**Atividades Práticas Supervisionadas (APS)**

**Java com MySQL**

**Alunos: Matheus Ferreira De Araújo, Bruno Ferreira Da Silva e Lucas Gabriel Francelino.**

**RA: N910810, N546JA6, N7426C0.**

**Introdução**

O desenvolvimento das novas tecnologias proporciona, cada vez mais, que as máquinas, e neste caso os computadores, tenham um papel mais ativo na nossa sociedade. Em todo o lado se verifica uma passagem de testemunho do homem para o computador nas tarefas mais variadas. No caso da comunicação, os computadores desempenham um papel preponderante, facilitando-a incrivelmente, podendo ser tanto uma comunicação individual entre duas pessoas ou com várias espalhadas por todo o planeta que seria o caso das redes sociais.

As redes sociais em si são sites e aplicativos usados por pessoas e organizações que se conectam com clientes, familiares, amigos e pessoas que compartilham seus interesses em comum, uma poderosa ferramenta de análise do comportamento do consumidor e é por meio dos seus algoritmos que é possível identificar os interesses do público e o seu comportamento nas variadas páginas da web.

Esses algoritmos utilizados nesses sites têm a mesma base da linguagem utilizada neste trabalho, como por exemplo JAVA e SQL.

JavaScript é uma linguagem de programação que permite a você implementar itens complexos em páginas web, toda vez que uma página da web faz mais do que simplesmente mostrar a você informação estática, mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados etc., você pode apostar que o JavaScript provavelmente está envolvido. É a terceira camada do bolo das tecnologias padrões da web, duas das quais ([HTML](https://developer.mozilla.org/pt-BR/docs/Learn/HTML) e [CSS](https://developer.mozilla.org/pt-BR/docs/Learn/CSS)).  Uma linguagem de programação que permite a você criar conteúdo que se atualiza dinamicamente, controlar multimídias, imagens animadas, e tudo o mais que há de interessante. Ok, não tudo, mas é maravilhoso o que você pode efetuar com algumas linhas de código JavaScript.

De uma forma geral, SQL é uma linguagem de programação feita para armazenar e processar informações de um **banco de dados relacional** (baseado em tabelas). Quando você vai desenvolver alguma aplicação, um website, por exemplo, as informações de cadastro dos usuários precisam ficar armazenadas em algum lugar e há diversas formas de fazer isso. Elas podem ser salvas em um arquivo de texto, planilhas ou em um banco de dados.

Usando esses dois tipos de linguagens citados acima, fará a composição deste trabalho semestral que consiste em criar um cronômetro que calcule o tempo que levará para um carrinho de controle remoto percorrer e finalizar uma volta. Na parte de algoritmos em Javascript será realizado a tela principal de interface do relógio marcador, onde vai ser colocado as funções de start, stop, segunda volta, primeira e etc..

Já utilizando o sistema SQL, efetuamos a conexão do programa criado em Java acima com o banco de dados para que seja armazenado depois de calculado o tempo da primeira volta, o tempo da segunda e o da terceira se for necessário, com esses dados salvos em nosso sistema conseguiremos entregar a proposta do trabalho em resultados de cada volta separadamente e o tempo total te todos os percursos percorridos somados em um só.

**Referencias Teóricas**

Telemetria é um processo que transmite e mede dados à distância. No geral, a telemetria envolve a utilização de sensores ou dispositivos eletrônicos que capturam dados em tempo real sobre um determinado objeto ou sistema e nisso transmite esses dados para uma estação base ou centro de controle.

Esses dados coletados pela telemetria podem incluir informações como temperatura, pressão, velocidade, localização, entre outras, dependendo do que está sendo monitorado.

Os dados que são coletados pela telemetria podem ser usados ​​para monitorar o desempenho de um sistema, identificar problemas ou falhas, ou tomar decisões controladas com base nas informações coletadas.

Dependendo do monitoramento que for utilizada na telemetria, as linguagens que podem ser utilizadas são C++,Python,Java Script, dentre outros tipos de linguagens.

O uso da telemetria no dia a dia, podemos ter como exemplo os caminhões que possuem esse tipo de monitoramento, onde é possível ver tudo com o que o caminhão faz desde a hora que é ligado. Nos caminhões podemos saber a quilometragem percorrida pelo veículo, localização do veículo, velocidade instantânea, consumo de combustível, frenagens bruscas e tempos de parada do veículo.

Outras formas utilizadas em veículos são a identificação do motorista, o monitoramento de RPMs, a temperatura do motor, a ignição ligada / desligada, o uso do cinto de segurança e também as condições da estrada.

Com a telemetria também é possível monitorar os sinais de espera de pacientes à distância, permitindo que médicos e enfermeiros tenham acesso a informações importantes em tempo real e possam tomar decisões mais rápidas e precisas.

