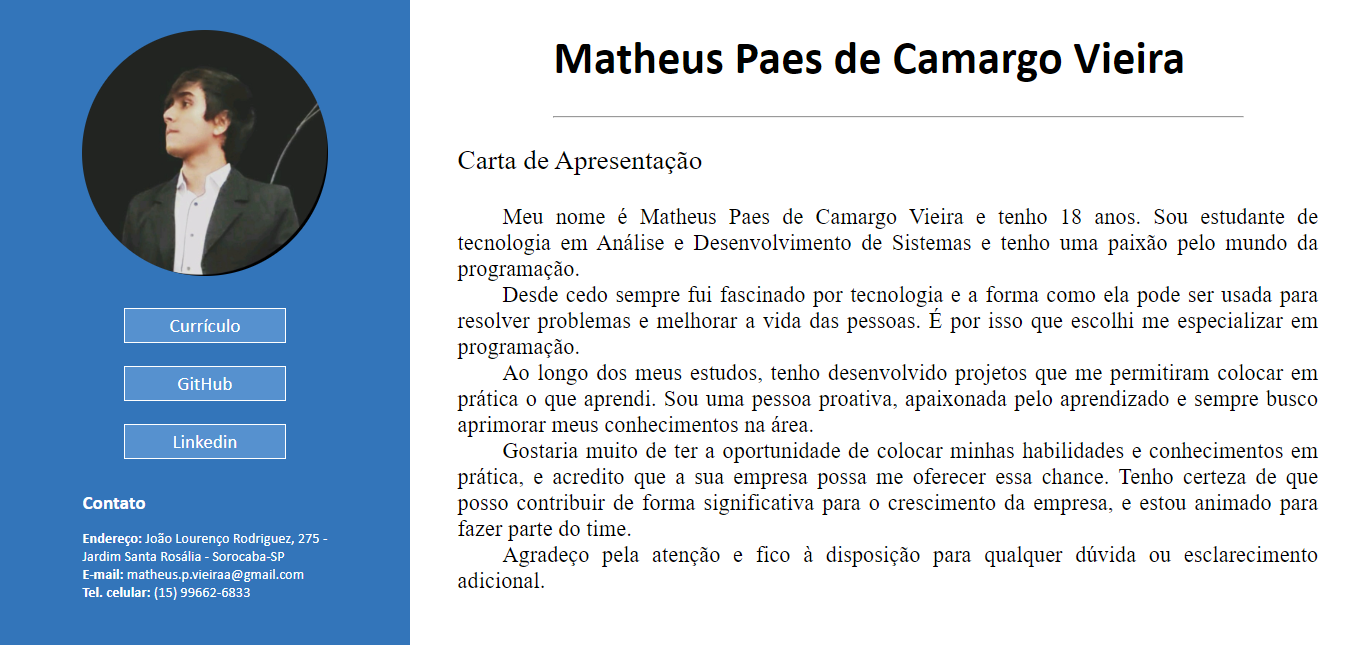
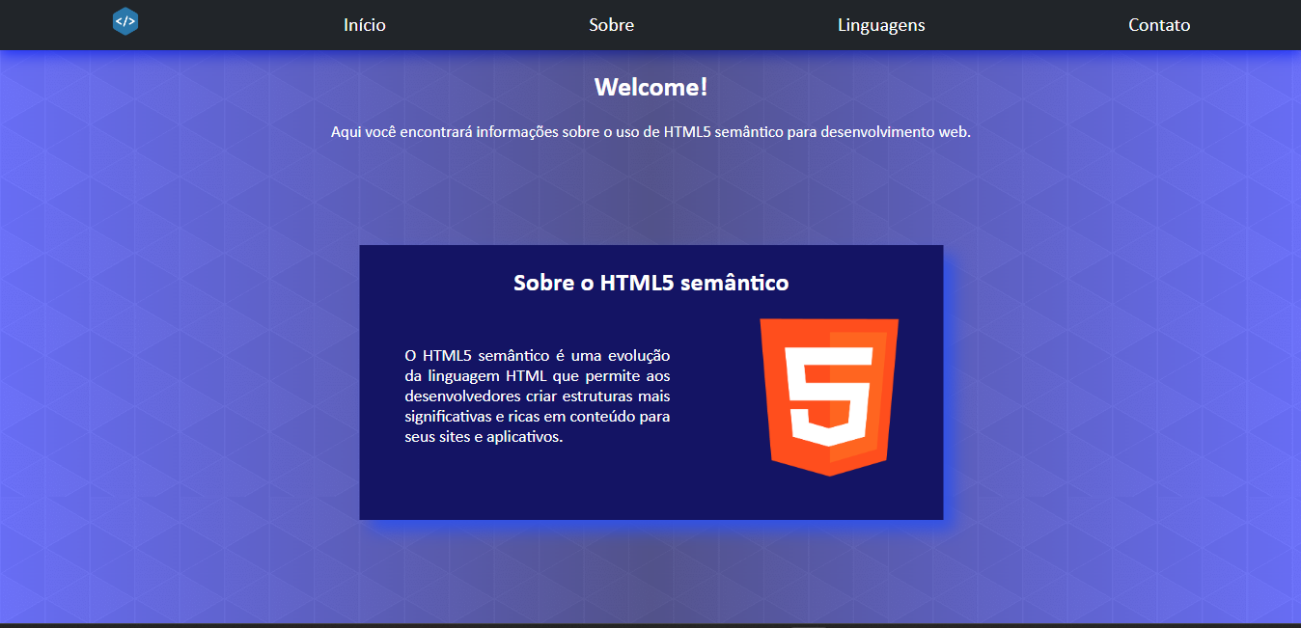
# **Linguagem de Programação:**

