DE ELABORAÇÃO DO BACK-END

BIOSTRINGDB RELATIONAL DATABASE WEB APPLICATION: ANALYSIS OF THE BACK-END ELABORATION PROCESS

Sergio Gustavo Mendonça Pyrrho Moreira¹

RESUMO: A tecnologia da informação atual permite com que novas áreas de estudo e áreas já existentes possam usufruir de diversos novos sistemas que auxiliam na pesquisa e na ciência, promovendo não só a observação de novos pontos de vista como também na agilidade com que são desenvolvidos tais estudos. Dentre essas áreas se encontra a bioinformática que utiliza da tecnologia para promover pesquisas no campo da biologia. No caso em questão, trata-se do estudo de sequenciamentos genéticos inseridos em um banco de dados relacional. Esse artigo busca retratar o processo de elaboração da aplicação web do banco de dados relacional BioStringDB, analisando seu processo de implementação do *back-end* e a maneira com que as tecnologias usadas foram aplicadas, descrevendo-as e apontando suas respectivas relevâncias para a criação da aplicação, também ressaltando as adaptações necessárias considerando a existência prévia do banco de dados que estaria sendo apenas replicado com a finalidade de testar a aplicação.

1672

Palavras-chave: Aplicação web. Django. PostgreSQL.

ABSTRACT: The current informational technology allows new fields of study and existing ones to benefit from multiple new systems that help in research and science, promoting not only the observation of new points of view as well as agility on which they are developed. Among those fields there is bioinformatics which uses technology to promote researches in the field of biology. In the case at hand the field of study is the genetic sequencing inside a relational database. This article seeks to describe the relational database BioStringDB web application elaboration process, analyzing its back-end implementation process and the way the technology was used for its creation, also highlighting the adaptations needed considering the pre-existing database that would only be replicated due testing of the application.

Keywords: Web application. Django. PostgreSQL.

INTRODUÇÃO

No mundo atual, a tecnologia da informação se faz presente em toda e qualquer área, compreendendo as ciências humanas, exatas e biológicas. A utilização de ferramentas tecnológicas possibilita que áreas como a biomedicina e biologia possam agilizar seus

¹ Discente do curso de Sistemas de Informação da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio).Email: sergiogmpmoreira@gmail.com.

processos de estudos e análise de dados. Um dos casos relevantes a que se pode retratar tal utilização é o caso da representação de sequenciamentos genéticos que não possui uma representação específica de dados, apesar de existirem propostas diversas que apresentam suas interpretações acerca do tema. Segundo Heine et. al (2021), a obtenção de algumas informações e visualizações de grafos de genes podem ser feitas por determinadas aplicações como *Cytoscape*² e *String-DB*³, porém tais aplicações possuem suas próprias maneiras de exibir e tratar os dados criando *clusters* representativos não simplificados para determinadas pesquisas.

A utilização de uma interface padrão de um banco de dados, independente de qual seja o seu sistema, muitas vezes não se faz amistosa considerando todos os comandos em *Structured Query Language* (SQL) necessários para exibir os dados requisitados. Além de que, ao exibir os dados relativos às pesquisas e funções executadas no banco na interface dos próprios sistemas dos bancos, muitas vezes não oferece uma visualização que permita uma fácil compreensão dos dados extraídos. Considerando então a necessidade de adequar a representação de sequenciamento genético a pesquisas específicas de genes, Tristão (2012) propõe em sua tese uma abordagem para armazenamento de tais dados através de um banco de dados relacional de modo a facilitar o acesso e compreensão dos dados em questão. A partir dessa visão foi sugerido por Sergio Lifschitz⁴ a elaboração de uma interface que intermediasse o acesso ao banco estruturado e permitisse que a visualização dos dados fosse mais amistosa aos usuários.

1673

A interface então foi criada através de uma aplicação web que permite a comunicação com o banco de dados através da internet e oferece uma intuitiva e explicativa busca pelos dados em questão. Assim permitindo que bioinformatas, biólogos, pesquisadores e profissionais da área da biomedicina, universidades e instituições de pesquisa possam aproveitar desta proposta para pesquisa e estudo. Através do caso exposto pode-se observar que determinadas tecnologias de informação apresentam possibilidades de transformações digitais em diversos setores, assim fazendo-se válido explorar tais recursos e a forma com que são aplicados.