Na vegetação também é uma outra forma de monitoramento possível com a telemetria, no qual é pode ser monitorada a temperatura do ambiente, saber como está a saúde da planta, dentre outros possíveis monitoramentos.

Em nosso trabalho, nós utilizamos a telemetria, buscando monitorar uma corrida de carrinhos de controle remoto, monitorando com um cronômetro as voltas que os veículos fazem, nisso marcando a primeira volta, segunda volta e a volta final.

Nessa forma de monitoramento que nós estamos utilizando, também serve para qualquer tipo de veículo, como até em drones. Como o monitoramento é em forma de cronômetro, basta saber utilizar da tecnologia naquilo no qual está buscando medir e monitorar.

**Desenvolvimento**

Primeiro vamos falar sobre a linguagem usado no trabalho.

A linguagem java foi criada e comercializada pela Sun Microsystems em 1995 e atualmente mantida pela Oracle. Ela é definida como uma linguagem de programação orientada a objetos que é amplamente usada para o desenvolvimento de sites e aplicativos.

Java: o que é?

Apesar de parecer um bicho de sete cabeças, você provavelmente já teve contato com essa linguagem de programação e nem imagina.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Uma linguagem de programação Java permite que você escreva instruções para serem executadas pelos computadores. Essas instruções construídas em Java são chamadas de software, popularmente conhecido como programa de computador.

Em resumo, software é o que permite a você controlar o hardware. Quando falamos de computador, devemos ter em mente que existem desde computadores grandiosos, até microcomputadores.

A principal função do Java é construir aplicações em rede, como jogos e programas. Ou seja, um programador Java consegue projetar softwares que podem ser operados e distribuídos em múltiplas plataformas, como mac, windows e linux sem a necessidade de modificá-los e, inclusive, sem pensar na arquitetura da máquina.

Java: onde é utilizado?

Apesar de grande parte das pessoas não saberem, o Java em qualquer lugar e atualmente está presente em muitas coisas do nosso cotidiano, podendo encontrá-lo em:

Anéis: o anel smart ring é um clássico que “roda” programas de java no seu interior, lhe dando diversas opções de gestos para acessar um app, trocar de música ou bloquear a tela do seu smartphone;

Supercomputadores: responsável pelos trilhões de cálculos por segundo feitos os supercomputadores utilizam a linguagem java;

GPS: o próprio Google Maps foi criado utilizando a linguagem de programação Java;

Cartão de crédito: os cartões da Visa, por exemplo, utilizam a linguagem de programação Java;

Terminais lotéricos: ao realizar o pagamento de um boleto, você está utilizando a linguagem Java para efetuar essa ação;

Jogos: o java auxilia bastante no desenvolvimento de jogos, devido à sua versatilidade e tamanho de conteúdo.

Além disso, podemos criar aplicações embarcadas através do Java Embedded, aplicações mobile para Android através do Android SDK, aplicações desktop com o Java SE e ainda aplicações web e corporativas de larga escala através do Java EE.

Java: por que utilizar?

A escolha da linguagem de programação é uma decisão crucial na carreira de um programador, afinal, existem inúmeras linguagens de programação e algumas foram construídas para propósitos específicos, enquanto outras foram desenvolvidas única e exclusivamente para desenvolver páginas na internet.

Tabela

Descrição gerada automaticamente  
Java é uma das poucas linguagens de programação que você pode utilizar para construir praticamente qualquer coisa.

Um dos principais fatores que levam as pessoas a escolherem esse tipo de linguagem é a facilidade de aprender, além de ser extremamente poderosa, permitindo a você fazer vários tipos de programas com diferentes propósitos.

Além disso, a linguagem de programação Java é madura, ou seja, você não corre o risco de estar escolhendo uma linguagem que possa desaparecer daqui a pouco tempo. O Java está em constante evolução e se mantém presente nas primeiras posições no ranking de linguagem de programação mais populares e promissoras do mundo.

Além dos motivos citados acima, a demanda por profissionais qualificados está crescendo a cada ano e, se você deseja entrar no mercado de TI, tornar-se um programador Java será uma boa decisão para garantir empregabilidade e um salário promissor.

De acordo com o Glassdoor, o salário médio de um desenvolvedor de Java no Brasil é de R$ 60.000,00 por ano e R$ 8.000 por mês para profissionais com cargo de desenvolvedor sênior.

Agora falaremos sobre o cronometro que é a ideia do projeto

História do cronometro

No século XVIII fora construído o primeiro aparelho que serviria como ponto de partida para o cronometro que se tem hoje. Naquela época, o instrumento se tratava de algo que servira para auxiliar em navegações. Por meio desse aparelho os navegadores conseguiriam traçar rotas mais seguras e com maior precisão.

