

## Arvore Binaria Com Endereçamento Dinamico

Generated by Doxygen 1.9.1



---

<b>1 Class Index</b>	<b>1</b>
1.1 Class List . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 ArvoreBinaria Struct Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.1.2 Member Data Documentation . . . . .	5
3.1.2.1 raiz . . . . .	5
3.1.2.2 tamanho . . . . .	6
3.2 NoArvore Struct Reference . . . . .	6
3.2.1 Detailed Description . . . . .	6
3.2.2 Member Data Documentation . . . . .	6
3.2.2.1 direito . . . . .	6
3.2.2.2 esquerdo . . . . .	6
3.2.2.3 matricula . . . . .	7
3.2.2.4 nome . . . . .	7
<b>4 File Documentation</b>	<b>9</b>
4.1 ArvoreBinariaDinamica.h File Reference . . . . .	9



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">ArvoreBinaria</a>	Representa uma árvore binária com raiz e tamanho . . . . .	<a href="#">5</a>
<a href="#">NoArvore</a>	Representa um nó da árvore binária . . . . .	<a href="#">6</a>



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">ArvoreBinariaDinamica.h</a>	
Definições e estruturas para a manipulação de uma árvore binária . . . . .	9





## Chapter 3

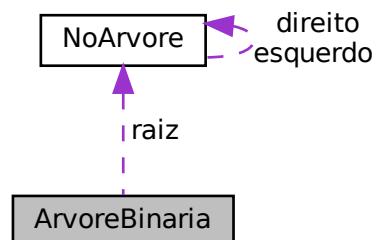
# Class Documentation

### 3.1 ArvoreBinaria Struct Reference

Representa uma árvore binária com raiz e tamanho.

```
#include <ArvoreBinariaDinamica.h>
```

Collaboration diagram for ArvoreBinaria:



#### Public Attributes

- `NoArvore *` `raiz`
- `int` `tamanho`

#### 3.1.1 Detailed Description

Representa uma árvore binária com raiz e tamanho.

#### 3.1.2 Member Data Documentation

### 3.1.2.1 raiz

`ArvoreBinaria::raiz`

Ponteiro para o nó raiz da árvore.

### 3.1.2.2 tamanho

`ArvoreBinaria::tamanho`

Número de nós presentes na árvore.

The documentation for this struct was generated from the following file:

- [ArvoreBinariaDinamica.h](#)

## 3.2 NoArvore Struct Reference

Representa um nó da árvore binária.

```
#include <ArvoreBinariaDinamica.h>
```

Collaboration diagram for NoArvore:



### Public Attributes

- long long int [matricula](#)
- [string](#) [nome](#)
- struct [NoArvore](#) \* [direito](#)
- struct [NoArvore](#) \* [esquerdo](#)

### 3.2.1 Detailed Description

Representa um nó da árvore binária.

### 3.2.2 Member Data Documentation

### 3.2.2.1 direito

`NoArvore::direito`

Ponteiro para o filho à direita.

### 3.2.2.2 esquerdo

`NoArvore::esquerdo`

Ponteiro para o filho à esquerda.

### 3.2.2.3 matricula

`NoArvore::matricula`

Matrícula do aluno.

### 3.2.2.4 nome

`NoArvore::nome`

Nome do aluno.

The documentation for this struct was generated from the following file:

- [ArvoreBinariaDinamica.h](#)



## Chapter 4

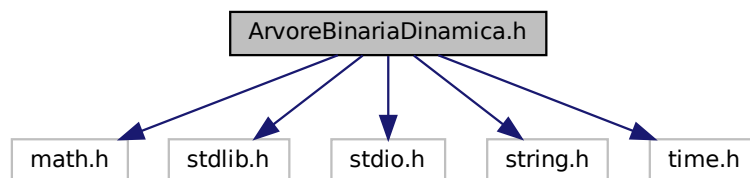
# File Documentation

### 4.1 ArvoreBinariaDinamica.h File Reference

Definições e estruturas para a manipulação de uma árvore binária.

```
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
```

Include dependency graph for ArvoreBinariaDinamica.h:



#### Classes

- struct [NoArvore](#)  
*Representa um nó da árvore binária.*
- struct [ArvoreBinaria](#)  
*Representa uma árvore binária com raiz e tamanho.*

#### Typedefs

- typedef char [string](#)[101]
- typedef clock\_t [processTime](#)
- typedef struct [NoArvore](#) **NoArvore**

## Functions

- FILE \* [abrirArquivo](#) (char \*nomeArq, char \*modo)  
*Abre um arquivo no modo especificado.*
- void [calcularTempo](#) (double ini, double fim)  
*Calcula e exibe o tempo de execução.*
- void [salvarDadosNoArquivo](#) ([NoArvore](#) \*no, FILE \*arquivoLista)  
*Salva os dados de um nó em um arquivo.*
- void [inicializarArvore](#) ([ArvoreBinaria](#) \*arvore)  
*Inicializa uma árvore binária.*
- void [liberarArvore](#) ([NoArvore](#) \*no)  
*Libera a memória alocada para a árvore.*
- void [imprimirEmOrdem](#) ([NoArvore](#) \*no)  
*Imprime a árvore binária em ordem.*
- [NoArvore](#) \* [inserirNo](#) ([NoArvore](#) \*no, long long int matricula, char \*nome)  
*Insere um novo nó na árvore binária.*
- [NoArvore](#) \* [buscarNo](#) ([NoArvore](#) \*no, long long int matricula)  
*Busca um nó pela matrícula.*
- [NoArvore](#) \* [removerNo](#) ([NoArvore](#) \*no, long long int matricula)  
*Remove um nó da árvore binária.*
- [NoArvore](#) \* [encontrarMinimo](#) ([NoArvore](#) \*no)  
*Encontra o menor valor em uma subárvore.*
- void [lerEInserirMatriculas](#) ([ArvoreBinaria](#) \*arvore, FILE \*arquivoLista)  
*Lê as matrículas de um arquivo e insere na árvore.*
- void [menuPrincipal](#) ([ArvoreBinaria](#) \*arvore)  
*Função principal do menu de operações.*
- int [contarMatriculas](#) (FILE \*arquivoLista)  
*Conta o número de matrículas em um arquivo.*

### 4.1.1 Detailed Description

Definições e estruturas para a manipulação de uma árvore binária.