² “is an open-source software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data” (CYTOSCAPE, 2022).

³ “is a database of known and predicted protein-protein interactions. The interactions include direct (physical) and indirect (functional) associations; they stem from computational prediction, from knowledge transfer between organisms, and from interactions aggregated from other (primary) databases” (STRING, 2022).

⁴ Docente do departamento de informática da Pontifícia Universidade Católica do Rio de Janeiro.

Dado tal contexto, este artigo promove a narrativa da construção e adaptação das ferramentas dispostas para o projeto da aplicação web para o banco de dados BioStringDB com intuito de fomentar e incentivar o uso das mesmas nas áreas de pesquisa, ciência e educação, ressaltando também as possibilidades através da exemplificação e contextualização da situação em mãos.

Projeto

A aplicação web foi elaborada utilizando o PostgreSQL⁵ como banco de dados, tendo seu *back-end* projetado com auxílio do *framework* Django⁶ que utiliza a linguagem Python⁷ como base e seu *front-end* elaborado com HTML⁸, também utilizando elementos CSS⁹ e Bootstrap¹⁰ para poder gerar a página de interface. Neste artigo, será analisado o processo de desenvolvimento o *back-end* e o banco de dados, apreciando a escolha das ferramentas, suas funcionalidades e como elas foram aplicadas nesse projeto.

Elaborando o back-end da aplicação

PostgreSQL

O Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado no projeto foi proposto em sua primeira versão pela tese de Tristão (2012) e mantido para o projeto da aplicação web, uma vez que já existia toda a estruturação para ele e que não somente é um banco de dados relacional *open-source* como também é um excelente banco voltado para ciência de dados, como apresenta Maayan (2019). Além disso, de acordo com o Developer Survey realizado em 2020, PostgreSQL era a segunda escolha dentre os desenvolvedores entrevistados para banco de dados em geral (STACK OVERFLOW, 2020).

Segundo Carvalho (2017), PostgreSQL pode ser descrito como fácil de usar tendo seus comandos SQL consistentes e padronizados, seguro por ser completamente transacional, poderoso por suportar diversos tipos de dados sofisticados, incluindo mas não limitando a JSON, XML, objetos geométricos, hierarquias, matrizes e funções em SQL, é confiável por ser amigoso tanto para a parte de desenvolvimento quanto para a parte de

⁵ <https://www.postgresql.org/>

⁶ <https://www.djangoproject.com/>

⁷ <https://www.python.org/>

⁸ <https://www.w3.org/html/>

⁹ <https://www.w3.org/Style/CSS/>

¹⁰ <https://getbootstrap.com/>

administração e também por ser rápido em sua execução ao fazer uso estratégico de indexação, além de consulta de otimização que proporciona um menor esforço operacional.

O PostgreSQL possui funcionalidades que possibilitam sua utilização em ambientes de grandes transações de dados, como por exemplo o controle de concorrência multiversionado (MVCC, em inglês) que garante a melhor manutenção dos dados caso haja múltiplas interações com o banco. Esse banco de dados *open-source* é apontado como sendo amplamente utilizado também tendo ambientes de produção que gerenciam até 4TB¹¹ de dados, o que demonstra sua capacidade de lidar com grandes quantidades de dados (CARVALHO, 2017).

Django framework

O *back-end* da aplicação é também conhecido como servidor, a parte de uma aplicação web que trata das requisições feitas pelo usuário na parte do *front-end* e contém as regras de negócio da aplicação para interagir com o banco de dados (MORRISON, 2008, APUD AMARAL E NERIS, 2015). No caso da aplicação feita para o banco *BioStringDB*, com o intuito de não somente agilizar o processo de criação como também permitir uma escalabilidade futura da aplicação e facilitar alterações caso necessárias foi utilizado o *framework* Django.

Django é um *framework* gratuito para aplicações web baseado na linguagem de programação Python, que possui um amplo suporte para integração com diversos bancos de dados. A escolha deste *framework* para criação do projeto deriva de três pontos relevantes: na tese de Tristão (2012) ele propõe uma modelagem voltada para uma versão do PostgreSQL¹² enquanto banco de dados e o Django possui nativamente suporte para tal, assim como também possui oficialmente suporte para MariaDB¹³, MySQL¹⁴, Oracle¹⁵ e SQLite¹⁶. Django mesmo possuindo uma estrutura já bem moldada para o interesse de criação de uma aplicação web, também permite alterações para melhor adequar ao projeto. Por último, mas não menos importante, Django é um *framework* baseado na linguagem Python que não somente é de fácil usabilidade como também possui bibliotecas para

¹¹ Significa Terabytes, unidade de medida informacional.