Mas depois que eles foram aceitos como aparelhos que auxiliariam em processos marítimos, os mesmos passara por melhorias, se tornando nos cronômetros que hoje se possui, também se tornando modelos mais compactos.

Sobre o uso desse aparelho

Teclado de computador com texto preto sobre fundo branco

Descrição gerada automaticamente  
É comum que os nadadores tenham cronômetros para marcarem o tempo que percorrem até o final da piscina, por exemplo, com o treinador marcando esse tempo. Aqueles que praticam mergulho podem ainda querer marcar o tempo que conseguem permanecer debaixo d’água.

Também se faz uso do cronometro digital, o qual possui um display onde é marcado o tempo. Esse e comum de ser usado quando se pretende preparar uma receita e é necessário marcar o tempo para cozimento, para algo assar, etc. Alguns pratos necessitam que seja contado o tempo para que o prepare seja perfeito.

Em outros casos, o cronômetro é usado para processos e procedimentos em laboratórios onde contar o tempo é crucial.

Agora falaremos sobre o banco de dados que foi utilizado.

O Que É MySQL?

Logotipo

Descrição gerada automaticamente  
Uma empresa sueca chamada MySQL AB desenvolveu o MySQL em 1994. Então, a companhia norte-americana Sun Microsystems obteve controle total do software ao comprar a MySQL AB em 2008.

Já em 2010, a gigante Oracle, também norte-americana, por sua vez comprou a Sun Microsystems, e o MySQL MySQL tem sido da Oracle desde então.

Quanto a sua definição, MySQL é um Banco de Dados relacional (RDBMS – Relational Database Management Systems) com um modelo de cliente-servidor.

RDBMS é um software de código aberto ou serviço usado na criação e gerenciamento de bancos de dados baseados no modelo relacional. Agora vamos analisar cada termo.

Banco de Dados

Um banco de dados nada mais é do que uma coleção de dados estruturados. Pense em uma selfie: você aperta o botão para tirar uma foto de você mesmo.

A foto é o dado, enquanto a galeria do seu celular onde a foto ficou armazenada é o banco de dados.

Banco de dados é um um local onde dados são armazenados e gerenciados. A palavra “relacional” significa que os dados armazenados estão organizados em tabelas.

Cada tabela está relacionada de alguma maneira. Caso o software não suporte o modelo relacional, então chamamos DBMS.

Código Aberto

Código aberto significa que você tem a liberdade de usar e modificar. Qualquer pessoa pode instalar o software. Você também pode aprender sobre o código e modificá-lo para melhor atender suas necessidades.

Mas o GPL (GNU – Licença Pública Geral) determina o que é possível ser feito de acordo com as condições. A versão comercializável da licença também está disponível caso você precise de mais flexibilidade e suporte avançado.

Modelo Cliente-Servidor

Computadores que rodam softwares de RDBMS são chamados clientes. Sempre que precisam de dados, eles se conectam a um servidor RDBMS.

MySQL é um dos muitos clientes RDBMS disponíveis. RDBMS e MySQL são muitas vezes confundidos como sendo a mesma coisa devido a popularidade do MySQL.

Algumas aplicações de renome, como Facebook, Twitter, YouTube, Google e Yahoo! utilizam MySQL para o armazenamento de dados.

Mesmo que tenha sido inicialmente criada para uso ilimitado, atualmente é ele compatível com muitas plataformas importantes como Linux, MacOS, Microsoft Windows e Ubuntu.

SQL

É também importante saber a diferença entre SQL e MySQL.

O MySQL é uma marca de software RDBMS mais populares que utiliza o modelo cliente-servidor.

Mas como o cliente e o servidor se comunicam em um ambiente RDBMS? Eles usam uma linguagem específica – Structured Query Language (SQL).

Se você encontrar outros nomes que também possuam SQL, como PostgreSQL e Microsoft SQL Server, provavelmente são outras marcas que utilizam a sintaxe SQL. Um software RDBMS é normalmente escrito em outra linguagem de programação, mas sempre utiliza SQL como linguagem primária para comunicação com o banco de dados. O próprio MySQL é escrito em C e C++. Pense em todos os países colonizados pela Inglaterra, cada um tem sua história e costumes próprios, mas todos falam inglês. O cientista da computação Ted Codd desenvolveu SQL no início dos anos 70 com um IBM baseado no modelo relacional. A linguagem se tornou mais popular em 1974 substituindo algumas linguagens ultrapassadas, como ISAM e VISAM.

Colocando a história de lado, o SQL informa ao servidor o que deve ser feito com o dado. Ele desempenha as seguintes funções:

Data Query. Solicitar informações específicas de um banco de dados.

Data Manipulation. Adicionar, excluir, mudar, ordenar e outras operações para modificar os dados, valores ou visuais.

