

# Documento de orientações para as disciplinas de Raciocínio Lógico Algorítmico e Raciocínio Lógico e Algoritmo.

## Disciplinas: T998 e T160.

Prof. Paulo Cirillo Souza Barbosa

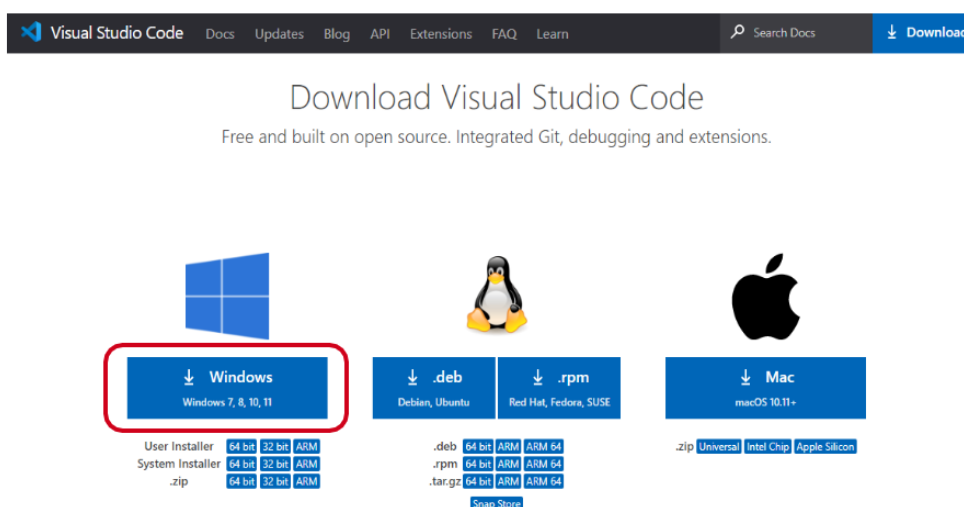
[pauloc@unifor.br](mailto:pauloc@unifor.br)

- Conteúdo:
  - Instalação do Visual Studio Code.
  - Como configurar o Visual Studio Code.
  - Como criar uma conta e configurar o Beecrowd.
  - Como entrar na sua turma no Beecrowd.
  - O que fazer para entregar questões no Beecrowd.

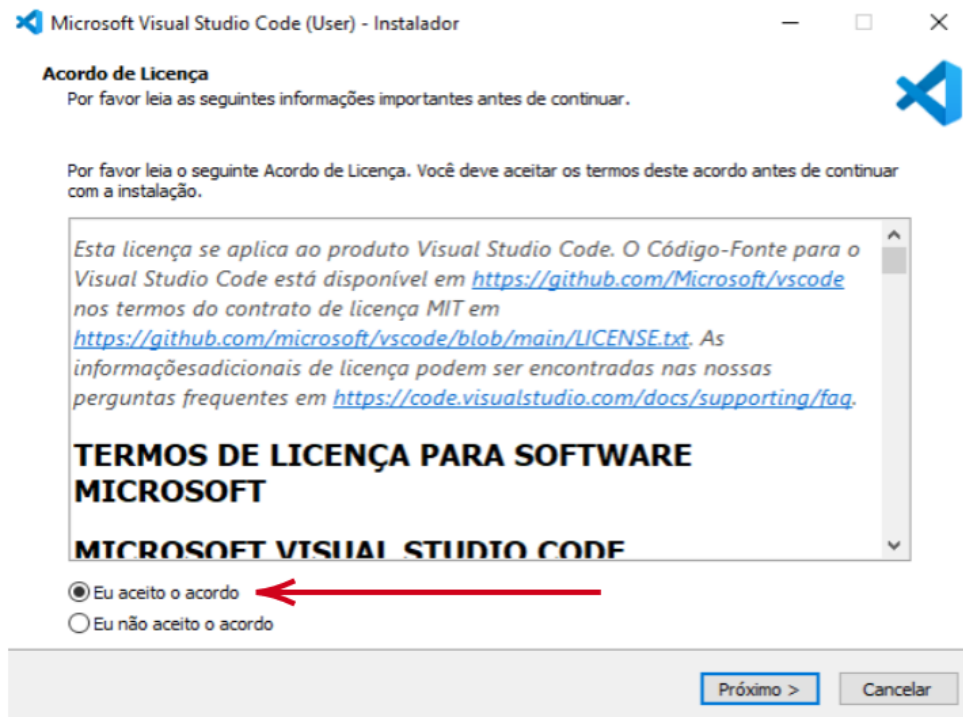
## ● Instalação e configuração do Visual Studio Code

Os algoritmos implementados ao longo da disciplina, serão desenvolvidos em uma ferramenta de sua escolha. Existem diversos Ambientes de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*), por exemplo, **Visual Studio Code**, **Notepad++**, **Sublime**, dentre outros. Porém, como discutido em sala de aula, o Visual Studio Code fornece suporte para diversas linguagens de programação, é leve, pode ser totalmente personalizado, possui uma facilidade para instalações de extensões sendo considerado um poderoso ambiente de desenvolvimento de algoritmos.