¹² <https://www.postgresql.org/>

¹³ <https://mariadb.org/>

¹⁴ <https://www.mysql.com/>

¹⁵ <https://www.oracle.com/br/database/>

¹⁶ <https://www.sqlite.org/index.html>

análise de dados que são próprias para pesquisas como a de sequenciamento genético por exemplo.

O *framework* possui uma arquitetura chamada *Model-View-Template* (MVT), bem similar a arquitetura comumente conhecida *Model-View-Controller* (MVC), porém diferenciando as funcionalidades e modos de interação entre seus respectivos módulos. Apesar de relativamente próximas, a primeira arquitetura funciona de modo em que apesar da *Model* em ambas ter a mesma funcionalidade, no MVT o *Template* possui quase a mesma funcionalidade da *View* em MVC e a *View* em MVT assemelha-se a *Controller* em MVC (PINKHAM, 2015). Resumidamente o modelo de arquitetura chamado MVT pode ser explicado por partes de forma que a *Model* descreve a estruturação dos dados a serem armazenados no banco de dados e é responsável pela manutenção e manipulação lógica dos dados para a aplicação web como um todo. A *Template* por sua vez age como a interface a ser exibida para o usuário, lidando com todas as partes estáticas da página web, bem como todo HTML, ao qual será apresentada aos usuários. Por fim as *Views* agem como a conexão entre a *Model* e os *Templates*, recebendo a solicitação do usuário e retornando os dados relativos, retornando no *Template* apropriado com os dados solicitados, como pode ser observado na **figura 01**.

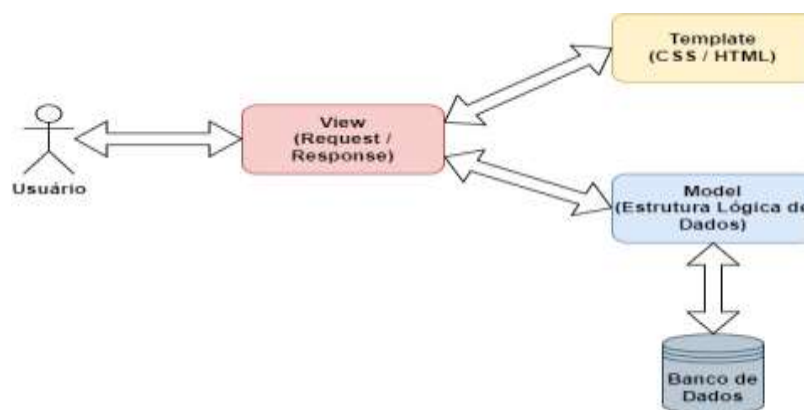


Figura 01: Arquitetura *Model-View-Template* (MVT) do Django.

Fonte: Elaborado pelo autor.

Observando que o projeto já possuía um modelo para o banco de dados, apresentando entidades, relacionamentos e funções que estariam no banco, a necessidade de adaptar os modelos no Django para atender tais especificações foi relativamente simples, porém a ferramenta dispõe da criação de funções no *back-end* e não no banco de dados. Se fez necessária então uma adaptação por parte do lado do servidor para chamar as funções existentes no banco, fazendo com que o servidor chamasse a função solicitada no

banco enviando os parâmetros inseridos no *front-end*. Apesar de ser uma adaptação e não a maneira tradicional com que a ferramenta nos oferece para realizar funções, que seria a criação das mesmas no lado do servidor, ainda é uma maneira eficaz e mais segura do que a injeção de um código SQL que realize a função.

Python

Python é a linguagem de programação base do *framework* Django, além de ser uma linguagem simples e já bem estabelecida no mercado de ciência de dados. Como menciona Eggleston (2018), 66% dos cientistas de dados reportaram usar Python todos os dias o que a torna a principal linguagem para ciência de dados. Tendo isso em mente, se torna uma linguagem propícia para o uso na pesquisa de sequenciamento genético bem como para futuras expansões, sem mencionar na possibilidade do uso de bibliotecas voltadas para análise de dados como Pandas¹⁷.