Inicialmente, nas aulas foram abordados o HTML, que é uma linguagem de marcação, e o CSS, que é uma linguagem de estilização. Essas tecnologias são fundamentais para o desenvolvimento de páginas web e aplicativos.

Durante esse período de estudo aprendemos a utilizar o HTML e CSS de forma mais básica e simples. Posteriormente desenvolvemos um currículo pessoal fazendo uso desses conhecimentos básicos em HTML e CSS para que exercitarmos em especial o posicionamento de elementos, o que pode ser muito desafiador em algumas situações.

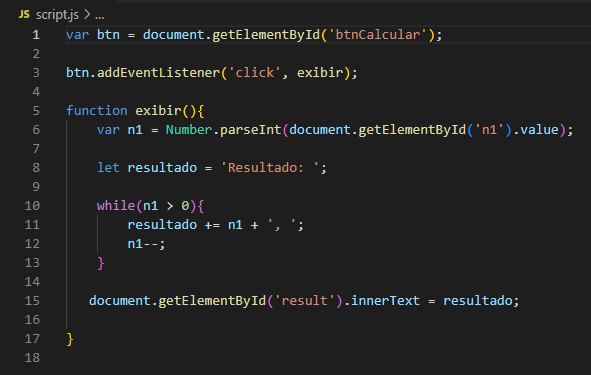
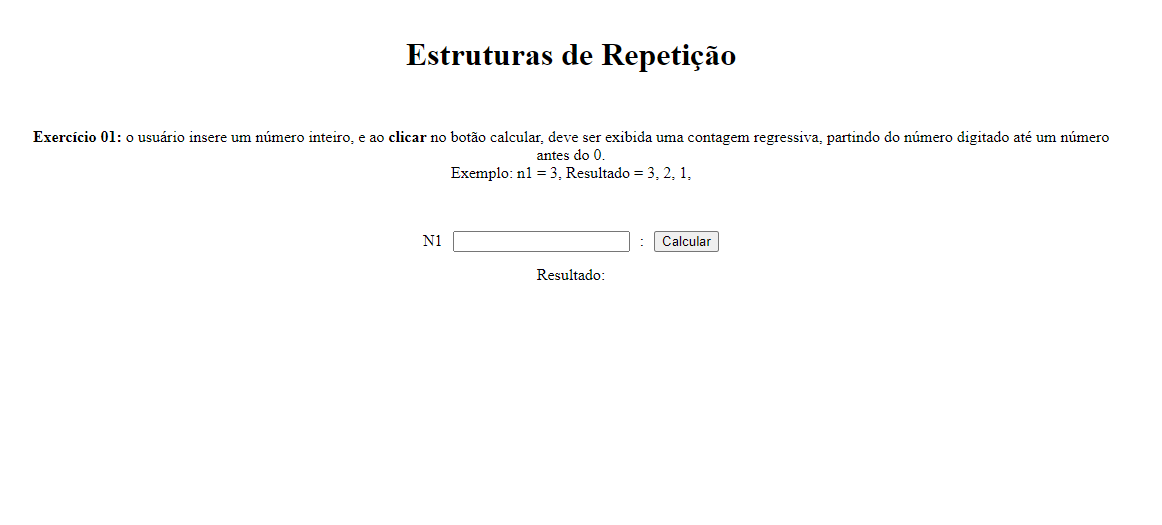
Em seguida nos foi apresentado o HTML Semântico, uma forma um pouco mais adequada de trabalhar com o HTML, por utilizar elementos que têm um significado claro e descritivo, tornando o código mais legível e compreensível para os desenvolvedores. Para colocarmos em prática nossa nova competência desenvolvemos uma outra atividade, porém esse trabalho foi realizado em duplas. Nele deveríamos “clonar” uma página já desenvolvida pelo professor, mas com nossas próprias peculiaridades, além de também pesquisar efeitos gerados pelo CSS que não foram mostrados anteriormente. Dessa forma não só exercitamos o novo conhecimento obtido, mas também demos nossos primeiros passos para a busca de saberes fora dos passados em sala de aula.