Data Identity. Definir tipos de dados. Ex: mudar dados numéricos para integrais. Isso também inclui definir a relação das tabelas no banco de dados.

Data Access Control. Fornece técnicas de segurança para proteger dados, incluindo quem pode visualizar ou usar as informações armazenadas no banco de dados.

Como Funciona o MySQL?

modelo cliente e servidor do banco de dados mysql

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média  
A imagem explica a estrutura básica de um modelo cliente-servidor. Um ou mais dispositivos (clientes) conectam com o servidor através de uma rede específica.

Cada cliente pode fazer a solicitação através de uma interface gráfica do usuário (IGU) em suas telas. E o servidor vai produzir o melhor resultado, desde que ambos entendam as instruções. Sem entrar muito nos méritos técnicos, os principais processos que acontecem em um ambiente MySQL são os mesmos.

O MySQL cria um banco de dados para armazenamento e manipulação de dados, definindo a relação de cada tabela.

Clientes podem fazer solicitações digitando comandos SQL específicos no MySQL.

A aplicação do servidor responde com a informação solicitada fazendo aparecer no cliente.

Quanto ao cliente, geralmente se enfatiza qual IGU MySQL é utilizado. Quanto mais leve e simples de usar for o GUI, mais eficiente será.

Alguns dos IGU mais populares são MySQL WorkBench, SequelPro, DBVisualizer e Navicat DB Admin Tool.

Alguns deles são gratuitos, enquanto outros são pagos. Alguns rodam exclusivamente no MacOS, enquanto outros são compatíveis com os maiores sistemas operacionais.

Cada cliente deve escolher um IGU de acordo com suas necessidades. Para gerenciamento de bancos de dados web, incluindo sites WordPress, a opção mais comum é o phpMyAdmin.

Por que o MySQL é tão popular?

ranking de popularidade do mysql

O Servidor MySQL não é o único (R)DBMS no mercado, mas é um dos mais populares, ficando atrás apenas do Oracle Database.

E isso apenas em parâmetros importantes, como o número de menções em pesquisas, perfis profissionais no LinkedIN e frequência de debates em fóruns técnicos.

O fato de muitas empresas de tecnologia gigantes utilizam seus serviços é um dos aspectos que mais impactam em sua popularidade.

Abaixo estão algumas das suas principais características.

Flexibilidade e Facilidade de Uso

Você pode modificar o código-fonte para atender suas expectativas e você não precisa pagar nada por isso. Além de ter a opção de upgrade para uma versão comercial. O processo de instalação é relativamente simples, não levando mais que 30 minutos.

Alto Desempenho

Um vasto array de servidores cluster estão disponíveis no MySQL. Quer você esteja armazenando uma quantidade gigante de dados ecommerce ou realizando atividades de inteligência pesadas, MySQL garante um alto desempenho e velocidade.

Um Padrão da Indústria

O MySQL tem sido usado por indústrias há muito tempo, o que significado que há abundância de recursos para desenvolvedores experientes. Sempre que precisar de alguma aplicação não vai ser difícil encontrar desenvolvedores que estejam dispostos a trabalhar com MySQL.

Segurança

Seus dados devem ser a principal preocupação ao escolher um software RDBMS. Com seu Sistema de Privilégios de Acesso e Gerenciamento de Contas de Usuários, MySQL possui um alto nível de segurança. Verificação auto-hospedada e criptografia de senhas estão disponíveis.

Falaremos também sobre o ambiente utilizado para desenvolvimento do codigo

Net Beans

Interface gráfica do usuário

Descrição gerada automaticamente  
História

O NetBeans foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente. Primeiramente, o nome do projeto era Xelfi, em alusão ao Delphi, pois a pretensão deste projeto era ter funcionalidades semelhantes aos IDEs então populares do Delphi que eram mais atrativas por serem ferramentas visuais e mais fáceis de usar, porém com o intuito de ser totalmente desenvolvido em Java.

Em 1999, o projeto já havia evoluído para uma IDE proprietário, com o nome de NetBeans DeveloperX2, nome que veio da ideia de reutilização de componentes que era a base do Java. Nessa época, a empresa Sun Microsystems havia desistido de sua IDE Java Workshop e, procurando por novas iniciativas, adquiriu o projeto NetBeans DeveloperX2 incorporando-o a sua linha de softwares.

Por alguns meses, a Sun mudou o nome do projeto para Forte for Java e o manteve por um bom tempo como software proprietário. Porém, em junho de 2000, a Sun disponibilizou o código fonte do IDE NetBeans, tornando-o uma plataforma OpenSource. Mais tarde, com a aquisição da Sun Microsystems pela Oracle em 2010, tornou-se parte da Oracle. Ao longo de sua história na Sun Microsystems e Oracle, o NetBeans foi gratuito e de código aberto; e foi alavancado pelo seu patrocinador como um mecanismo para impulsionar o ecossistema Java para a frente.