Python permite não somente um amplo uso para análise de dados como em geral é uma linguagem de programação com menos detalhes que outras linguagens, como em C#¹⁸ e Java¹⁹ o uso de ponto e vírgula ao finalizar uma linha de código que não existem em Python. Considerando principalmente as bibliotecas disponíveis que permitem uma futura expansão do projeto em questão de análise de dados e o uso do *framework* voltado para o auxílio da criação da aplicação, Python se fez preferencial na elaboração do projeto na parte do *back-end*. O seu uso dentro do *framework* definiu a estrutura lógica dos dados e como as funções dentro do banco seriam acessadas de maneira a não o comprometer, sendo criadas chamadas a essas funções do banco juntamente com o envio dos parâmetros inseridos pelo usuário na pesquisa como exibido abaixo na **figura 02**.

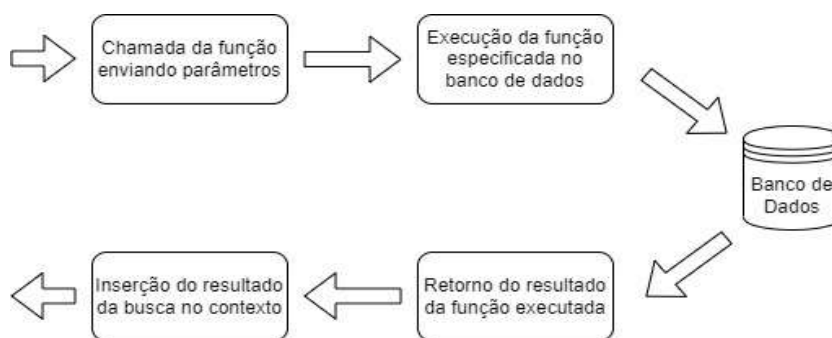


Figura 02: Caminho de chamada das funções dentro do banco.

Fonte: Elaborado pelo autor.

¹⁷ <https://pandas.pydata.org/>

¹⁸ <https://docs.microsoft.com/pt-br/dotnet/csharp/>

¹⁹ <https://www.java.com/pt-BR/>

A adaptação realizada na estruturação de chamada de função se deve pela necessidade e pré-existência das funções no banco de dados. Como o interesse era utilizar o que já estava disponível, as funções foram apenas chamadas pelo servidor enviando os parâmetros necessários para execução das mesmas, garantindo que elas fossem executadas corretamente e atendendo a necessidade de preservação das funções no banco ao invés de serem criadas funções com o Python dentro do lado do servidor, como é normalmente feito usando o *framework* Django.

DISCUSSÃO A CERCA DO USO DAS FERRAMENTAS

Ao analisar os devidos ajustes feitos por parte da elaboração do *back-end* com intuito de atender o banco de dados que já existia, o *framework* se mostrou bem maleável na medida que ainda de forma segura apresentou a possibilidade de criar chamadas de funções no banco sem prejudicar a operação da aplicação. Toda a estrutura já existente no *framework*, incluindo sua nativa compatibilidade com o PostgreSQL, permitiu com que a conectividade entre banco e servidor ocorresse suavemente.

Django se mostrou completo em, quando inicializado, já criar toda a estrutura básica para conexão entre servidor e banco de dados, necessitando apenas da alteração do nome do banco, usuário, senha e porta de conexão. Além de já montar toda a estrutura do *framework* de maneira bem intuitiva, as possibilidades que possui de adaptar ao tema de bioinformática na criação do modelo lógico permitiram com que o projeto fosse apresentado em um curto prazo de tempo. A ferramenta se mostra capaz de entregar uma robusta base para desenvolvimento do lado do servidor e integrar banco de dados, *back-end* e *front-end*.

PostgreSQL foi mantido na aplicação com intuito de preservar a proposta original, além de já possuir o modelo pronto para uso também é um banco capaz de suportar facilmente a transação de uma grande quantidade de dados, ideal para a proposta de busca de sequenciamentos genético que em diversas pesquisas trará dados em larga escala. Ao apresentar um banco robusto, PostgreSQL é capaz de lidar com tais dados de maneira eficiente e entregar pesquisas dos sequenciamentos sem pesar para o banco em termos de processamento.