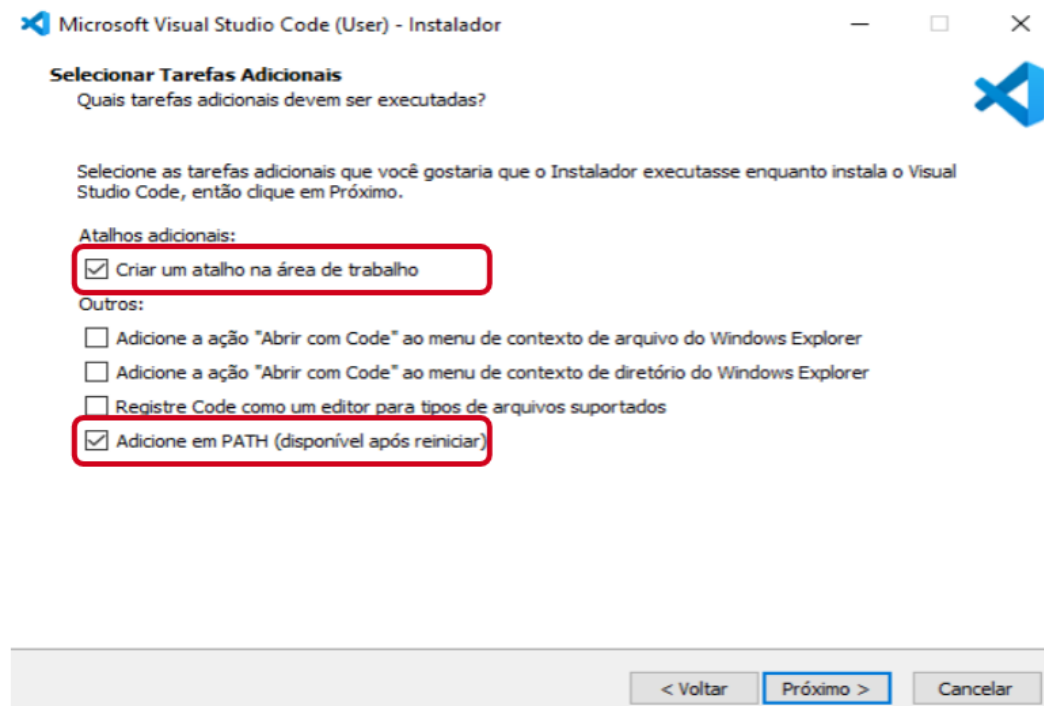
Para instalação da ferramenta, acesse <https://code.visualstudio.com/download> e clique no botão referente ao sistema operacional. O exemplo que se segue, é para um computador Windows baseado em arquitetura de 64bits.



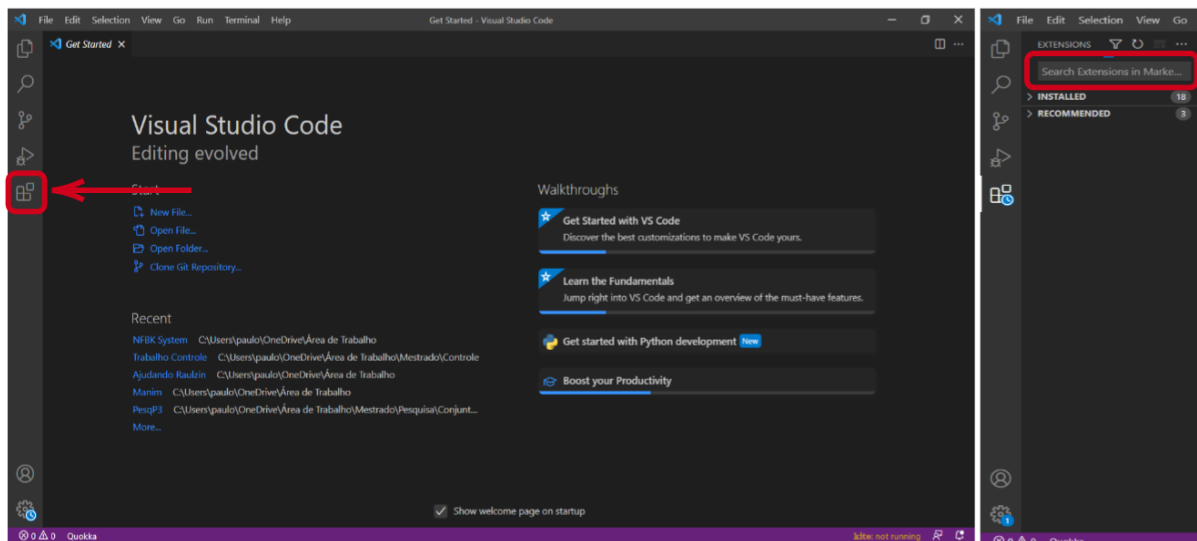
Após o download, clique no executável e quando a primeira tela ser apresentada, marque a opção “Eu aceito o acordo”, indicado pela seta em vermelho.



Em seguida, marque as opções destacadas em vermelho e clique em **Próximo>>Instalar**.



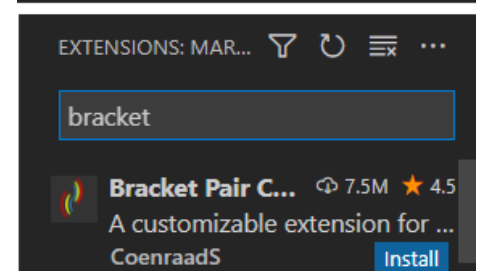
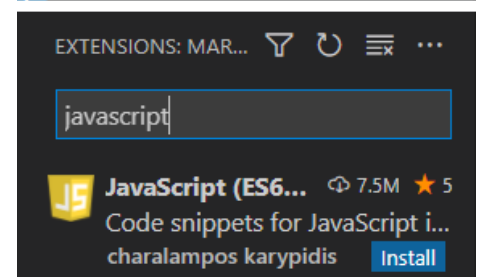
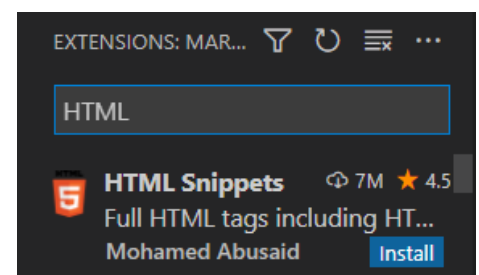
Quando a instalação terminar, acesse no menu na lateral esquerda o menu “Extensões”. Quando este for expandido, aparecerá um campo como ilustrado na figura que se segue.



Recomenda-se a instalação das seguintes extensões:

- HTML Snippets.
- JavaScript (ES6) code snippets
- Bracket Pair Colorizer

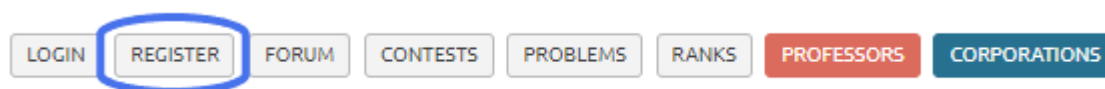
Para instalar, basta clicar no botão azul chamado “Install” como mostrado na Figura. Este caracteriza o fim da configuração do ambiente de desenvolvimento para as disciplinas T160 e T998.