Em 2016, durante o JavaOne, a Oracle propôs contribuir com o NetBeans IDE como um novo projeto de código aberto dentro da Incubadora Apache, chamado Apache NetBeans (incubating). Entre diversas razões, mudá-lo para um local neutro, como o Apache, com seu modelo de governança forte, é suscetível de levar a um aumento no número de contribuições de várias organizações. Os principais desenvolvedores virão de uma série de organizações, incluindo a Oracle, que continuará seu investimento no NetBeans.

Recursos

A IDE NetBeans auxilia programadores a escrever, compilar, depurar e instalar aplicações, e foi arquitetada em forma de uma estrutura reutilizável que visa simplificar o desenvolvimento e aumentar a produtividade, pois reúne em uma única aplicação todas estas funcionalidades. Totalmente escrita em Java, mas que pode suportar qualquer outra linguagem de programação que desenvolva com Swing, como C, C++, Ruby e PHP. Também suporta linguagens de marcação, como XML e HTML.

O NetBeans fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e APIs (Application Program Interface, um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de software) além de uma documentação vasta — inclusive em português — bem organizada. Tais recursos auxiliam o desenvolvedor a escrever seu software de maneira mais rápida. A distribuição da ferramenta é realizada sob as condições da SPL (Sun Public License), uma variação da MPL (Mozilla Public License). Esta licença tem como objetivo garantir a redistribuição de conhecimento à comunidade de desenvolvedores quando novas funcionalidades forem incorporadas à ferramenta.

Interface gráfica do usuário

Descrição gerada automaticamente  
Atualmente, está distribuído em diversos idiomas; e isto tem o tornado cada vez mais popular, facilitando o acesso a iniciantes em programação e possibilitado o desenvolvimento de aplicativos multilíngue.

Como o NetBeans é escrito em Java, é independente de plataforma, funcionando em qualquer sistema operacional que suporte a máquina virtual Java (JVM).

Alguns dos seus principais recursos são:

Editor de código fonte integrado, rico em recursos para aplicações Web (Servlets e JSP, JSTL, EJBs) e aplicações visuais com Swing que é uma API (Interface de Programação de Aplicativos) Java para interfaces gráficas. A API Swing procura desenhar por contra própria todos os componentes, ao invés de delegar essa tarefa ao sistema operacional, como a maioria das outras APIs de interface gráfica trabalham;

Visualizador de classes integrado ao de interfaces, que gera automaticamente o código dos componentes de forma bem organizada, facilitando assim o entendimento de programadores iniciantes;

Suporte ao Java Enterprise Edition, plataforma de programação de computadores que faz parte da plataforma Java voltada para aplicações multicamadas, baseadas em componentes que são executados em um servidor de aplicações;

Plugins para UML (Unified Modeling Language - Linguagem de Modelagem Unificada) não proprietária de terceira geração, e desenvolvimento remoto em equipes; interface amigável com CVS ou Concurrent Version System (Sistema de Versões Concorrentes) é um sistema de controle de versão que permite que se trabalhe com diversas versões de arquivos organizados em um diretório e localizados local ou remotamente, mantendo-se suas versões antigas e os logs de quem e quando manipulou os arquivos;

CSS, algumas funcionalidades para editar folhas de estilos como destaques, recursos de autocompletar, análise de código;

Ajuda local e on-line;

Debug apurado de aplicações e componentes;

Autocompletar avançado; total suporte ao ANT, ferramenta de automatização da construção de programas e TOMCAT, servidor de aplicações Java para web;

Integração de módulos;

Suporte a Database (banco de dados), Data view e Connection wizard, que são os módulos embutidos na IDE; geração de Javadoc: a ferramenta permite a geração automática de arquivos javadoc em HTML a partir dos comentários inseridos no código, além de recursos que facilitam a inclusão de comentários no código;

Atalhos para copiar linhas inteiras de código.

Falaremos sobre o ambiente de dessenvolvimento do banco de dados

XAMPP

XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl. De plataforma, software livre, que consiste principalmente na base de dados MySQL, o qual foi substituído pelo MariaDB (embora ainda seja utilizado MySQL em algumas versões)[1], o servidor web Apache e os interpretadores para linguagens de script: PHP e Perl, além de um cliente FTP. O nome provem da abreviação de X (para qualquer dos diferentes sistemas operativos), Apache, MariaDB, PHP, Perl. É um método que torna extremamente fácil para os desenvolvedores a criar um servidor web local para fins de teste.[2]

Interface gráfica do usuário

Descrição gerada automaticamente  
O programa está liberado sob a licença GNU e atua como um servidor web livre, fácil de usar e capaz de interpretar páginas dinâmicas. Atualmente XAMPP está disponível para Microsoft Windows, GNU/Linux, Solaris, e MacOS X.

