

# Apresentação do R e Rstudio

## R

- R é uma linguagem e um ambiente de desenvolvimento integrado, livre e gratuito;
- O software R disponibiliza recursos para vários tipos de análises:
  - Estatística descritiva: medidas de posição, dispersão, tabelas e gráficos;
  - Modelos lineares, não lineares, generalizados: estimação dos parâmetros, resíduos, métodos de diagnóstico;
  - Análise multivariada: Análise de componentes principais, clusters, fatorial;
  - Aprendizado de Máquina
  - DeepLearning,

## RStudio

- O RStudio é uma interface de desenvolvimento integrada (IDE) para a linguagem de programação R;
- Diferentemente do R em que todas as versões são gratuitas, o RStudio disponibiliza versões gratuitas e versões pagas.
- O [vs code](#) é outra IDE que permite utilizar códigos R.

## Instalação do R e RStudio

- A seguir é apresentado os passos para instalação do R para o sistema operacional Windows. Para outros sistemas operacionais, clique [aqui](#)
- **Passo a passo de instalação do R para Windows:**
  1. Acesse o site do CRAN (Comprehensive R Archive Network) para Windows [aqui](#)
  2. Baixe a versão mais recente do R para Windows.
  3. Execute o instalador e siga as instruções para completar a instalação.
- **Passo a passo de instalação do RStudio**
- O **RStudio** é uma IDE (Integrated Development Environment) para a linguagem R que oferece uma interface e ferramentas adicionais:
  1. Acesse o site de download do RStudio [aqui](#)
  2. Baixe a versão do RStudio Desktop correspondente ao seu sistema operacional.
  3. Execute o instalador e siga as instruções para completar a instalação.

## Apresentação do RStudio

### Telas

- Ele oferece várias telas e painéis para facilitar o desenvolvimento, análise de dados e criação de visualizações.