### ● Criando e configurando uma conta no Beecrowd.

A ferramenta beecrowd será utilizada ao longo do semestre para a submissão dos algoritmos das listas de exercícios e resolução das provas de AV1 e AV2. A ferramenta trata-se de uma comunidade de programadores que fornece um ambiente propício para ensino no meio acadêmico e profissional. Além disso, a ferramenta provê diversos concursos e competições ao nível mundial, para programadores medirem seus conhecimentos. Por fim, análises entre universidades são feitas através de rankings internos e externos.

Para realizar o cadastro na ferramenta, acesse o link <https://www.beeccrowd.com.br/judge/en/login>. Quando a página é aberta, outras informações sobre a ferramenta são exibidas na página. Para realizar o cadastro, clique no menu superior no botão “REGISTER”.



SIGN UP WITH



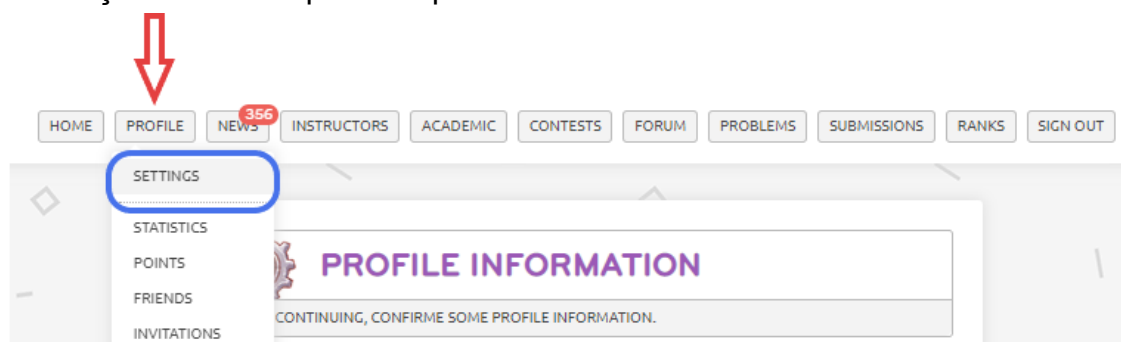
§ By signing up with a social platform you AGREE with the beecrowd Terms & Conditions.

You need to allow your EMAIL to be provided by the social sign in. Therefore, allow the default permissions when prompted.

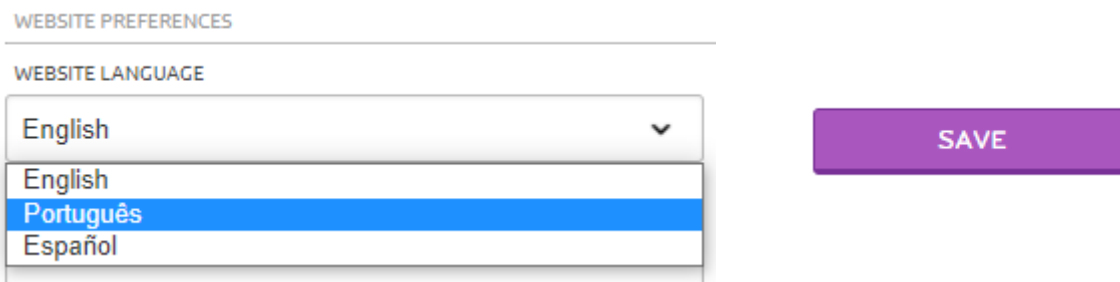
Para realizar o cadastro, o Beecrowd fornece possibilidade de vínculo com contas como **Facebook, Google, Gitlab, Bitbucket ou Twitter**. Faça a escolha dentre estas opções para realizar o cadastro e em seguida o login.

Em seguida, passe o mouse no menu superior na opção “PROFILE” e após a expansão do submenu, faça o clique em “SETTINGS”.

**Obs:** Há uma possibilidade do Beecrowd solicitar o cadastro inicial de algumas informações antes de permitir que você acesse “SETTINGS”



Nesta página carregada, há diversas seções, porém, algumas são obrigatórias. De qualquer forma, é interessante inicialmente trocar o idioma para português. Em “*WEBSITE PREFERENCES*”, troque a opção **English** para **Português** e clique em **SAVE** no canto inferior direito da página.



WEBSITE PREFERENCES

WEBSITE LANGUAGE

English

English

Português

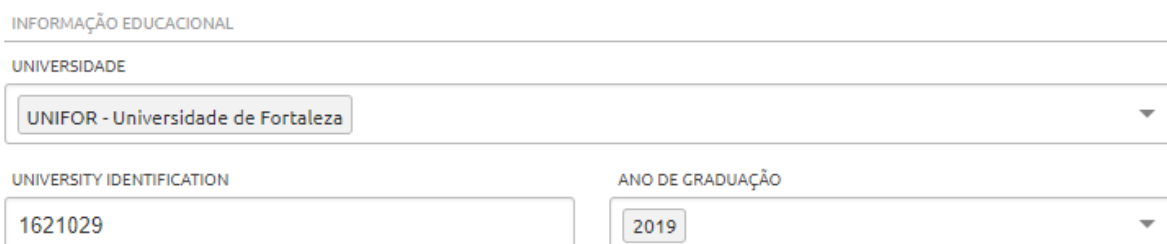
Español

SAVE

Após isso, na seção “*INFORMAÇÃO EDUCACIONAL*”, faça a atualização das seguintes informações:

- Universidade: Clique e busque a UNIFOR - Universidade de Fortaleza
- University Identification: Preencha com o número de sua matrícula sem o dígito final.
- Ano de graduação: preencha com o ano de previsão de conclusão.