Características e Requerimentos

XAMPP somente requer um arquivo ZIP, RAR, ou exe a baixar e executar, com algumas pequenas configurações específicas ao sistema em alguns de seus componentes necessários para o funcionamento do servidor web. XAMPP é regularmente atualizado para incorporar as últimas versões de Apache/MySQL/PHP e Perl. Também são incluídos outros módulos como OpenSSL, e phpMyAdmin. Para instalar, o XAMPP necessita de uma pequena fração do tempo necessário para instalar e configurar programas separadamente.

Aplicações

Oficialmente, os desenvolvedores do XAMPP só pretendiam usá-lo como uma ferramenta de desenvolvimento, para permitir aos programadores de websites e programadores testar seu trabalho em seus próprios computadores, sem necessitar acesso algum à Internet. Na prática atual, XAMPP é utilizado atualmente para servir sites web na WWW, e com algumas modificações é geralmente seguro para uso em servidor público. Uma ferramenta especial é incluída para proteger facilmente as partes mais importantes e sensíveis do pacote.

O XAMPP possui muitos aplicativos, dentre eles o Apache, MySQL, phpMyAdmin, FileZilla FTP Server e OpenSSL.

Uma explicação final sobre o projeto

A aplicação final ficou um cronometro para uso em corridas, suas funcionalidades são:

Marcar o tempo com o cronometro assim que é iniciado, pausar o cronometro, e redefinir (zerar o cronometro para que seja feita uma nova marcação de tempo).

A marcação do tempo de cada volta é feita manualmente em dois campos, um para a volta 1 e outro para volta 2!

Também é possível calcular a média do tempo para obter uma parcial dos tempos, e também salvar os dados obtidos no banco de dados.

A também um campo para colocar o nome e um id para a equipe que vai ser cronometrada o tempo no momento! É possível acessar o banco de dados para ver as informações salvas e alteralas ou excluilas.

O sistema tem integração com o banco de dados mySQL.

**Resultados e Discursões**

As discussões do nosso grupo se voltaram em questões especificas e divergentes de área, a nossa primeira divergência é em qual método iríamos utilizar para construir o nosso front-end ( a parte gráfica do nosso projeto), após debates entramos em concordância de utilizar os componentes gráficos do Java Swing, dessa forma visamos buscar seus benefícios.

  
O Java Swing é uma biblioteca de componentes gráficos para a criação de interfaces de usuário (UI) em aplicativos Java. Ele oferece uma ampla gama de recursos e benefícios que o tornam uma escolha popular para o desenvolvimento de aplicativos desktop. Aqui estão alguns dos benefícios de usar o Java Swing:

Portabilidade: Uma das principais vantagens do Java Swing é sua portabilidade. Ele é baseado na plataforma Java, o que significa que os aplicativos desenvolvidos usando o Java Swing podem ser executados em várias plataformas, como Windows, macOS e Linux, sem a necessidade de alterar o código-fonte.

Interface personalizável: O Java Swing permite criar interfaces de usuário altamente personalizáveis. Ele fornece uma ampla variedade de componentes gráficos, como botões, caixas de seleção, barras de progresso, tabelas e muito mais, que podem ser personalizados para atender às necessidades específicas do aplicativo. Além disso, é possível aplicar temas e estilos visuais diferentes para alterar a aparência dos componentes.

Rich set de recursos: O Java Swing oferece um conjunto rico de recursos para o desenvolvimento de interfaces de usuário interativas. Ele suporta recursos avançados, como arrastar e soltar, redimensionamento de componentes, eventos de mouse e teclado, e suporte a várias janelas e diálogos. Isso permite criar interfaces de usuário dinâmicas e intuitivas.

Suporte a internacionalização: O Java Swing oferece suporte a internacionalização e localização, permitindo que os aplicativos sejam facilmente adaptados para diferentes idiomas e regiões. É possível criar interfaces de usuário com texto localizado e formatos de data, hora e números específicos de cada localidade.

Compatibilidade com outras tecnologias Java: O Java Swing é compatível com outras tecnologias Java, como JDBC (Java Database Connectivity) para acesso a bancos de dados, API de rede para comunicação em rede e APIs de persistência de dados, como Java Persistence API (JPA). Isso torna mais fácil integrar o Swing com outras partes do aplicativo e aproveitar a vasta gama de recursos do ecossistema Java.

Comunidade e documentação: O Java Swing possui uma comunidade ativa de desenvolvedores e uma ampla documentação disponível. Há uma abundância de recursos online, tutoriais, exemplos de código e fóruns de suporte onde os desenvolvedores podem encontrar ajuda e compartilhar conhecimentos.

