**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**Faculdade de Tecnologia de Jundiaí – “Deputado Ary Fossen”**

**Curso Superior de Tecnologia em Gestão da Tecnologia da Informação**

Allan Vitor Fontes dos Santos

Artur Smaniotto Mesquita de Oliveira

Gabriel Henrique Pereira Leite

**MINHA BIBLIOTECA GAMER: PLATAFORMA DE REGISTRO E AVALIAÇÃO DE JOGOS DE VIDEOGAME**

**Jundiaí**

**2024**

**Allan Vitor Fontes dos Santos**

**Artur Smaniotto Mesquita de Oliveira**

**Gabriel Henrique Pereira Leite**

**MINHA BIBLIOTECA GAMER: PLATAFORMA DE REGISTRO E AVALIAÇÃO DE JOGOS DE VIDEOGAME**

Projeto Interdisciplinar II - III apresentado à Faculdade de Tecnologia de Jundiaí - “Deputado Ary Fossen”, sob a orientação do Professor Wagner Luiz De Andrade Junior.

**Jundiaí**

**2024**

Lista de Ilustrações

[Figura 1 - Barra de Estilo do Word – Citação 17](#_Toc145431998)

[Figura 2 - Estrutura do TG 21](#_Toc145431999)

[Figura 3 - Quadro de Abreviaturas das Titulações 22](#_Toc145432000)

[Figura 4 - Quadro de Cursos da FATEC Jundiaí em Inglês 24](#_Toc145432001)

[Figura 5 - Barra de Estilos do Word – Fechada 25](#_Toc145432002)

[Figura 6 - Exemplo de formatação para tabelas em mais de uma página 27](#_Toc145432003)

[Figura 7 - Barra de Estilos do Word – Fechada 27](#_Toc145432004)

[Figura 8 - Barra de Estilos do Word – Aberta 28](#_Toc145432005)

[Figura 9 - Marcadores e Numeração do Word 29](#_Toc145432006)

LISTA DE TABELAS

[Tabela1 - Exemplo de Formatação de Tabela 26](#_Toc145432013)

Lista de Abreviaturas e Siglas

**ABNT**  Associação Brasileira de Normas Técnicas

**FATEC** Faculdade de Tecnologia

**IBGE** Instituto Brasileiro de Geografia e Estatística

**TG** Trabalho de Graduação

Lista de Símbolos

dab Distância euclidiana

Sumário

[1 Introdução 7](#_Toc178101445)

[2 Revisão Bibliográfica 8](#_Toc178101446)

[2.1 Ciclo de vida de desenvolvimento de Software (SLDC) principais modelos 8](#_Toc178101447)

[2.2 Requisitos funcionais e não funcionais 8](#_Toc178101448)

[2.3 Implementação de software - Linguagem de programação, bibliotecasm 9](#_Toc178101449)

[2.4 Tipos de dados armazenados em tabelas de bancos de dados 10](#_Toc178101450)

[2.5 SGBDs - O que é e quais os mais usados? 10](#_Toc178101451)

[Referências 12](#_Toc178101452)

# Introdução

A crescente popularidade dos videogames tem gerado um interesse significativo entre jogadores de todas as idades em registrar e compartilhar suas experiências. Com o objetivo de atender a essa demanda, nossa pesquisa se propõe a desenvolver uma aplicação inovadora focada no cadastro e na informação de jogos de videogame. Esta plataforma permitirá que os usuários cataloguem, avaliem e compartilhem suas experiências com jogos, oferecendo informações detalhadas como desenvolvedores, datas de lançamento e plataformas disponíveis.

Além disso, a aplicação permitirá a criação de listas personalizadas, onde os jogadores poderão organizar os jogos que já jogaram, desejam jogar ou estão jogando atualmente. A justificativa para o desenvolvimento desta pesquisa reside no desejo dos jogadores de ter um espaço dedicado para registrar suas experiências e buscar informações sobre seus jogos favoritos. Nosso grupo acredita que, ao proporcionar um ambiente onde os jogadores possam compartilhar suas opiniões e informações relevantes, contribuiremos para a comunidade gamer de maneira significativa.

# Revisão Bibliográfica

## Ciclo de vida de desenvolvimento de Software (SLDC) principais modelos

O SDLC é uma metodologia para projetar, construir e manter sistemas de software, com vários modelos, incluindo Waterfall, Spiral e Incremental. Cada modelo possui fases sequenciais que devem ser seguidas para desenvolver soluções de software. O artigo examina os pontos fortes e fracos dos modelos Waterfall, Spiral e Incremental.