Segue exemplo de preenchimento ao final:



INFORMAÇÃO EDUCACIONAL

UNIVERSIDADE

UNIFOR - Universidade de Fortaleza

UNIVERSITY IDENTIFICATION

1621029

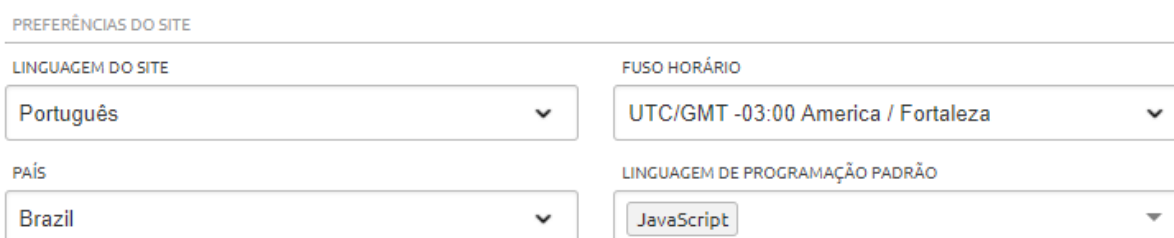
ANO DE GRADUAÇÃO

2019

A outra seção que terá informações modificadas é a “**PREFERÊNCIA DO SITE**”. Portanto, faça as seguintes alterações:

- FUSO HORÁRIO: Modifique para UTC/GMT - 03:00 America / Fortaleza
- LINGUAGEM DE PROGRAMAÇÃO PADRÃO: JavaScript

Segue imagem com exemplo já preenchido:



PREFERÊNCIAS DO SITE

LINGUAGEM DO SITE

Português

FUSO HORÁRIO

UTC/GMT -03:00 America / Fortaleza

PAÍS

Brazil

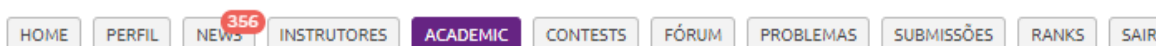
LINGUAGEM DE PROGRAMAÇÃO PADRÃO

JavaScript

Por fim, no final da página terão as opções de **PREFERÊNCIAS** e **OPÇÕES DE EMAIL**. Nesta, realize todas as marcações como mostrado na Figura que se segue:

PREFERÊNCIAS	OPÇÕES DE EMAIL
<input checked="" type="checkbox"/> <b>MOstrar TEMPLATE DO CÓDIGO FONTE</b> Eu quero ver o template do código fonte na página de submissão.	<input checked="" type="checkbox"/> <b>ME ENVIE ATUALIZAÇÕES</b> Sim! Eu quero receber e-mails de atualização do portal beecrowd!
<input checked="" type="checkbox"/> <b>MOstrar ASSUNTO DO PROBLEMA</b> Eu quero spoilers! Por favor mostre os assuntos dos problemas na lista de problemas.	<input checked="" type="checkbox"/> <b>ME ATUALIZE SOBRE CONTESTS E TORNEIOS</b> Claro, eu quero receber e-mails de atualizações sobre contests e torneios da beecrowd!
	<input checked="" type="checkbox"/> <b>ME ATUALIZE SOBRE NOVIDADES DE PARCEIROS</b> Claro, eu quero receber e-mails de atualizações sobre contests e torneios da beecrowd!

Com isto, as configurações necessárias estão completas e então é possível entrar na sua turma para envio das soluções dos exercícios propostos, bem como das questões da prova. Portanto, no menu superior da página, clique em **“ACADEMIC”**.



Será exibido uma tela com informações sobre acesso e termos de serviços. Faça a leitura destes e em seguida marque a opção **“EU LI E ACEITO OS TERMOS DE SERVIÇO”**. Após isso, clique no botão **“SALVAR”** como mostrado na Figura:

INFORMAÇÃO
<p>O professor ou coach não terá acesso ao seu email e as informações referentes as soluções de outros problemas que não estão nas listas.</p> <p>O professor ou coach não tem permissão de reproduzir ou de utilizar o código fonte ou qualquer informação do aluno sem a permissão dele por escrito. A não conformidade com esta regra implicará na remoção da conta do professor.</p>

TERMOS DE SERVIÇO
<p>Você deve aceitar os seguintes termos de serviço para ter acesso ao beecrowd Academic:</p> <p>I. Ao aceitar o convite enviado pelo seu professor(a)/coach, você dará a ele/ela a permissão de acompanhar o seu progresso, isto é, visualizar quais problemas das listas você resolveu, quais você tentou e quando foi a última tentativa;</p> <p>II. Você autoriza o professor a visualizar o código fonte de todas as suas submissões para os problemas das listas de exercício desde a data de início até o prazo final das mesmas;</p> <p>III. Após aceitar um convite, caso o usuário não deseja mais participar de determinada lista, a exclusão só poderá ser feita pelo professor ou coach da lista. O portal não se responsabiliza por convites aceitos;</p> <p>IV. O portal beecrowd poderá alterar estes termos de serviço conforme a necessidade, para prover um serviço de melhor qualidade aos seus usuários. Para continuar utilizando o Academic, os usuário terão que concordar com os novos termos.</p> <p>Os Termos de Serviço do beecrowd Academic foram atualizados em 25/02/2014.</p> <p><input checked="" type="checkbox"/> <b>EU LI E ACEITO OS TERMOS DE SERVIÇO.</b></p>

VOCÊ DEVE CONCORDAR COM OS TERMOS PARA PODER TER ACESSO AS FERRAMENTAS DO ACADEMIC	<b>SALVAR</b>
--	---------------


Agora que já os termos foram aceitos, você pode acessar a disciplina ao clicar no botão **ACESSAR DISCIPLINA** exibido na Figura:


**DISCIPLINAS**
ACESSAR DISCIPLINA

ESTAS SÃO AS DISCIPLINAS QUE VOCÊ FOI CONVIDADO A PARTICIPAR.