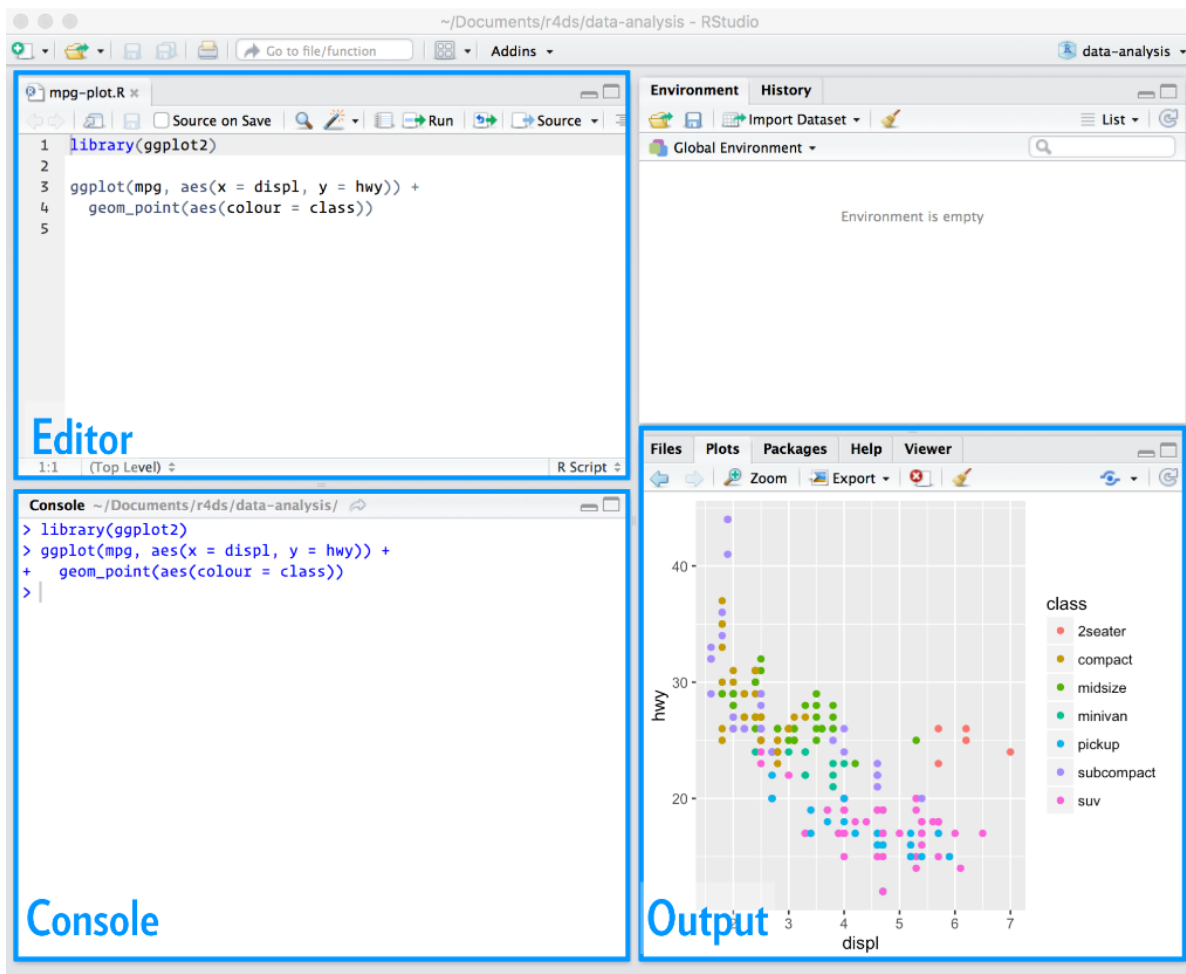


Figura 1: Painés do Rstudio

- Segue uma breve explicação das telas principais do RStudio.

### 1. Console:

- A tela principal onde você digita comandos R e recebe a saída imediata.

### 2. Script Editor:

- Onde você escreve e edita seus scripts em R. Os scripts são arquivos contendo código R que podem ser executados no console.

### 3. Environment/Painel de Ambiente:

- Exibe informações sobre objetos (variáveis, datasets) em sua sessão R atual.

4. **History/Histórico:**

- Mantém um histórico de comandos R previamente executados.

5. **Files/Painel de Arquivos:**

- Mostra os arquivos e diretórios do seu projeto. Facilita a navegação e a organização.

6. **Plots/Painel de Gráficos:**

- Exibe visualizações geradas a partir do R.

7. **Packages/Pacotes:**

- Gerencia pacotes R instalados e fornece ferramentas para instalação e atualização.

8. **Help/Ajuda:**

- Fornece acesso à documentação e ajuda sobre funções e pacotes R.

9. **Viewer/Visualizador:**

- Exibe visualizações mais complexas, como páginas web, no próprio IDE.

10. **Git:**

- Facilita a integração com sistemas de controle de versão Git, permitindo o rastreamento de alterações em projetos.

11. **Connections/Conexões:**

- Permite conectar-se a fontes de dados externas

12. **Terminal:**

- Um terminal embutido para execução de comandos do sistema operacional.

## **Cheatsheets**

- As “cheatsheets” (folhas de dicas) no RStudio são recursos visuais e resumidos que fornecem informações rápidas e úteis sobre tópicos específicos relacionados à linguagem de programação R, ao ambiente RStudio e a pacotes específicos do R.
- Para acessar as cheatsheets, você pode ir diretamente à [página de cheatsheets do RStudio](#)
- Também pode ser acessada clicando em Help -> Cheat Sheets

## Caminhos e Diretório de trabalho

### Caminhos

- Entende-se por **caminho** o endereço do arquivo no computador
- Existem duas formas de passarmos o caminho de arquivo: **caminho absoluto** ou **caminho relativo**

### Caminho Absoluto

- O **caminho absoluto** especifica o local exato de um arquivo desde a raiz do sistema de arquivo.
- O diretório raiz é o que está no topo da hierarquia do sistema, isso significa que outros diretórios estão contidos nele.