Com todos esses benefícios em mente, montamos nosso projeto em torno do Java Swing, para que possamos usufruir de uma interface poderosa, flexível e com sua portabilidade e personalização avançadas. Segue nas imagens 1.1.1 à 1.1.2 o resultado dessas aplicações.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Usar um banco de dados online em uma aplicação Java pode trazer uma série de benefícios significativos. Aqui estão algumas razões pelas quais você deve considerar o uso de um banco de dados online em sua aplicação Java:

Acessibilidade global: Com um banco de dados online, seus dados podem ser acessados a partir de qualquer lugar do mundo, desde que haja conexão com a internet. Isso é especialmente útil em aplicativos distribuídos ou quando você precisa compartilhar dados entre diferentes usuários ou dispositivos. Independentemente de onde seus usuários estejam localizados, eles poderão acessar e atualizar os dados de maneira rápida e eficiente.

Confiabilidade e escalabilidade: Um banco de dados online geralmente é hospedado em servidores confiáveis e robustos, com redundância e backups regulares. Isso significa que seus dados estarão protegidos contra falhas de hardware ou perda de dados. Além disso, os bancos de dados online geralmente oferecem opções de escalabilidade, permitindo que você dimensione seus recursos de acordo com as necessidades do seu aplicativo à medida que ele cresce.

Gerenciamento simplificado: Com um banco de dados online, você não precisa se preocupar com a instalação, configuração e manutenção do servidor de banco de dados. Essas tarefas são geralmente tratadas pelo provedor de hospedagem. Isso economiza tempo e esforço, permitindo que você se concentre no desenvolvimento do aplicativo em si.

Backup e recuperação de dados: Um banco de dados online geralmente oferece recursos de backup e recuperação de dados automatizados. Isso significa que você não precisa se preocupar em perder dados importantes devido a falhas de hardware ou erros humanos. Caso ocorra algum problema, é possível restaurar facilmente os dados a partir dos backups disponíveis.

Integração com outras tecnologias: Ao usar um banco de dados online, você pode facilmente integrá-lo com outras tecnologias e serviços web. Por exemplo, você pode usar APIs para integrar seu aplicativo Java com serviços de autenticação, envio de e-mails, notificações push, análise de dados e muito mais. Isso permite que você aproveite a ampla gama de recursos e serviços disponíveis na nuvem.

Escalabilidade e desempenho: Os provedores de banco de dados online geralmente têm infraestrutura robusta que permite escalar os recursos do banco de dados conforme necessário. Isso significa que seu aplicativo pode lidar com um número crescente de usuários e transações sem comprometer o desempenho. Além disso, muitos provedores oferecem recursos avançados, como balanceamento de carga e caches, para melhorar ainda mais o desempenho do banco de dados.

Em resumo, usar um banco de dados online em uma aplicação Java oferece uma série de benefícios, como acessibilidade global, confiabilidade, gerenciamento simplificado, backup e recuperação de dados automatizados, integração com outras tecnologias e escalabilidade de recursos e desempenho. Esses benefícios contribuem para uma experiência aprimorada para os usuários finais e permitem que você se concentre no desenvolvimento de seu aplicativo, em vez de lidar com a infraestrutura de banco de dados. Como forma de demonstrar os resultados, segue nas figuras 1.1.3 e 1.1.4 o código que faz a ligação com o host online.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Outro quesito das nossas discussões foi respetivo do uso das DAOs, e pensamos que graças ao que ela poderia somar ao projeto utilizaríamos a mesma, como resultado trazemos ao projeto:

Separação de responsabilidades: O uso de DAOs ajuda a separar a lógica de negócios da lógica de acesso aos dados. Isso facilita a manutenção e a evolução do código, pois cada componente tem uma responsabilidade bem definida.

Abstração do acesso aos dados: As DAOs fornecem uma camada de abstração que oculta os detalhes específicos do armazenamento de dados subjacente, como um banco de dados relacional ou um serviço web. Isso permite que a lógica de negócios trabalhe com objetos de domínio em um nível mais alto, sem se preocupar com os detalhes de como os dados são armazenados ou recuperados.

Reutilização de código: Ao encapsular a lógica de acesso aos dados em DAOs, é possível reutilizar essa lógica em diferentes partes da aplicação. Isso evita a duplicação de código e promove a consistência e a coesão do sistema.

Flexibilidade no acesso aos dados: As DAOs permitem que você altere a forma como os dados são armazenados sem afetar a lógica de negócios. Por exemplo, você pode trocar de um banco de dados relacional para um banco de dados NoSQL sem precisar modificar o código que usa as DAOs. Isso facilita a adaptação a novas tecnologias e requisitos.