<div>NENHUMA DISCIPLINA OU GRUPO</div> <div>Nenhum convite de um professor ou coach.</div>	<div>NENHUMA DISCIPLINA OU GRUPO</div> <div>Nenhum convite de um professor ou coach.</div>
--	--

Para ingressar na turma é necessário fornecer o ID DISCIPLINA e KEY. Estas informações são fornecidas pelo professor (Provavelmente estarão no AVA). Copie e cole o ID e a KEY e em seguida clique em **SIM**.


**ACESSAR**

INSIRA A CHAVE PARA ACESSAR A DISCIPLINA

**INFORMAÇÃO DA DISCIPLINA**

Ao aceitar o convite você está **permitindo** o professor a:

- Monitorar seu progresso nas listas de exercícios
- Visualizar todas as submissões para as listas
- Dar feedback para as suas soluções

ID DISCIPLINA

KEY

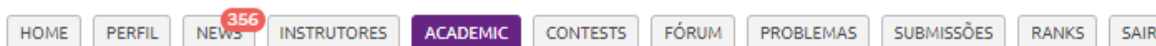
VOCÊ DESEJA PERMITIR O ACESSO?

**SIM**

## • Como submeter uma questão da lista de exercícios.

A sequência de passos descrita na presente seção, é uma **orientação** do que pode ser realizado para o desenvolvimento de algoritmos e em seguida sua submissão na plataforma Beecrowd. Por se tratar uma orientação, as instruções aqui descritas são apenas **recomendações**, portanto, podem ser realizadas de outra forma ou ordem. Entretanto, se esta é sua primeira experiência com a disciplina de Raciocínio Lógico e Algoritmo (T160) ou Raciocínio Lógico Algorítmico (T998), a recomendação que se siga a sequência de passos aqui descrita, é **reforçada**.

1. O primeiro passo é acessar sua turma no Beecrowd e iniciar a lista de exercícios liberada pelo professor. Portanto, com o login efetuado, clique no botão “ACADEMIC” no menu superior da página como mostrado na Figura:



2. Será exibido na tela a(s) disciplina(s) que você está vinculado, procure a referente ao semestre e professor atual e clique em “ABRIR”. A figura exibe um exemplo de como deve ser feito:



3. Em sequência, a ferramenta exibe um painel com todas as atividades que o professor já construiu até o momento. Essas atividades podem ser uma lista, uma prova ou exercício proposto. No painel, é mostrado também a data de início e a data limite para entrega da atividade. É importante destacar que a entrega somente é contabilizada no intervalo estipulado pelo professor. Por exemplo, na Figura que se segue o professor indicou o início da Lista AV1 para a data 16/02/2022 às 00:00 e com data limite 16/03/2022 às 00:00. Se por acaso você decida realizar uma questão da lista no dia 15/02/2022 esta não será contabilizada para o exercício do professor, bem como se essa submissão for realizada no dia 16/03/2022 às 07:00.

DISCIPLINA			
RACIOCÍNIO LÓG. ALGORÍTMICO N24CD			
#	HOMEWORK	INÍCIO	DATA LIMITE
28119	Lista AV1	16/02/2022 00:00	16/03/2022 00:00
27965	Primeiro acesso ao Beecrowd	02/02/2022 00:32	03/02/2022 10:00



4. Clique na atividade desejada que irá desenvolver e em seguida a tela com as questões será exibida. Na Figura exemplo, são mostrados o progresso da lista de exercícios, bem como todas as questões selecionadas pelo professor. Clique e selecione o exercício que será resolvido. De modo a exemplificar tal processo, será escolhido o segundo exercício, problema 1001 Extremamente Básico.

PROGRESSO						
<div></div>						
#		PROBLEMA	SUBMISSÃO	ACEITO	NÍVEL	PESO
1	1000	Hello World!	-	-	5	100
2	1001	Extremamente Básico	-	-	4	100
3	1002	Área do Círculo	-	-	4	100
4	1003	Soma Simples	-	-	1	100
5	1004	Produto Simples	-	-	1	100
6	1005	Média 1	-	-	2	100
7	1006	Média 2	-	-	1	100
8	1007	Diferença	-	-	1	100
9	1008	Salário	-	-	2	100
10	1009	Salário com Bônus	-	-	2	100
11	1011	Esfera	-	-	2	100
12	1014	Consumo	-	-	1	100
13	1016	Distância	-	-	1	100
14	1017	Gasto de Combustível	-	-	1	100
15	1018	Cédulas	-	-	4	100
16	1019	Conversão de Tempo	-	-	1	100
17	1020	Idade em Dias	-	-	2	100

5. Será apresentado as informações do problema, e os requisitos para aceitação do mesmo. A ferramenta beecrowd é criteriosa nesta aceitação, portanto, a leitura do problema deve ser feita com calma para que todos os detalhes do que é solicitado sejam entendidos. Nesta tela, também é possível verificar no painel direito, a presença do campo de submissão do seu algoritmo. Se as configurações foram realizadas corretamente, aparecerá um breve trecho de algoritmo padrão na linguagem **Javascript** como mostrado na Figura:

PROBLEMA

1001

LINGUAGEM

JavaScript

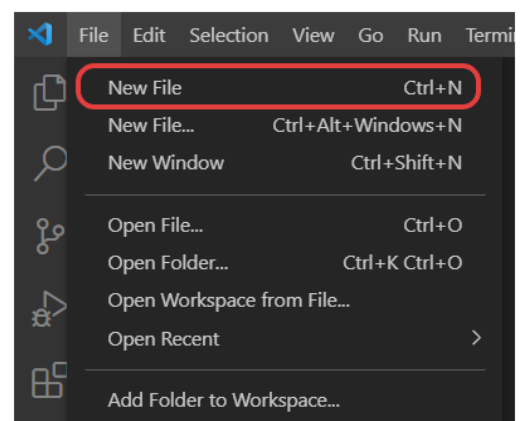
SOURCE CODE

```
1 var input = require('fs').readFileSync('/dev/stdin', 'utf8');
2 var lines = input.split('\n');
3
4 /**
5  * Escreva a sua solução aqui
6  * Code your solution here
7  * Escriba su solución aquí
8  */
9
```