Modelo em Cascata: O Modelo em Cascata é um dos mais antigos e conhecidos, caracterizado por suas fases sequenciais: requisitos, design, codificação, testes e manutenção. É ideal para projetos que exigem controle de qualidade rigoroso, pois envolve documentação detalhada. Cada fase deve ser concluída antes de passar para a próxima.

Modelo Espiral: O Modelo Espiral combina design e prototipagem, focando na avaliação e minimização de riscos. Os projetos são divididos em segmentos menores, permitindo mudanças durante o desenvolvimento. Cada iteração resulta em um protótipo que incorpora feedback e funcionalidades adicionais, até a conclusão do produto.

Modelo Iterativo e Incremental: Este modelo também combina elementos do modelo em cascata de forma iterativa, permitindo a entrega de incrementos do software. O desenvolvimento é gradual, onde cada versão adiciona novas funcionalidades até que todas as especificações sejam atendidas.

O artigo conclui que cada modelo SDLC tem suas vantagens e desvantagens, sendo importante escolher o mais adequado conforme as necessidades do projeto. Futuros trabalhos visam explorar mais modelos e simulações.

## Requisitos funcionais e não funcionais

## Implementação de software - Linguagem de programação, bibliotecasm

Para Leite (2006) um software é um programa de computador , já um programa é um conjunto de algoritmos que são implementados por uma linguagem de programação, com os dados do software sendo armazenados por um banco de dados, portanto a implementação de um software necessita de uma linguagem de programação consistente, eficiente e que faça a conexão com a base de dados. Nesse contexto, o Python é uma boa escolha de linguagem de programação, sendo código aberto e capaz de implementar e desenvolver aplicativos complexos, ao mesmo tempo que mantém uma sintaxe simples e direta.

Segundo PRESSMAN (1995) o software segue instruções que quando executadas, devem ter o desempenho desejado, possui estrutura de dados que possibilitem os programas manipularem a informação e que tenham documentação que descrevam a operação, logo, o Python é uma excelente opção, visto que, devido a sua sintaxe simples, diversos engenheiros de software conseguem desenvolver aplicações com menos código comparado a outras linguagens de programação, tendo como vantagens a otimização de recursos de desenvolvimento, o amplo suporte com a documentação e a grande quantidade de frameworks e bibliotecas.

Um dos grandes frameworks utilizados pelo Python e que possui uma grande quantidade de bibliotecas é o Django, uma ferramenta full-stack e de código aberto muito utilizado para criação de aplicativos web e back-end. É amplamente utilizado pelo seu desenvolvimento rápido, versatilidade, interface administrativa e o grande apoio da comunidade na melhoria do framework.

Outro grande framework do Python é o Tornado, sendo de código aberto e que possui bibliotecas de rede assíncrona de alto desempenho, utilizado para desenvolvimento de aplicações web. Reconhecido pela sua facilidade de aprendizagem, otimização em aplicações que requerem muitos clientes simultâneos e seu núcleo pequeno e extensível, é uma excelente opção para o desenvolvimento web.

Por fim, a implementação de software utilizando o Python é uma excelente escolha, visto que, possui uma grande quantidade de bibliotecas e frameworks, a sua grande versatilidade com a possibilidade de automação de processos e sua alta flexibilidade, tornam o Python uma grande linguagem de programação no processo de desenvolvimento de software.

## Tipos de dados armazenados em tabelas de bancos de dados

**MySQL (**Versão** 8.0)**

Cada coluna em uma tabela de banco de dados precisa ter um nome e um tipo de dado.

Um desenvolvedor SQL deve decidir que tipo de dados serão armazenados dentro de cada coluna ao criar uma tabela. O tipo de dados é uma diretriz para SQL entender que tipo de dados é esperado dentro de cada coluna, e também identifica como SQL irá interagir com os dados armazenados.

No MySQL, há três tipos principais de dados: string, numérico e data e hora.