Figura 2: Caminho Absoluto

- `getwd()` é uma forma abreviada de dizer *get working directory* (me diga qual a pasta de trabalho atual).

```
getwd()
```

[1] "C:/Users/Leonardo\_Nascimento/Documents/GitHub/Introducao\_R/Cadernos"

- Observe que o local de referência é a pasta “*Cadernos*”, pois este é o caminho absoluto para a pasta onde essa aula foi produzida. Vamos voltar para a pasta “*Introducao\_R*” utilizando o conceito de caminho relativo

## Caminho Relativo

- O **caminho relativo** é especificado em relação ao diretório de trabalho atual ou a outro local de referência. Assim, se você quiser acessar alguma base de dados na pasta “*dados*” partindo da pasta “*Cadernos*”, o caminho seria: `../dados/base_de_dados.formato`.
- Nesse caso, `../` é o comando para voltar uma pasta dentro do caminho e `./` representa a pasta/diretório atual.
- Caso você queira trocar o local referência da pasta “*caderno*” para a pasta “*Introducao\_R*”, use o seguinte código:

```
setwd("../") # trocando local referência ou diretório de trabalho
getwd()
```

```
[1] "C:/Users/Leonardo_Nascimento/Documents/GitHub/Introducao_R"
```

## Diretório de trabalho

- Basicamente, diretório de trabalho refere-se a uma pasta específica no sistema de arquivos do computador em que um programa está atualmente operando ou onde ele procura por arquivos para ler e salvar por padrão.
- No contexto do R, o diretório de trabalho é particularmente importante porque é o local onde o R procura por arquivos. Isso significa que, se você carregar ou salvar um arquivo sem especificar a pasta, o R assumirá que o arquivo está no diretório de trabalho.
- É no diretório de trabalho que o R procura os arquivos que você pede para carregar e onde ele coloca todos os arquivos que você pede para salvar. De forma geral, é o local que está localizada sua análise.
- RStudio mostra seu diretório de trabalho atual na parte superior do console:

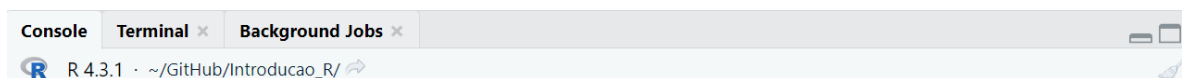


Figura 3: Diretório

- Você também pode visualizar o diretório de trabalho atual usando a função `getwd()`.
- Você também pode definir o diretório de trabalho dentro do R utilizando a função `setwd("colocar_o_caminho")`.

## Observações

- Caminhos e diretórios são um pouco complicados porque existem dois estilos básicos de caminhos: Mac/Linux e Windows. Existem três maneiras principais pelas quais eles diferem
  1. Como você separa os componentes do caminho. Mac e Linux usam barras (por exemplo `plots/diamonds.pdf`) e o Windows usa barras invertidas (por exemplo `plots\diamonds.pdf`). R pode funcionar com qualquer tipo. Porém, barras invertidas significam algo especial para R, e para obter uma única barra invertida no caminho, você precisa digitar duas barras invertidas! Recomendo sempre usar o estilo Linux/Mac com barras.
  2. Ao criar scripts ou projetos em R, é uma boa prática **usar caminhos relativos**, pois isso torna o código mais portátil e facilita a colaboração. Além disso, evita problemas quando você compartilha seu código com outros ou move seu projeto para um novo sistema operacional.
  3. A última pequena diferença é o local para onde `~` aponta. `~` é um atalho conveniente para o seu diretório inicial. No Windows, ele aponta para o diretório de documentos.

**Tabela 1: Localização de pastas no Linux**

	Nome fantasia	Localização real
Pasta pessoal		<code>/home/estudante</code> ou <code>~/</code>
Área de trabalho		<code>/home/estudante/Desktop</code>
Pendrive		<code>/media/RotuloDoPendrive</code>

**Tabela 2: Localização de pastas em versão recente do Windows**

	Nome fantasia	Localização real
Pasta pessoal		<code>C:/Users/estudante/Meus documentos</code> ou <code>~/</code>
Área de trabalho		<code>C:/Users/estudante/Desktop</code> ou <code>~/../Desktop</code>
Pendrive		<code>F:/</code> (ou outra letra)

- Nesses casos, criar projetos no R é uma prática recomendada e traz vários benefícios para a organização, colaboração e reprodutibilidade do trabalho
- Crie um diretório onde você possa colocar todos os programas/projetos em R

## Projetos

- Projetos no RStudio são uma maneira organizada e estruturada de trabalhar em análises de dados e programação em R ou em qualquer linguagem de programação suportada pelo RStudio.

- Aqui estão algumas razões pelas quais criar projetos no RStudio é importante:

1. **Organização Estruturada:**

- Projetos fornecem uma estrutura organizada para seus arquivos, dados e scripts. Isso facilita a navegação e a localização de recursos específicos relacionados ao projeto.