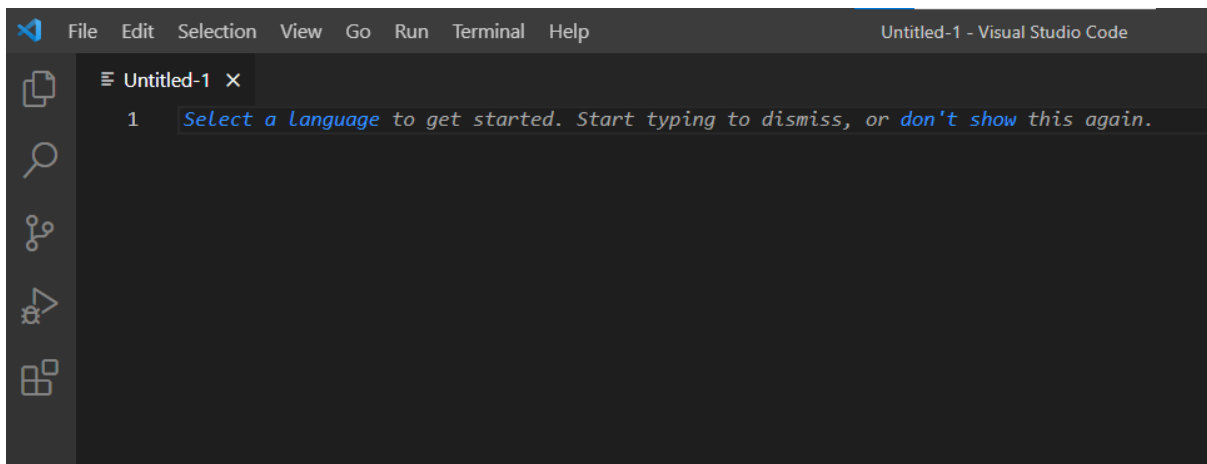
CONSTRUA A SUA SOLUÇÃO E ENVIE!

ENVIAR

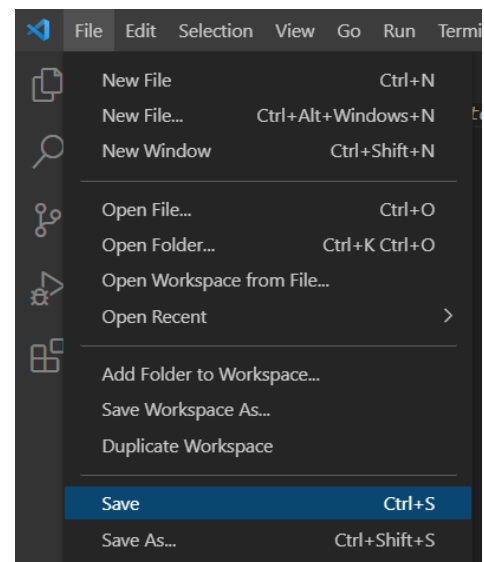
6. Após a leitura do problema, você deve então iniciar a implementação do seu algoritmo. É importante destacar que você **NÃO** realizará a implementação do código na ferramenta Beecrowd, pois, esta não se configura como uma ferramenta de desenvolvimento. Dito isto, abra o Visual Studio Code e clique em **File > New File** (ou aperte o atalho **CTRL+N** no teclado) como exibido na Figura que se segue:



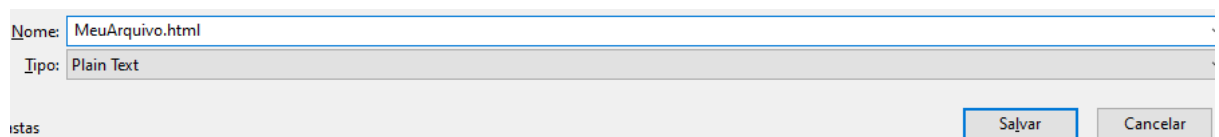
7. A seguinte tela deve aparecer, após a realização do passo 6:



8. Desta maneira um arquivo novo e sem nome é criado. O Visual Studio Code faz a identificação de tal arquivo com o símbolo indicado na aba do arquivo. Portanto, se este símbolo (📄) aparecer na aba em conjunto do nome “Untitled-1” significa que este arquivo ainda deve ser salvo com o nome de sua escolha e formato necessário. Dito isso, clique em File> Save (ou pelo atalho no teclado **CTRL+S**) para salvar o seu arquivo antes de iniciar a implementação do algoritmo.



9. É aconselhado que você salve este arquivo numa pasta de desejo, e escolha um nome qualquer para o arquivo, porém, é **NECESSÁRIO** que a extensão seja do tipo **.html** (ou **.HTML**). Logo, **TODOS** os arquivos que vocês salvarem ao longo do semestre terão essa extensão. Em seguida, clique em **Salvar**. Segue exemplo com um nome genérico de arquivo:



10. Pode-se perceber que após o procedimento anterior, algumas informações estarão modificadas. A primeira é que o símbolo (☰) desapareceu e agora tem-se (<>). Este é um primeiro indicativo de que tudo está sendo realizado corretamente, e você pode iniciar a implementação do algoritmo. Portanto, faz-se necessário inicialmente a construção padrão em html para iniciar o desenvolvimento em Javascript. Segue exemplo de como iniciar a estrutura:

```
<meta charset="UTF-8">

<script>

</script>
```

Há uma confusão natural neste momento, pois, o algoritmo escrito não se trata de Javascript e sim de uma estrutura em html. É dito que esta é uma confusão natural, pois, desde o início do semestre será dito pelo professor que os algoritmos serão desenvolvidos na linguagem Javascript. A explicação da presença da estrutura do html é simples: todos seus algoritmos serão executados no navegador que você tenha preferência (exemplo: Chrome, Edge, Opera, FireFox, etc...). Desta maneira, é necessário informar via html para o seu navegador, qual é o bloco em que será executado seu algoritmo em Javascript. Por isso há a necessidade de escrever a estrutura **<script></script>** e todas as linhas de comando que estão entre > ... < serão na linguagem Javascript e podem ser executadas pelo navegador. Segue exemplo. Obs: Todo texto escrito em verde (qualquer texto entre /\* e \*/), trata-se de um comentário e não é executado pelo navegador.

```
<meta charset="UTF-8">

<script>

/*
Todas as linhas aqui
dentro serão desenvolvidas
em Javascript.
*/

</script>
```