Testabilidade: As DAOs podem ser facilmente testadas de forma isolada, permitindo a criação de testes unitários para verificar se a lógica de acesso aos dados está funcionando corretamente. Isso é especialmente útil ao trabalhar com bancos de dados, pois você pode usar mocks ou bancos de dados em memória para os testes, sem depender de um ambiente real de desenvolvimento ou produção.

Dessa forma, o uso de DAOs em Java proporciona uma camada de abstração entre a lógica de negócios e o acesso aos dados, promovendo a separação de responsabilidades, reutilização de código, flexibilidade e testabilidade. Esses benefícios contribuem para o desenvolvimento de aplicativos mais modularizados, robustos e de fácil manutenção. E pensando em tudo isso trouxemos as DAOs para o projeto, segue nas figuras 1.1.5, 1.1.6 e 1.1.7, a representação do código utilizado para colocar as DAOs no projeto.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Considerações**

O uso de cronômetros em Java juntamente com bancos de dados MySQL oferece uma série de benefícios e considerações importantes. Aqui estão algumas considerações finais sobre essa combinação:

Conexão com o banco de dados: Ao utilizar um cronômetro em Java com banco de dados MySQL, é necessário estabelecer uma conexão adequada entre a aplicação Java e o banco de dados. Isso envolve o uso de bibliotecas JDBC (Java Database Connectivity) para estabelecer a conexão, enviar consultas SQL e recuperar os resultados.

Armazenamento de dados de cronômetros: O banco de dados MySQL pode ser usado para armazenar informações relacionadas aos cronômetros, como tempos registrados, datas, durações e outras informações relevantes. É importante projetar adequadamente a estrutura do banco de dados para armazenar e recuperar esses dados de maneira eficiente.

Consultas e operações com cronômetros: É possível executar consultas SQL para recuperar informações específicas sobre os cronômetros armazenados no banco de dados. Isso pode incluir a obtenção de tempos específicos, cálculos de duração, filtragem com base em critérios e muito mais. O conhecimento de SQL é fundamental para executar essas operações.

Gerenciamento de concorrência: Se sua aplicação Java estiver lidando com vários cronômetros simultaneamente e permitindo a interação do usuário, é importante considerar o gerenciamento de concorrência. Isso pode envolver o uso de bloqueios ou mecanismos de sincronização para garantir que as operações de leitura e gravação no banco de dados sejam executadas corretamente e que não ocorram conflitos.

Desempenho e otimização: Ao lidar com grandes quantidades de dados de cronômetros ou executar consultas complexas no banco de dados MySQL, é importante otimizar o desempenho. Isso pode incluir a criação de índices adequados nas colunas relevantes, ajuste de consultas para melhorar a eficiência e o uso de técnicas de cache, se necessário.

Segurança: A segurança dos dados é uma preocupação crucial ao usar um banco de dados MySQL. Certifique-se de implementar medidas de segurança adequadas, como criptografia de dados, autenticação e autorização, para proteger as informações dos cronômetros e evitar acesso não autorizado.

Em resumo, o uso de cronômetros em Java com banco de dados MySQL oferece uma maneira poderosa de armazenar e manipular dados temporais. No entanto, é essencial ter um bom entendimento das práticas recomendadas para garantir uma implementação segura, eficiente e confiável

**Referências Bibliográficas**

Budi Kurniawan. (2017). Java for Android: Developing Android Apps Using Java. Self-published.

Cay S. Horstmann. (2019). Core Java Volume I -- Fundamentals (11th Edition). Pearson.

Joshua Bloch. (2018). Effective Java (3rd Edition). Addison-Wesley Professional.

Herbert Schildt. (2018). Java: A Beginner's Guide (8th Edition). McGraw-Hill Education.

Venkat Subramaniam. (2016). Functional Programming in Java: Harnessing the Power of Java 8 Lambda Expressions. Pragmatic Bookshelf.

**Código Fonte**

O nosso programa foi divido em 7 pastas de código fonte sendo a primeira pasta Beans, segunda conexão pasta onde e criado a conexão com mysql, terceira cronometro sendo que no cronometro ainda existe mais 3 pastas de código para o front end do programa , quarta DAO, e última teste sendo ela responsável por testa se a conexão com banco foi bem sucedida.

* **Beans**

**Tela de computador

Descrição gerada automaticamente**

* **Conexão**

**Tela de computador

Descrição gerada automaticamente**

* **Cronometro**

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador com jogo

Descrição gerada automaticamente**

* **Tela Equipes**

**Interface gráfica do usuário, Texto

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

* **Tela principal**

**Texto

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

* **Dao**

**Tela de computador

Descrição gerada automaticamente**

**Tela de computador

Descrição gerada automaticamente**

* **Teste**

**Tela de computador

Descrição gerada automaticamente**