2. **Reprodutibilidade:**

- Ao criar um projeto, você organiza seu trabalho de forma que seja mais fácil reproduzir suas análises. Outros usuários podem reproduzir facilmente os resultados, garantindo a reprodutibilidade do trabalho.

3. **Configuração do Diretório de Trabalho:**

- O projeto define automaticamente o diretório de trabalho, eliminando a necessidade de definir caminhos relativos em seus scripts. Isso reduz erros e torna o código mais portátil.

4. **Controle de Versão:**

- Projetos podem ser vinculados a sistemas de controle de versão, como o Git. Isso facilita o controle de alterações em seus scripts e colaboração em equipes.

5. **Ambiente Isolado:**

- Projetos têm seu próprio ambiente no RStudio. Isso significa que as bibliotecas (pacotes) e as configurações específicas do projeto não interferem em outros projetos ou em seu ambiente global.

- A seguir são apresentados alguns passos para criar um projeto no RStudio.

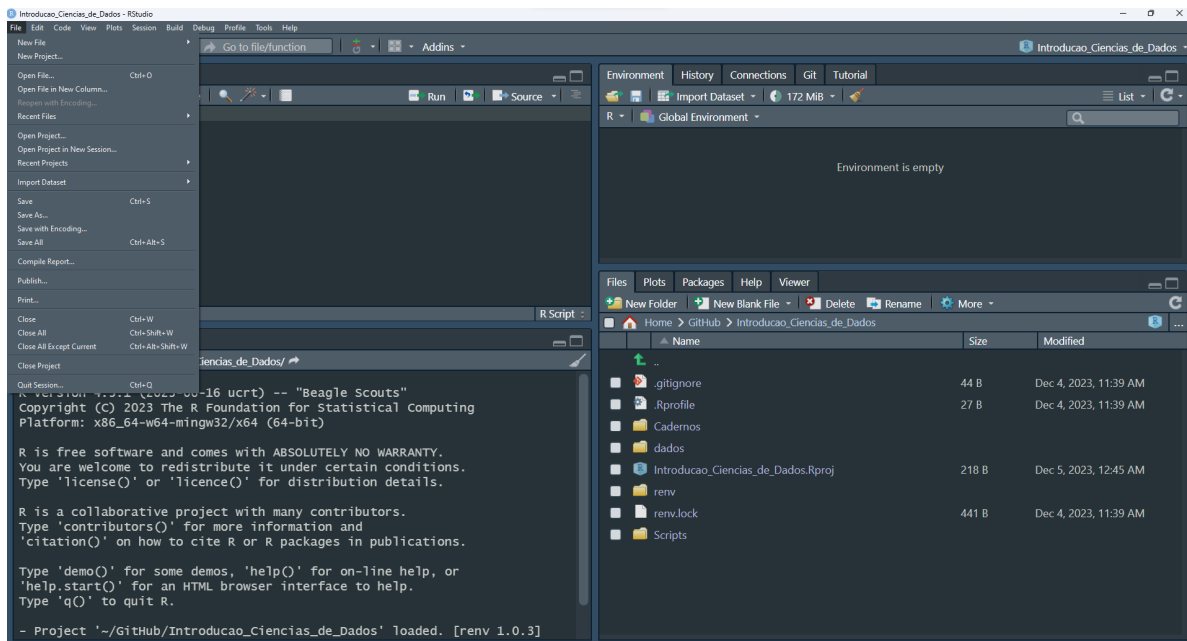
- **Passo 1:** Abra o RStudio

- Inicie o RStudio em seu computador.