11. Agora o algoritmo referente ao problema pode ser desenvolvido em Javascript. Como este material não se trata de uma explicação sobre lógica de programação e sim como guiá-lo para o desenvolvimento de um algoritmo e em sequência submetê-lo na ferramenta Beecrowd, segue a solução do problema 1001 escolhido no passo 4.

```
<meta charset="UTF-8">

<script>

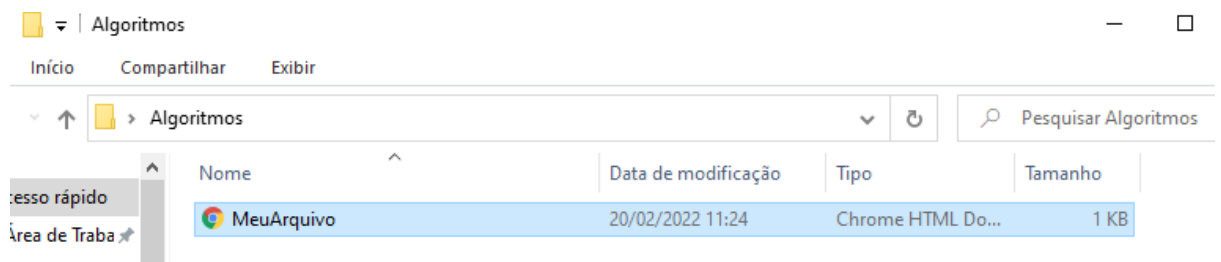
var A = parseInt(prompt('Digite o valor de A'))
var B = parseInt(prompt('Digite o valor de A'))

X = A+B

console.log('X = '+X)

</script>
```

12. Após a implementação do algoritmo, é interessante verificar sua execução para constatar se ele está operando conforme o esperado. Logo, vá até o caminho destino onde o mesmo foi salvo, e dê um duplo clique com o botão esquerdo no ícone do navegador.



13. Realizando o passo 12 você poderá verificar se o algoritmo em Javascript desenvolvido, é executado em um html corretamente. Se esta execução estiver correta (sem erros de compilação ou de execução), o algoritmo poderá então ser submetido a ferramenta Beecrowd.

14. Para a submissão do algoritmo, alguns fatores devem ser considerados. Inicialmente vale lembrar que a estrutura completa exibida pelo algoritmo no passo 10, trata-se de um **HTML** que possui um bloco de script na linguagem **Javascript**. O beecrowd espera um Javascript puro, portanto, algumas adequações devem ser realizadas.

- 14.1. A primeira é que apenas as linhas de código que estão entre <script> e </script> serão enviadas ao beecrowd. Exemplo: Da implementação exibida no Passo 10 apenas as linhas dentro do bloco do script são copiadas para a ferramenta Becrowd, como ilustrado pela Figura:

The screenshot shows the Beecrowd submission interface. At the top, there are two fields: 'PROBLEMA' with the value '1001' and 'LINGUAGEM' with a dropdown menu showing 'JavaScript'. Below these is a 'SOURCE CODE' section with a text area containing the following JavaScript code:

```

1  var input = require('fs').readFileSync('/dev/stdin', 'utf8');
2  var lines = input.split('\n');
3
4
i 5  var A = parseInt(prompt('Digite o valor de A'))
i 6  var B = parseInt(prompt('Digite o valor de A'))
7
i 8  X = A+B
9
i 10 console.log('X = '+X)
11
12  /**
13   * Escreva a sua solução aqui
14   * Code your solution here
15   * Escriba su solución aquí
16   */
17

```

- 14.2. A segunda adequação está associada aos comandos de entrada. Isto, pois, o prompt é um comando executado no navegador, e como o Beecrowd espera o javascript puro, outro comando de entrada deveria ser utilizado. Para não conflitar com o que foi ensinado em sala de aula, faz-se uma atualização do comando no beecrowd. Portanto, a seguinte linha de código:
- ```
var prompt = function(texto) { return lines.shift();};
```
- deve ser adicionada à linha 3 no beecrowd. A Figura exhibe como deve ficar o algoritmo até o presente momento:

## SOURCE CODE

```

1 var input = require('fs').readFileSync('/dev/stdin', 'utf8');
2 var lines = input.split('\n');
3 var prompt = function(texto) { return lines.shift();};
4
5 var A = parseInt(prompt('Digite o valor de A'))
6 var B = parseInt(prompt('Digite o valor de A'))
7
8 X = A+B
9
10 console.log('X = '+X)
11
12 /**
13  * Escreva a sua solução aqui
14  * Code your solution here
15  * Escriba su solución aquí
16  */
17

```

É possível submeter o presente algoritmo ao clicar no botão “**ENVIAR**” no canto inferior direito.

Com isso, seu algoritmo entrará na fila de submissões e terá uma resposta após alguns segundos.

| SUBMISSÃO # 26370208 |                                                                                                               |
|----------------------|---------------------------------------------------------------------------------------------------------------|
| PROBLEMA:            | 1001 - Extremamente Básico                                                                                    |
| RESPOSTA:            | <span style="color: red;">●</span> <span style="color: orange;">●</span> <span style="color: green;">●</span> |
| LINGUAGEM:           | JavaScript (nodejs 8.4.0) [+2s] {beta}                                                                        |
| TEMPO:               | 0.000s                                                                                                        |
| TAMANHO:             | 391 Bytes                                                                                                     |
| MEMÓRIA:             | -                                                                                                             |
| SUBMISSÃO:           | 20/02/2022 12:26:04                                                                                           |

Se todos os requisitos solicitados no problema do beecrowd forem atendidos, a seguinte mensagem aparecerá na tela:

| SUBMISSÃO # 26370208 |                                        |
|----------------------|----------------------------------------|
| PROBLEMA:            | 1001 - Extremamente Básico             |
| RESPOSTA:            | <b>Accepted</b>                        |
| LINGUAGEM:           | JavaScript (nodejs 8.4.0) [+2s] {beta} |
| TEMPO:               | 0.824s                                 |
| TAMANHO:             | 391 Bytes                              |
| MEMÓRIA:             | -                                      |
| SUBMISSÃO:           | 20/02/2022 12:26:04                    |


- **Demais adequações para a ferramenta Beecrowd.**

Pode-se lembrar inicialmente que para **TODOS** os problemas em que se tenha algum tipo de entrada, o comando:

```
var prompt = function(texto) { return lines.shift();};
```

deve ser adicionado na terceira linha no Beecrowd e antes da sua implementação. A explicação deste comando, é que o prompt está sendo redefinido para o comando que o Beecrowd espera como ser de entrada.