**Tipos de dados de string**

| Tipo de dado | Descrição |
| --- | --- |
| CHAR(tamanho) | Uma string de comprimento FIXO (pode conter letras, números e caracteres especiais). O parâmetro de tamanho especifica o comprimento da coluna em caracteres - pode ser de 0 a 255. O padrão é 1 |
| VARCHAR(tamanho) | Uma string de comprimento VARIÁVEL (pode conter letras, números e caracteres especiais). O parâmetro tamanho especifica o comprimento máximo da coluna em caracteres - pode ser de 0 a 65535 |
| BINARY(tamanho) | Igual a CHAR(), mas armazena strings de bytes binários. O parâmetro tamanho especifica o comprimento da coluna em bytes. O padrão é 1 |
| VARBINARY(tamanho) | Igual a VARCHAR(), mas armazena strings de bytes binários. O parâmetro tamanho especifica o comprimento máximo da coluna em bytes. |
| TINYBLOB | Para BLOBs (Binary Large OBjects). Comprimento máximo: 255 bytes |
| TINYTEXT | Contém uma string com um comprimento máximo de 255 caracteres |
| TEXT(tamanho) | Contém uma string com um comprimento máximo de 65.535 bytes |
| BLOB(tamanho) | Para BLOBs (Binary Large OBjects). Armazena até 65.535 bytes de dados |
| MEDIUMTEXT | Contém uma string com um comprimento máximo de 16.777.215 caracteres |
| MEDIUMBLOB | Para BLOBs (Binary Large OBjects). Armazena até 16.777.215 bytes de dados |
| LONGTEXT | Contém uma string com um comprimento máximo de 4.294.967.295 caracteres |
| LONGBLOB | Para BLOBs (Binary Large OBjects). Armazena até 4.294.967.295 bytes de dados |
| ENUM(val1, val2, val3, ...) | Um objeto string que pode ter apenas um valor, escolhido de uma lista de valores possíveis. Você pode listar até 65535 valores em uma lista ENUM. Se um valor for inserido que não esteja na lista, um valor em branco será inserido. Os valores são classificados na ordem em que você os insere |
| SET(val1, val2, val3, ...) | Um objeto string que pode ter 0 ou mais valores, escolhidos de uma lista de valores possíveis. Você pode listar até 64 valores em uma lista SET |

**Tipos de dados numéricos**

| Tipo de dado | Descrição |
| --- | --- |
| BIT(tamanho) | Um valor de tipo bit. O número de bits por valor é especificado em tamanho. O parâmetro tamanho pode conter um valor de 1 a 64. O valor padrão para tamanho é 1. |
| TINYINT(tamanho) | Um inteiro muito pequeno. O intervalo assinado é de -128 a 127. O intervalo não assinado é de 0 a 255. O parâmetro tamanho especifica a largura máxima de exibição (que é 255) |
| BOOL | Zero é considerado falso, valores diferentes de zero são considerados verdadeiros. |
| BOOLEAN | Igual a BOOL |
| SMALLINT(tamanho) | Um pequeno inteiro. O intervalo assinado é de -32768 a 32767. O intervalo não assinado é de 0 a 65535. O parâmetro tamanho especifica a largura máxima de exibição (que é 255) |
| MEDIUMINT(tamanho) | Um inteiro médio. O intervalo assinado é de -8388608 a 8388607. O intervalo não assinado é de 0 a 16777215. O parâmetro tamanho especifica a largura máxima de exibição (que é 255) |
| INT(tamanho) | Um inteiro médio. O intervalo assinado é de -2147483648 a 147483647. O intervalo não assinado é de 0 a 4294967295. O parâmetro tamanho especifica a largura máxima de exibição (que é 255) |
| INTEGER(tamanho) | Igual a INT(tamanho) |
| BIGINT(tamanho) | Um inteiro grande. O intervalo assinado é de -9223372036854775808 a 9223372036854775807. O intervalo não assinado é de 0 a 18446744073709551615. O parâmetro tamanho especifica a largura máxima de exibição (que é 255) |
| FLOAT(tamanho, d) | Um número de ponto flutuante. O número total de dígitos é especificado em tamanho. O número de dígitos após o ponto decimal é especificado no parâmetro d. Esta sintaxe está obsoleta no MySQL 8.0.17 e será removida em futuras versões do MySQL |
| FLOAT(p) | Um número de ponto flutuante. O MySQL usa o valor p para determinar se deve usar FLOAT ou DOUBLE para o tipo de dados resultante. Se p for de 0 a 24, o tipo de dados se torna FLOAT(). Se p for de 25 a 53, o tipo de dados se torna DOUBLE() |
| DOUBLE(tamanho, d) | Um número de ponto flutuante de tamanho normal. O número total de dígitos é especificado em tamanho. O número de dígitos após o ponto decimal é especificado no parâmetro d |
| DOUBLE PRECISION(tamanho, d) |  |
| DECIMAL(tamanho, d) | Um número de ponto fixo exato. O número total de dígitos é especificado em tamanho. O número de dígitos após o ponto decimal é especificado no parâmetro d. O número máximo para tamanho é 65. O número máximo para d é 30. O valor padrão para tamanho é 10. O valor padrão para d é 0. |
| DEC(tamanho, d) | Igual a DECIMAL(tamanho,d) |