### 4.1.2 Typedef Documentation

#### 4.1.2.1 processTime

```
typedef clock_t processTime
```

Define um tipo para medir o tempo de processamento

#### 4.1.2.2 string

```
typedef char string[101]
```

Define um tipo string com tamanho 101

### 4.1.3 Function Documentation

#### 4.1.3.1 abrirArquivo()

```
FILE* abrirArquivo (
    char * nomeArq,
    char * modo )
```

Abre um arquivo no modo especificado.

##### Parameters

<i>nomeArq</i>	Nome do arquivo a ser aberto.
<i>modo</i>	Modo de abertura (ex: "r" para leitura).

##### Returns

Ponteiro para o arquivo.

#### 4.1.3.2 buscarNo()

```
NoArvore* buscarNo (
    NoArvore * no,
    long long int matricula )
```

Busca um nó pela matrícula.

##### Parameters

<i>no</i>	Ponteiro para o nó da árvore.
<i>matricula</i>	Matrícula a ser buscada.

##### Returns

Ponteiro para o nó encontrado.

#### 4.1.3.3 calcularTempo()

```
void calcularTempo (
    double ini,
    double fim )
```

Calcula e exibe o tempo de execução.

**Parameters**

<i>ini</i>	Tempo inicial.
<i>fim</i>	Tempo final.

**4.1.3.4 contarMatriculas()**

```
int contarMatriculas (
    FILE * arquivoLista )
```

Conta o número de matrículas em um arquivo.

**Parameters**

<i>arquivoLista</i>	Ponteiro para o arquivo.
---------------------	--------------------------

**Returns**

Número total de matrículas.

Conta o número de matrículas em um arquivo.

**4.1.3.5 encontrarMinimo()**

```
NoArvore* encontrarMinimo (
    NoArvore * no )
```

Encontra o menor valor em uma subárvore.

**Parameters**

<i>no</i>	Ponteiro para o nó da árvore.
-----------	-------------------------------

**Returns**

Ponteiro para o nó com o menor valor.

Encontra o menor valor em uma subárvore.

**4.1.3.6 imprimirEmOrdem()**

```
void imprimirEmOrdem (
    NoArvore * no )
```

Imprime a árvore binária em ordem.



## Parameters

<i>no</i>	Ponteiro para o nó da árvore a ser impresso.
-----------	--

Imprime a árvore binária em ordem.

#### 4.1.3.7 inicializarArvore()

```
void inicializarArvore (  
    ArvoreBinaria * arvore )
```

Inicializa uma árvore binária.

## Parameters

<i>arvore</i>	Ponteiro para a árvore binária a ser inicializada.
---------------	--

Inicializa uma árvore binária.

#### 4.1.3.8 inserirNo()

```
NoArvore* inserirNo (  
    NoArvore * no,  
    long long int matricula,  
    char * nome )
```

Insere um novo nó na árvore binária.

## Parameters

<i>no</i>	Ponteiro para o nó da árvore.
<i>matricula</i>	Matrícula a ser inserida.
<i>nome</i>	Nome do aluno.

## Returns

Ponteiro para o nó inserido.

#### 4.1.3.9 lerEInserirMatriculas()

```
void lerEInserirMatriculas (  
    ArvoreBinaria * arvore,  
    FILE * arquivoLista )
```

Lê as matrículas de um arquivo e insere na árvore.

## Parameters

<i>arvore</i>	Ponteiro para a árvore binária.
<i>arquivoLista</i>	Ponteiro para o arquivo contendo as matrículas.

Lê as matrículas de um arquivo e insere na árvore.

**4.1.3.10 liberarArvore()**

```
void liberarArvore (
    NoArvore * no )
```

Libera a memória alocada para a árvore.

## Parameters

<i>no</i>	Ponteiro para o nó a ser liberado.
-----------	------------------------------------

Libera a memória alocada para a árvore.

**4.1.3.11 menuPrincipal()**

```
void menuPrincipal (
    ArvoreBinaria * arvore )
```

Função principal do menu de operações.

## Parameters

<i>arvore</i>	Ponteiro para a árvore binária.
---------------	---------------------------------

Função principal do menu de operações.

**4.1.3.12 removerNo()**

```
NoArvore* removerNo (
    NoArvore * no,
    long long int matricula )
```

Remove um nó da árvore binária.

## Parameters

<i>no</i>	Ponteiro para o nó da árvore.
<i>matricula</i>	Matrícula do nó a ser removido.

#### Returns

Ponteiro para o nó removido.

#### 4.1.3.13 salvarDadosNoArquivo()

```
void salvarDadosNoArquivo (
    NoArvore * no,
    FILE * arquivoLista )
```

Salva os dados de um nó em um arquivo.

#### Parameters

<i>no</i>	Ponteiro para o nó da árvore.
<i>arquivoLista</i>	Ponteiro para o arquivo de saída.

Salva os dados de um nó em um arquivo.