Vale abrir outro parêntese na presente seção. Nem todos os problemas que exigem um comando de entrada tem o mesmo padrão. Por exemplo, o problema 1001 em questão possui duas entradas que estão em linhas diferentes. Pode-se verificar isto ao ler os exemplos de entrada na explicação do problema, exibido pela Figura que se segue:

beecrowd | 1001  
**Extremamente Básico**  
Adaptado por Neilor Tonin, URI  Brasil  
**Timelimit: 1**

---

Leia 2 valores inteiros e armazene-os nas variáveis **A** e **B**. Efetue a soma de **A** e **B** atribuindo o seu resultado na variável **X**. Imprima **X** conforme exemplo apresentado abaixo. Não apresente mensagem alguma além daquilo que está sendo especificado e não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá *"Presentation Error"*.

**Entrada**

A entrada contém 2 valores inteiros.

**Saída**

Imprima a mensagem "X = " (letra X maiúscula) seguido pelo valor da variável **X** e pelo final de linha. Cuide para que tenha um espaço antes e depois do sinal de igualdade, conforme o exemplo abaixo.

| <u>Exemplos de Entrada</u> | Exemplos de Saída |
|----------------------------|-------------------|
| 10<br>9                    | X = 19            |
| -10<br>4                   | X = -6            |
| 15<br>-7                   | X = 8             |

Entradas em linhas diferentes

Outro exemplo dos diferentes padrões de entrada, é o problema 1044 que possui também mais de uma entrada, porém todas em mesma linha, como é mostrado pela Figura que se segue:



beecrowd | 1044

## Múltiplos

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Leia 2 valores inteiros (A e B). Após, o programa deve mostrar uma mensagem "Sao Multiplos" ou "Nao sao Multiplos", indicando se os valores lidos são múltiplos entre si.

### Entrada

A entrada contém valores inteiros.

### Saída

A saída deve conter uma das mensagens conforme descrito acima.

| Exemplo de Entrada | Exemplo de Saída  |
|--------------------|-------------------|
| 6 24               | Sao Multiplos     |
| 6 25               | Nao sao Multiplos |

Entradas em mesma linha, separadas por um espaço

Para este caso, em que as múltiplas entradas estão todas em uma única linha, é necessária uma adequação além da descrita no passo 13.2. Tal procedimento é exibido no Algoritmo:

```
<meta charset="UTF-8">

<script>

var [A, B] = prompt('Digite as entradas separadas por espaço').split(' ')

A = parseInt(A)
B = parseInt(B)

</script>
```

Pode ser que você não entenda inicialmente o que está acontecendo neste algoritmo exibido. Provavelmente pelo simples motivo que você ainda não tem o conhecimento necessário. Porém, na lista 2 da AV1 o professor da disciplina explicará como esta execução acontece e, porque é necessário o uso dos colchetes nas variáveis A e B presentes na linha do prompt.

Ainda tratando de entrada de dados, existem problemas que se utilizam de muitos casos de teste e alertam sobre a finalização da entrada de dados com a notificação EOF (do inglês, End Of File). Normalmente quando o Beecrowd solicita isso, o problema está diretamente relacionado ao conteúdo de Repetições. Segue exemplo de um problema com este requisito:

O Brasil é o país sede da copa esse ano. Porém, há muitas pessoas protestando contra o governo. Em redes sociais é possível ver pessoas afirmando que não vai ter copa devido aos protestos.

Mas esses rumores de que não haverá copa são totalmente falsos, a presidente Dilma Rousseff já avisou: vai ter copa sim, e se reclamar vai ter duas!

## Entrada

A entrada contém vários casos de teste e termina com EOF. Cada caso de teste consiste de uma linha contendo o número **N** de reclamações sobre a copa encaminhadas para a presidente ( $0 \leq N \leq 100$ ).

## Saída

Para cada teste, a saída consiste de uma linha dizendo "vai ter copa!" caso não haja reclamações para a presidente. Caso haja reclamações, a saída deverá dizer "vai ter duas!".

É perceptível pelos critérios descritos, que a entrada é finalizada quando encontra o “final do arquivo” (EOF). Para este caso, deve-se realizar uma verificação para cada repetição do valor de entrada. Por exemplo, se valor lido em uma determinada rodada **NÃO** for um número, o programa deve parar. Em Javascript, há um comando para testar se uma variável é um número. No algoritmo que se segue, pode-se verificar a presença do comando `isNaN()`, o qual retorna *true* se o seu parâmetro não for um número ou retorna *false*, se seu parâmetro for um número.

```
<meta charset="UTF-8">
<script>

    var x = parseInt(prompt())
    if (isNaN(x)){
        console.log('Não é um número, portanto deve-se parar...')
    }else{
        console.log('É um número, então continue...')
    }

</script>
```

Por fim, você aprendeu em sala de aula que existem três comandos de **saída**:

- *alert()*
- *document.write()*
- *console.log()*

Entretanto, no Beecrowd para **TODOS** os problemas que seja solicitado a exibição de algum dado, a submissão somente será aceita, caso seja utilizado o comando *console.log()*. A explicação disso, é similar ao que foi explicado anteriormente, já que os comandos *alert()* e *document.write()* exibem algum tipo de informação na página do navegador (.html) e o Beecrowd espera o Javascript puro.