Python possuindo amplas bibliotecas de análise de dados permite com que posteriormente a aplicação tenha novos desenvolvimentos em âmbitos de novas pesquisas e interpretações de dados, apresentando por exemplo comparações de dados de proteínas e

incidências de sequencias genéticas em dadas proteínas. Uma linguagem que já possui bibliotecas como Pandas para criação de tais análises acaba permitindo que sejam elaboradas diversas análises com os dados do banco com pouco esforço a ser executado na criação.

CONSIDERAÇÕES FINAIS

Diante do que foi observado até aqui sobre o funcionamento do projeto aplicado, o conjunto do banco de dados PostgreSQL e do *framework* Django mostram-se altamente eficientes na criação da aplicação web do banco de dados BioStringDB, dando ênfase na facilidade e maleabilidade na criação do projeto ao considerar as adaptações feitas. Django se apresenta como uma ferramenta poderosa que permite em sua proposta nativa a criação de sistemas capazes de entregar rapidamente um ambiente web robusto e leve. A aplicação foi capaz de apresentar o proposto e se torna uma primeira etapa em um possível processo de elaboração de um sistema mais completo de análise de dados de sequenciamentos genéticos. Analogamente a ferramenta é capaz de realizar não somente para a bioinformática, mas para diversas áreas, a construção de sistemas de pesquisas web com facilidade e escalabilidade.

1679

REFERÊNCIAS BIBLIOGRÁFICAS

AMARAL, R. A.; NERIS, V. P. A. Análise comparativa entre frameworks de front-end para aplicações web ricas visando o aproveitamento do back-end. *Tecnologias, Infraestrutura e Software*, São Carlos, v. 4, n. 1, p. 88-96, jan/abr. 2015.

CARVALHO, V. PostgreSQL: banco de dados para aplicações web modernas. Casa do Código, 2017. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=KIFVDgAAQBAJ&oi=fnd&pg=PT3&dq=postgresql+banco+de+dados&ots=1Zrql-QTYt&sig=FPO-lz2oKq5dakqvI6beUFGV9QY#v=onepage&q=postgresql%20banco%20de%20dados&f=false>. Acesso em: 10 maio 2022

CYTOSCAPE. What is Cytoscape? Disponível em: https://cytoscape.org/what_is_cytoscape.html. Acesso em: 10 maio 2022.

EGGLESTON, L. Course Report. How is Python Used for Data Science? 2021. Disponível em: <https://www.coursereport.com/blog/how-is-python-used-for-data-science-metis#:~:text=In%202018%2C%2066%25%20of%20data,one%20language%20for%20data%20science.> Acesso em: 16 maio 2022.

HEINE, A. A. P.; SANTOS, E. P. S.; LIMA, M. F.; LIFSCHITZ, S. Aplicação Web de Processamento, Clustering e Visualização de Grafos de Genes de Workflows Científicos. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBBB), 36. , 2021, Rio de Janeiro. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 56-62.

IBM. Linguagem de consulta estruturada (SQL). Disponível em: <https://www.ibm.com/docs/en/db2/10.5?topic=fundamentals-sql>. Acesso em: 19 maio 2022

MAAYAN, G. D. Why You Should Learn PostgreSQL for Data Science. 2019. Disponível em: <https://www.dataversity.net/why-you-should-learn-postgresql-for-data-science/>. Acesso em: 10 maio 2022.

MORRISON, M. Head First JavaScript. Sebastopol: O'Reilly Media, 2008.

PINKHAM, A. Django Unleashed.: Sams, 2015. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=gCoBCwAAQBAJ&oi=fnd&pg=PT17&dq=django+mvc&ots=GO_QEY-MU8&sig=bHivJ4oHbfMovMV_aBKmEypW4Kg#v=onepage&q=django%20mvc&f=false. Acesso em: 10 maio 2022.

STACK OVERFLOW. Developer Survey. 2020. Disponível em: <https://insights.stackoverflow.com/survey/2020#technology-web-frameworks-professional-developers2>. Acesso em: 12 maio 2022.

STRING. About String Database. Disponível em: https://string-db.org/cgi/about?footer_active_subpage=content. Acesso em: 12 maio 2022.

TRISTÃO, C. “Uma Abordagem para Modelar, Armazenar e Acessar Sequências Biológicas”, Tese de Doutorado, Departamento de Informática, PUC-Rio, 2012.