- **Passo 2:** Crie um Novo Projeto

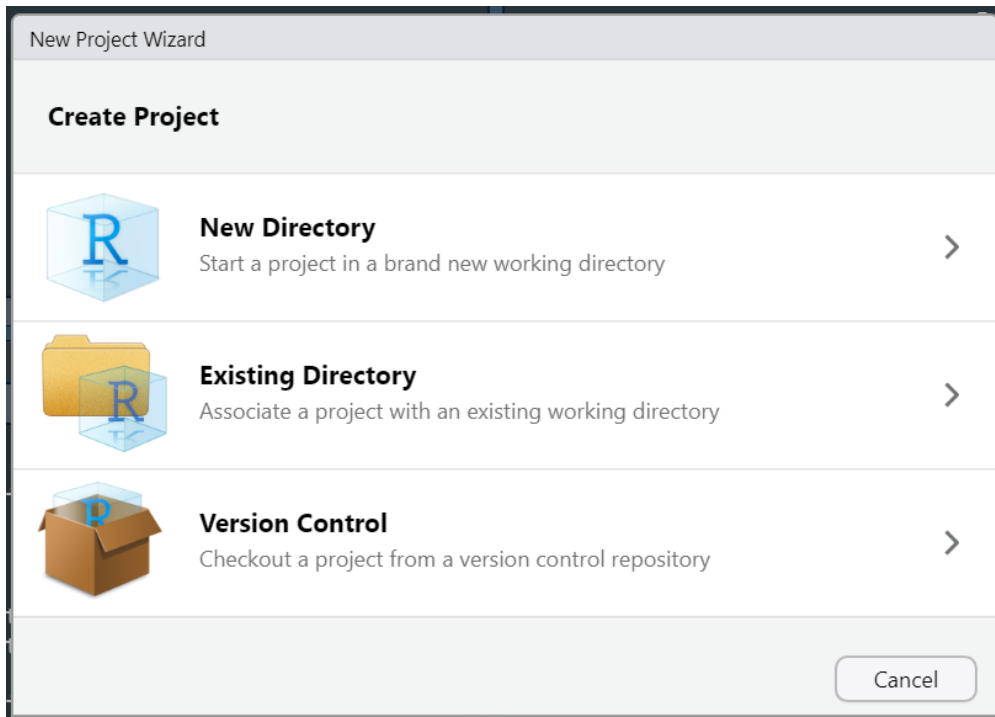
- No canto superior direito do RStudio, clique em “File” (Arquivo) e selecione “New Project” (Novo Projeto).





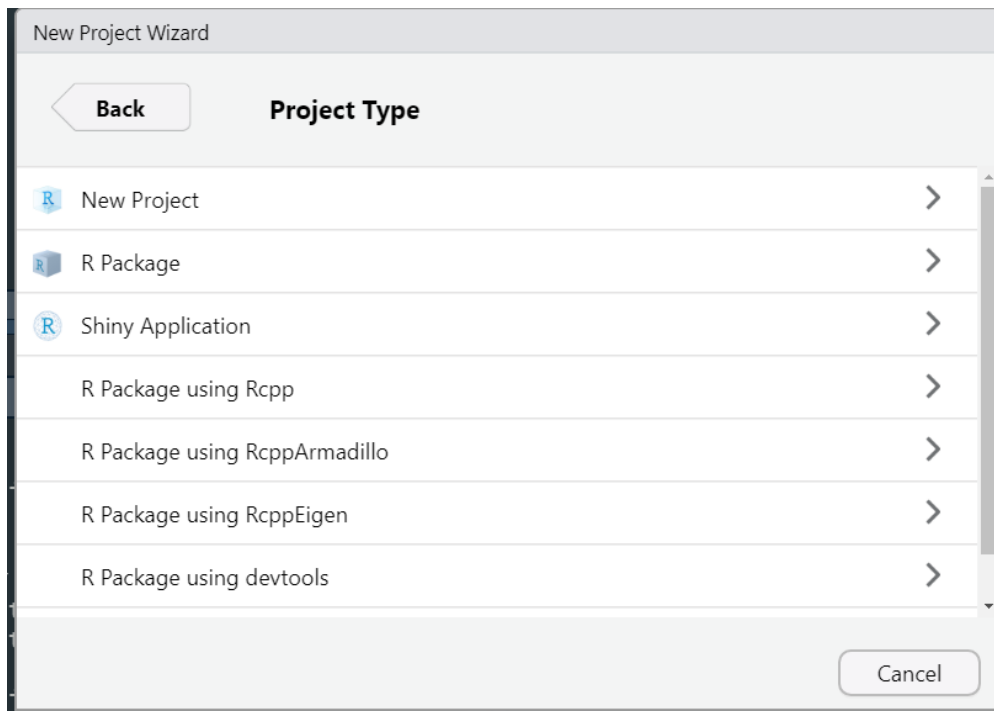
- **Passo 3:** Escolha o Tipo de Projeto

1. New Directory (Novo Diretório): Cria um novo diretório para o projeto.
2. Existing Directory (Diretório Existente): Usa um diretório existente como projeto.