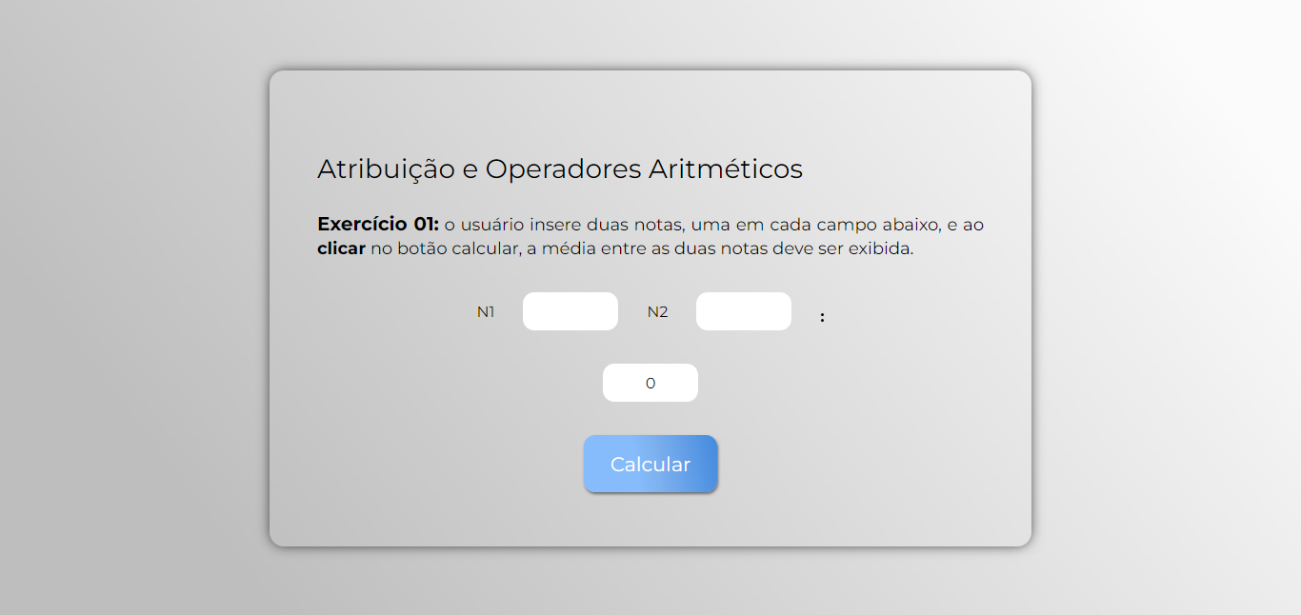
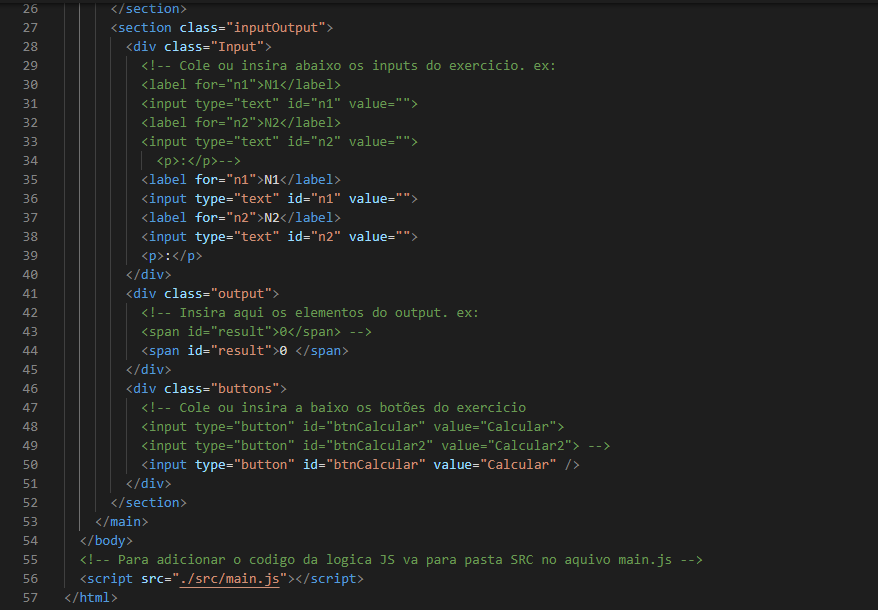
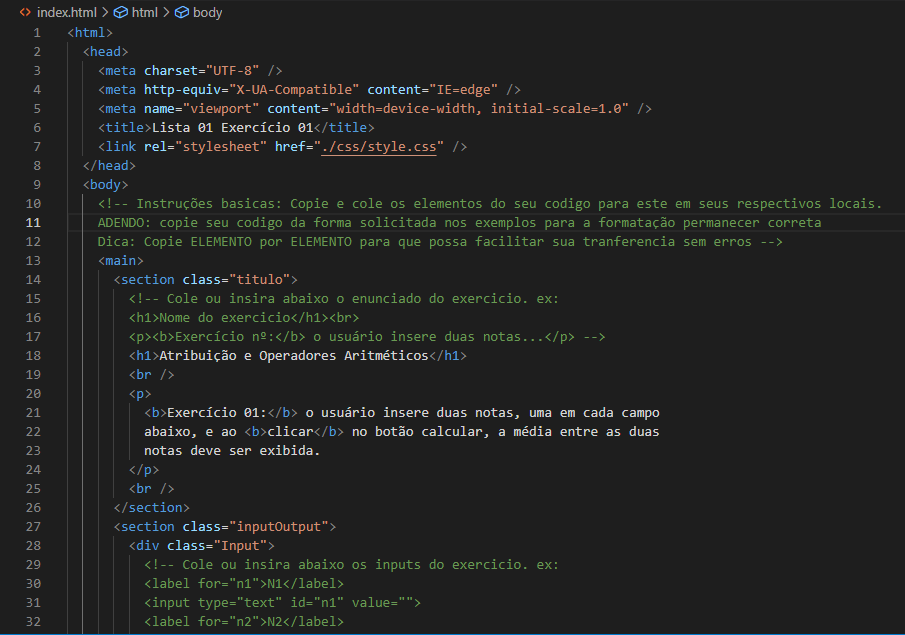
JavaScript foi a primeira linguagem de programação trabalhada, devido a possibilidade de integração dela com o HTML e CSS. Ela é uma linguagem de alto nível, interpretada pelo navegador, que permite adicionar uma maior interatividade por parte do usuário e um dinamismo para os elementos da página.

Justamente por se tratar da primeira linguagem de programação explorada, os principais conhecimentos básicos foram estudados por meio do JavaScript. Dentre eles podemos citar os dados primitivos (char, boolean, int e float), e como eles se aplicam nas variáveis que também podem ser fracas ou fortes. Inicialmente também vimos alguns tipos de operadores no JavaScript, os aritméticos, condicionais, lógicos e ternário.

Com a devida introdução a linguagem de programação JavaScript, demos início ao estudo do DOM. Nesse período vimos alguns métodos de busca de informações, eventos do DOM e operações condicionais. Os exercícios passados para fixação desse conteúdo foram 3 listas de atividades. A primeira delas tinha um enfoque em fazer uso de operadores básicos no JavaScript, a segunda utilizava as estruturas de decisão e a terceira as estruturas de repetição. Abaixo podemos ver um dos exercícios da terceira lista. Nele fazemos uso da estrutura de repetição “while” com a função de repetir uma soma até que o número inserido pelo usuário seja atingido.



Agora que já tínhamos um conhecimento em HTML, CSS e JavaScript, algumas atividades nos foram propostas, a fim de testar nossa aprendizagem. Na primeira delas, um grupo de alunos, no qual eu estava incluído, teve que desenvolver um CSS mais robusto para apresentar os exercícios das listas de atividades passadas anteriormente. Portanto, desenvolvemos uma página, onde o usuário deveria acessar o código HTML e inserir parte do HTML já pré-estabelecido nos exercícios, e o CSS iria funcionar independente de qual exercício fosse inserido.



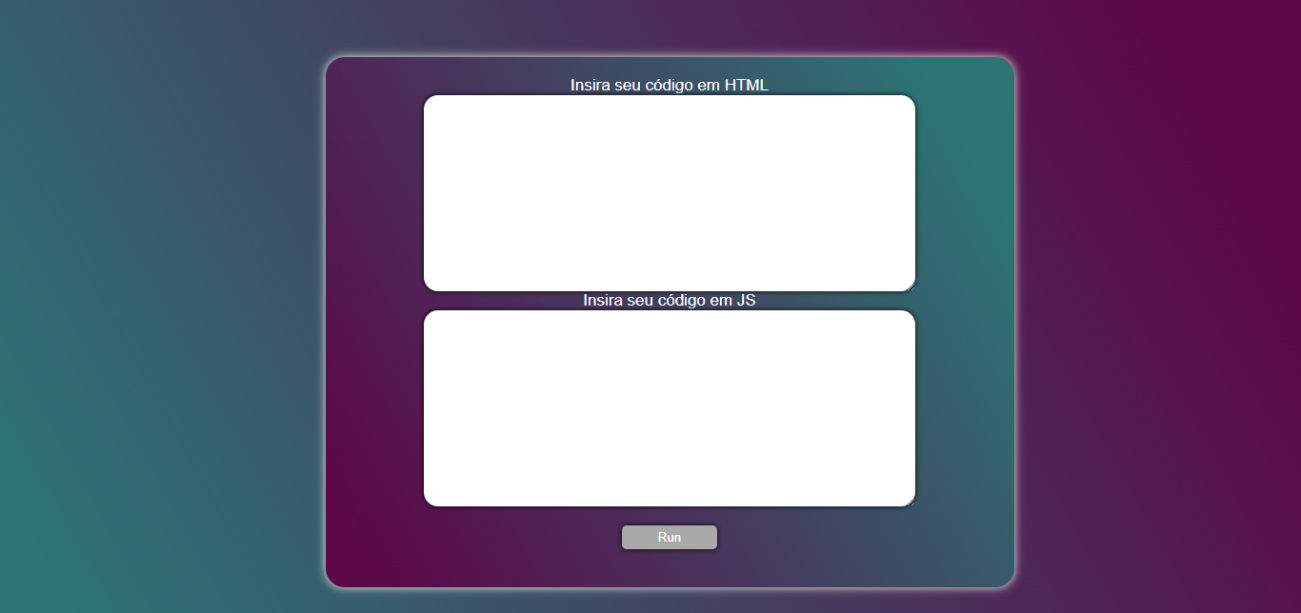
Após apresentado esse projeto, a sala foi dividida em outros grupos que deveriam se inspirar nesse projeto inicial, e desenvolver uma interface onde o usuário pudesse inserir o código HTML e JS do exercício que deseja testar, sem a necessidade de acessar o código HTML dessa página. Durante esse projeto, o uso dos seletores DOM foi imprescindível.

Texto

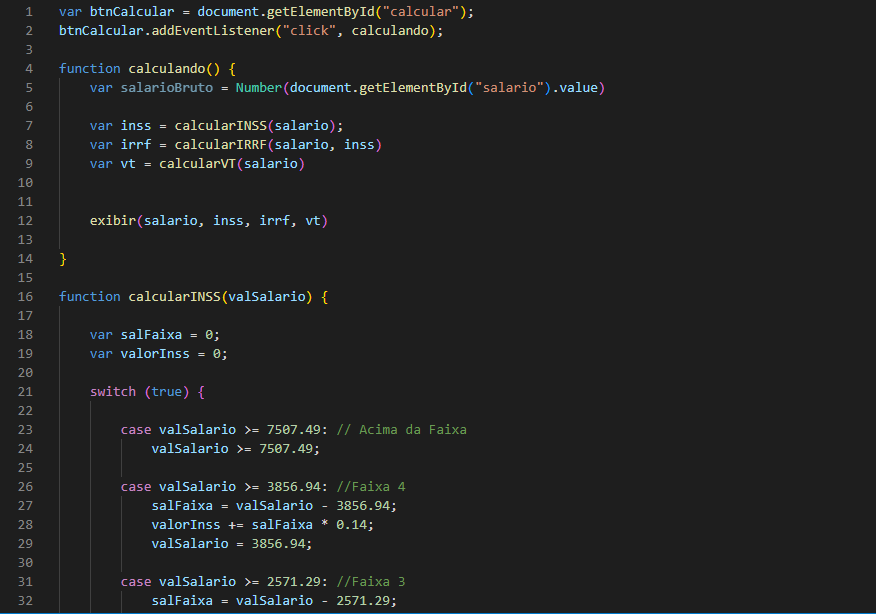
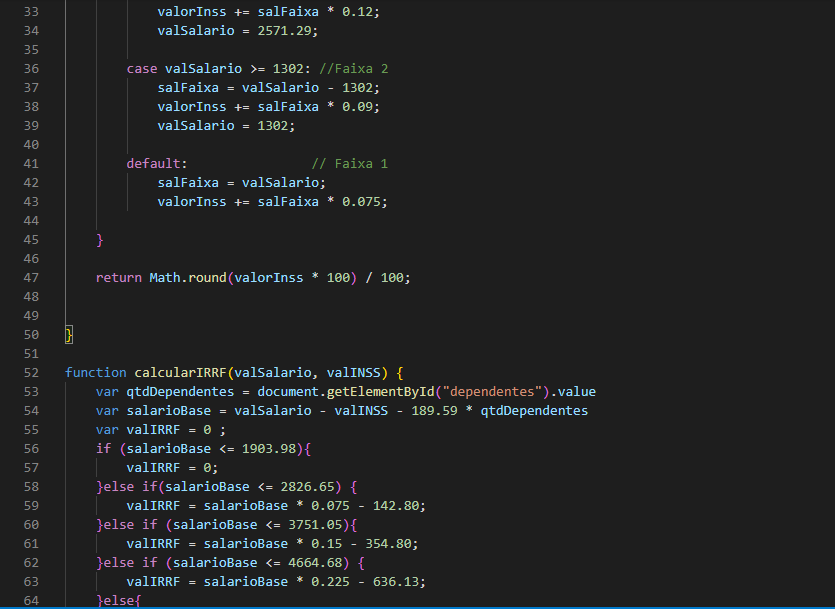
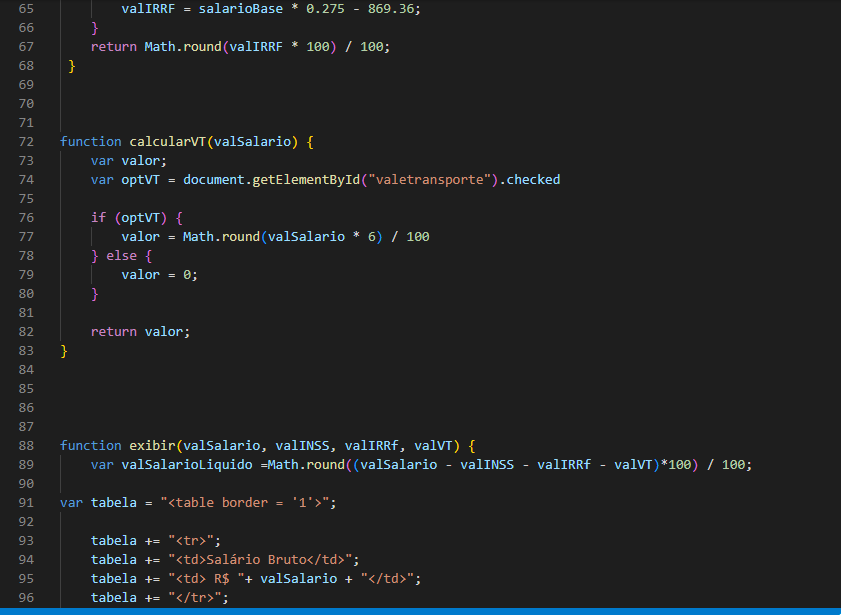
Descrição gerada automaticamente

Texto

Descrição gerada automaticamente



Por fim, foi proposta a criação de uma página para calcular o um salário liquido com base nos descontos do INSS, IRRF, vale transporte e quantidade de dependentes. Nesse exercício o calculo do INSS foi desenvolvido com o uso do “switch case”, enquanto o cálculo do IRRF utilizou do “if-else” para que pudéssemos praticar o uso de ambos os métodos.

# **Automação:**

O segundo tópico tratado durante as aulas foi Automação Industrial. Iniciamos com uma explicação teórica sobre os robôs. Nessa explicação vimos diversas características de diferentes tipos de robôs, como controle, mobilidade, estrutura cinemática, volume de trabalho e tipos de acionamento. Alguns elementos dos robôs foram citados sendo esses, base, ombro, braço, cotovelo, antebraço e ligamento.



Além desse estudo teórico, também utilizamos da plataforma TinkerCad para realizarmos alguns exercícios fazendo uso de componentes eletrônicos. Com o uso dessa plataforma também tivemos acesso ao Arduino, uma placa baseada em microcontroladores que trabalha com a linguagem C e C++. Por meio de alguns pequenos projetos desenvolvemos um conhecimento básico referente ao Arduino e a linguagem C. Nesses projetos foram usados sensores ultrassônicos, leds e botões, onde cada um deles contava com uma proposta diferente, seja simular o funcionamento de um semáforo ou exibir a distância medida pelo sensor ultrassônico. Referente a linguagem de programação C contemplamos algumas peculiaridades com a linguagem trabalhada anteriormente (JavaScript), como o uso de estruturas condicionais e de repetição, que embora tenham uma sintaxe diferente, tem os mesmas características e usos.

Uma imagem contendo Diagrama

Descrição gerada automaticamente Texto

Descrição gerada automaticamenteUma imagem contendo Diagrama

Descrição gerada automaticamenteTexto, Carta

Descrição gerada automaticamente Uma imagem contendo Diagrama

Descrição gerada automaticamente Texto

Descrição gerada automaticamente Texto

Descrição gerada automaticamente Interface gráfica do usuário, Diagrama

Descrição gerada automaticamente Texto

Descrição gerada automaticamente Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente com confiança média Interface gráfica do usuário, Diagrama

Descrição gerada automaticamente Texto

Descrição gerada automaticamente Texto

Descrição gerada automaticamente Aplicativo

Descrição gerada automaticamente com confiança média

# **Redes:**

Para darmos início ao conteúdo de redes, primeiramente estudamos conceitos básicos a fim de nivelar o conhecimento de todos sobre a temática. Nos foram apresentados os tipos de rede, sendo eles PAN (Personal Area Network), CAN - Controller Area Network, LAN - Local Area Network, MAN - Metropolitan Area Network, WAN - Wide Area Network. Ethernet também foi um tópico citado, por se tratar da conexão com fio amplamente utilizada para a conexão em uma rede local.

A topologia também foi um dos assuntos iniciais, já que se trata da forma em que a rede foi desenvolvida, seja de maneira física a forma que se dá a conexão dos cabos (estrela, barramento, anel, malha, árvore), ou lógica, o modo de circulação dos dados.

Posteriormente conhecemos as 7 camadas do modelo OSI suas funções e características. Elas são:

1. Camada Física: É responsável pela transmissão e recepção dos bits brutos por meio de um meio de comunicação físico, como cabos de rede. Lida com as características elétricas, mecânicas e físicas da transmissão de dados.
2. Camada de Enlace de Dados: Gerencia a transmissão confiável de dados entre dois nós adjacentes na rede. Trata de problemas como detecção de erros, controle de fluxo e acesso ao meio.
3. Camada de Rede: Lida com o roteamento dos pacotes de dados pela rede. Determina a melhor rota para enviar os pacotes do ponto de origem ao destino, levando em consideração fatores como congestionamento, priorização e endereçamento lógico.
4. Camada de Transporte: Fornece serviços de transporte confiáveis e transparentes para os processos de aplicação. Divide os dados em segmentos, fornece controle de fluxo e controle de erros para garantir que os dados cheguem corretamente e em ordem.
5. Camada de Sessão: Estabelece, mantém e encerra sessões de comunicação entre os aplicativos em diferentes dispositivos. Gerencia a troca de dados entre os pontos de extremidade, incluindo sincronização, controle de diálogo e recuperação em caso de falhas.
6. Camada de Apresentação: Lida com a representação dos dados em um formato compreensível para as aplicações. Realiza funções como criptografia, compressão, conversão de formatos e tradução de caracteres.
7. Camada de Aplicação: Fornece interfaces para que os aplicativos acessem os serviços de rede. Inclui protocolos como HTTP, FTP, SMTP, DNS, que permitem a comunicação e o compartilhamento de dados entre aplicativos em diferentes dispositivos.

Assim como o modelo OSI, temos também o modelo TCP/IP. Diferente do modelo OSI, este conta apenas com 4 camadas: interface de rede, internet, transporte e aplicação. Cada camada tem funções específicas, como comunicação física, endereçamento, roteamento, serviços confiáveis e suporte a aplicativos.

Além disso, as principais características e diferenças de UDP e TCP nos foram apresentadas. Ambos os protocolos de comunicação estão na camada de transporte do modelo OSI (Open Systems Interconnection). No entanto, existem algumas diferenças significativas entre os dois:

1. Confiabilidade: O TCP é um protocolo confiável, o que significa que ele garante que todos os dados enviados serão recebidos pelo destinatário sem erros ou perdas. O UDP, por outro lado, não é confiável e não garante a entrega de dados.
2. Controle de fluxo: O TCP implementa o controle de fluxo para garantir que o transmissor não envie mais dados do que o receptor pode processar. O UDP não possui esse recurso.
3. Orientação à conexão: O TCP é um protocolo orientado à conexão, o que significa que ele estabelece uma conexão antes que os dados sejam transmitidos. O UDP é um protocolo sem conexão, o que significa que ele não estabelece uma conexão antes da transmissão de dados.
4. Overhead: O TCP tem um overhead maior do que o UDP, já que ele adiciona informações adicionais para garantir a confiabilidade dos dados. O UDP tem um overhead mínimo.

Em resumo, o TCP é uma escolha mais apropriada quando a confiabilidade e a integridade dos dados são cruciais, enquanto o UDP é mais adequado quando a velocidade e a eficiência são mais importantes do que a confiabilidade.

Ainda na camada de rede temos o IP e a máscara de rede. O IP atribui um endereço único a cada dispositivo, enquanto a máscara de rede define a estrutura da rede e permite a comunicação entre dispositivos dentro da mesma rede e entre redes diferentes. Tópicos como DNS (um serviço de rede que converte nomes de domínio em endereços IP), DHCP (um protocolo de rede que fornece configuração automática de endereços IP e outros parâmetros de rede para dispositivos em uma rede), FTP (um protocolo de rede utilizado para transferir arquivos entre computadores em uma rede), HTTP (um protocolo de comunicação usado para transferir dados na World Wide Web) e HTTPS (uma extensão do protocolo HTTP que utiliza criptografia SSL/TLS para fornecer uma conexão segura entre um cliente e um servidor), também foram estudados durante esse período.

Por fim, o IPv4 e o IPv6 são protocolos essenciais no campo das redes de computadores. Estudamos em detalhes esses protocolos, sendo o IPv4 um protocolo de camada de rede que utiliza endereços de 32 bits, enquanto o IPv6 utiliza endereços de 128 bits. A compreensão das diferenças entre eles, suas características e implicações é crucial para o projeto e a implementação de redes modernas e escaláveis.

Após esse período de estudo teórico, desenvolvemos alguns projetos a fim de colocar em prática o conhecimento adquirido. Para isso utilizamos o software Cisco Packet Tracer, onde conhecemos alguns componentes essenciais para a criação de uma rede, como os switches, roteadores e servidores. Com o devido conhecimento sobre esses componentes demos início ao desenvolvimento de 3 projetos, o primeiro deles buscava apenas introduzir o conteúdo de forma prática, o segundo simulava a comunicação entre um determinado aparelho localizado no Zimbabue com outro localizado no Cazaquistão, e por último uma rede que simulava a conexão de um funcionário que trabalha de casa (home office) para uma determinada empresa.

Diagrama

Descrição gerada automaticamente com confiança médiaDiagrama

Descrição gerada automaticamente

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente com confiança média

# **Banco de Dados:**

Para o início do estudo de banco de dados, a introdução a alguns determinados tópicos foi necessária, minimundo, MER e DER. Isso se deu por se tratarem de métodos para o desenvolvimento inicial do banco de dados, onde podemos analisar a ideia inicial e adequar ela as necessidades do banco de dados. Posteriormente os relacionamentos e as entidades foram trabalhados, afinal para o desenvolvimento do banco de dados com os métodos citados fazemos uso desses elementos.

Outro tópico abordado foram as 3 formas normais de bancos de dados, normas essas que visam eliminar redundâncias e inconsistências em um banco de dados relacional. Ainda existem outras duas, que não foram abordadas no decorrer das aulas, e que se aplicam apenas em situações mais complexas e específicas.

1ª Forma Normal (1NF): Nesta forma normal, as tabelas devem possuir uma estrutura tabular, ou seja, os valores em cada coluna devem ser atômicos, não sendo permitido agrupar múltiplos valores em uma única célula. Além disso, cada tabela deve possuir uma chave primária única que identifica de forma única cada registro.

2ª Forma Normal (2NF): Nesta forma normal, além de atender aos critérios da 1NF, cada coluna não chave deve depender completamente da chave primária, e não de apenas uma parte dela. Isso significa que as dependências funcionais devem ser eliminadas, movendo as colunas dependentes para outras tabelas.

3ª Forma Normal (3NF): Nesta forma normal, além de atender aos critérios da 2NF, não devem existir dependências transitivas, ou seja, uma coluna não chave não deve depender de outra coluna não chave. Para eliminar essas dependências, é necessário criar tabelas adicionais.

Com as ideias básicas referentes a modelagem de banco de dados, realizamos uma lista de exercícios onde eram apresentados os minimundos e era necessária a construção de um DER com base nesse conteúdo. Essa atividade foi realizada dentro da plataforma do Draw.io, que conta com as ferramentas necessárias para a criação de um modelo DER.

Após a finalização desses exercícios, cada um deles foi direcionado a uma equipe, e então realizamos o desenvolvimento do modelo físico desse exercício. Para isso o Postgree, um SGBD (Sistema de Gerenciamento de Banco de Dados), foi utilizado na criação das tabelas e suas relações. Para essa construção foram utilizados comandos DDL (Data Definition Language) como o “CREATE”, “ALTER” e “DROP”.

Com a apresentação dos bancos criados, foi solicitada a realização de comando DQL (Data Query Language) para a apresentação de determinados dados específicos, para aferir nosso conhecimento dobre esses comandos.