**Tipos de dados de data e hora**

| Tipo de dado | Descrição |
| --- | --- |
| DATE | Uma data. Formato: AAAA-MM-DD. O intervalo suportado é de '1000-01-01' a '9999-12-31’ |
| DATETIME(fsp) | Uma combinação de data e hora. Formato: AAAA-MM-DD hh:mm:ss. O intervalo suportado é de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'. Adicionar DEFAULT e ON UPDATE na definição da coluna para obter inicialização automática e atualização para a data e hora atuais |
| TIMESTAMP(fsp) | Um timestamp. Os valores TIMESTAMP são armazenados como o número de segundos desde a época do Unix ('1970-01-01 00:00:00' UTC). Formato: AAAA-MM-DD hh:mm:ss. O intervalo suportado é de '1970-01-01 00:00:01' UTC a '2038-01-09 03:14:07' UTC. A inicialização e atualização automáticas para a data e hora atuais podem ser especificadas usando DEFAULT CURRENT\_TIMESTAMP e ON UPDATE CURRENT\_TIMESTAMP na definição da coluna |
| TIME(fsp) | Um horário. Formato: hh:mm:ss. O intervalo suportado é de '-838:59:59' a '838:59:59’ |
| YEAR | Um ano em formato de quatro dígitos. Valores permitidos em formato de quatro dígitos: 1901 a 2155 e 0000. O MySQL 8.0 não suporta ano em formato de dois dígitos. |

## SGBDs - O que é e quais os mais usados?

Segundo Elmasri (2018, p. 4) um banco de dados é uma coleção de dados que podem ser relacionados. Nessa coleção estão registros que podem ser gravados, e em conjunto se tornam um volume de várias dados e informações, logo, o domínio sobre os sistemas Gerenciadores de Banco de Dados (SGBDs) é um diferencial entre os especialistas e das empresas, visto que, aprimorar e criar inovações relacionadas à segurança e armazenamento de dados são tarefas primordiais no processo de controle de um software com um banco de dados.

SGBDs são softwares que gerenciam o armazenamento, organização e a manipulação do fluxo de dados em um banco de dados. Por meio de uma interface, onde os usuários e os dados se comunicam, é possível haver a manipulação e análise dos dados armazenados. Os softwares SGBDs garantem a integridade dos dados, isso é, a consistência das informações armazenadas, qualidade e a eficiência da gestão dos dados de determinadas organizações.

Aplicações como o ERP (sistemas de gestão empresarial), CRM (sistemas de gestão de relacionamento com clientes), RH (sistemas de gestão de recursos humanos) e sistemas de e-commerce utilizam softwares SGBDs para armazenar e gerenciar as informações necessárias de aplicações específicas, garantindo integridade e segurança.

Devido ao aumento e volume de dados, algumas dificuldades podem aparecer

como problemas em ter uma melhor eficiência na leitura de registros em banco de dados em grande volume de dados e dificuldade em facilitar a leitura e gravação de registros (TIWARI, 2011). Portanto, os sistemas SGBDs contam com ferramentas para garantir otimização e eficiência no software e banco de dados, exemplos como segurança (conduzindo regras e um sistema de segurança robusto), possibilidades de backups e esquematizações (cruzando resultados em mecanismos de buscas e gerando maiores possibilidades de tabelas).

Existem diversas maneiras de classificar os bancos de dados, tais como os SGBDS Relacionais, organizando os dados em tabelas e podendo relacioná-las por meio de chaves primárias e estrangeiras (exemplos como MySQL, PostgreSQL e Microsoft SQL Server) , SGBDs Orientado a objetos, com os dados representados como objeto, SGBDs Hierárquicos (organizando os dados em formatos de uma estrutura de árvore) , e os SGBDs NoSQL, que abrangem banco de dados não relacionais e projetados para trabalhar com grandes volumes de dados.

Referências

Elemento