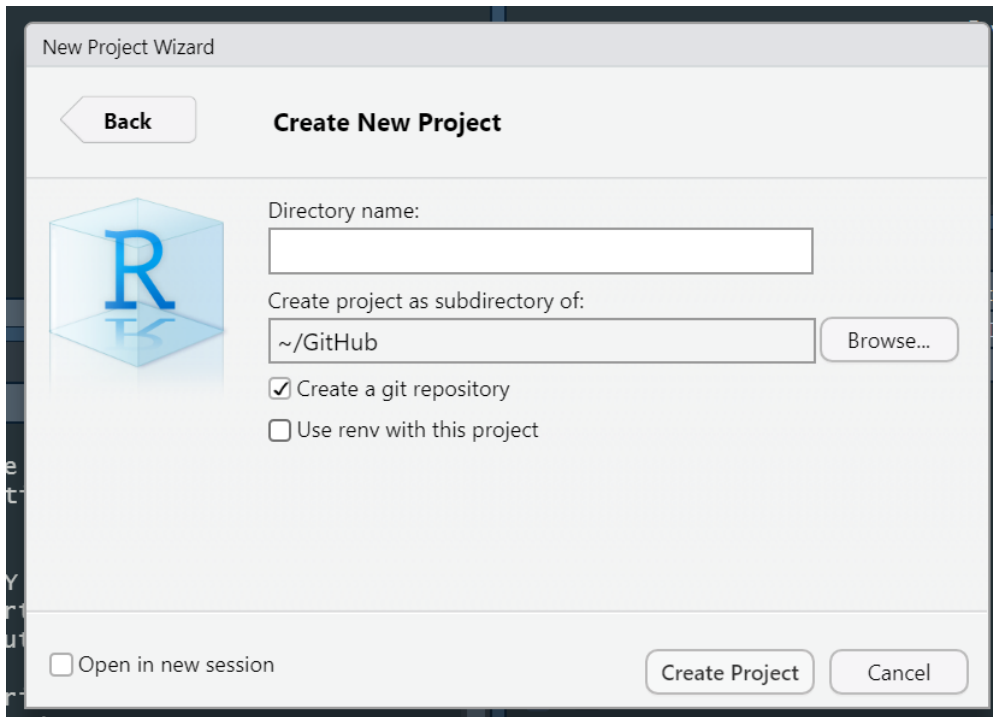


- **Passo 4:** Escolha um Modelo

- Você pode escolher um modelo específico se estiver disponível. Caso contrário, escolha “New Project” (Novo Projeto).



- **Passo 5:** Nomeie e Localize o Projeto
  - Dê um nome ao seu projeto e escolha o local onde o diretório do projeto será armazenado.



- **Passo 6:** Clique em “Create Project”
  - Clique em “Create Project” (Criar Projeto) para finalizar a criação do projeto.
- Após criar o projeto, você verá a estrutura do diretório do projeto no painel inferior direito do RStudio.
- Organize seus Arquivos. Dentro do diretório do projeto, você pode criar subdiretórios (por exemplo, “data”, “scripts”, “reports”) para organizar seus arquivos.
- Tudo o que você precisa está em um só lugar e bem separado de todos os outros projetos nos quais você está trabalhando.

## Configuração

- Basicamente, podemos definir uma **pesquisa reproduzível** como uma pesquisa que documenta todas as etapas entre os dados brutos e os resultados de uma forma que possa ser verificada.
- Isso envolve escrever scripts que realizem algumas análises do início ao fim de forma completa e transparente, de maneira que produza o mesmo resultado para pessoas diferentes usando o mesmo software em computadores diferentes.

- Nesse caso, é **recomendado** realizar dois ajustes na configuração do RStudio para maximizar a reprodutibilidade. Será desabilitado `.RData` e `.Rhistory`.
- O primeiro armazena todos os objetos gerados durante uma sessão R, enquanto o segundo mantém uma lista dos comandos mais recentemente executados.
- Ao reabrir o RStudio, o conteúdo desses arquivos é carregado no ambiente de trabalho atual, proporcionando a sensação de continuidade.
- Selecione **Tools > Global Options...** na aba de ferramentas do RStudio e então ajustar as configurações.
- A página de configurações gerais deve apresentar semelhanças com a imagem a seguir:

