

horaTrabalhada

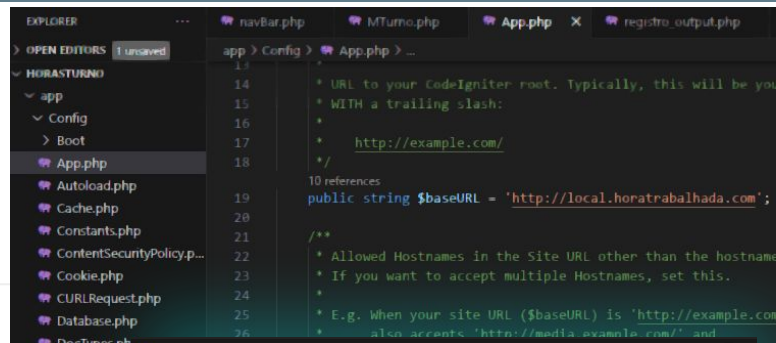
Cálculo básico de período de trabalho



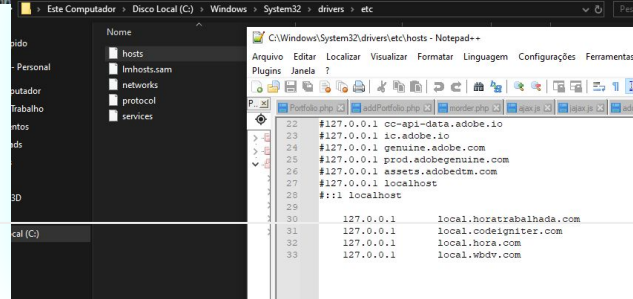
Instalação XAMPP

1. Após baixar o arquivo, extrair-lo na pasta do servidor, alterar no arquivo App/Config/App.php

```
public string $baseURL = 'http://local.horatrabalhada.com';
```



2. Configurar virtual hosts windows: C:\Windows\System32\drivers\etc adicionar linha:
127.0.0.1 local.horatrabalhada.com

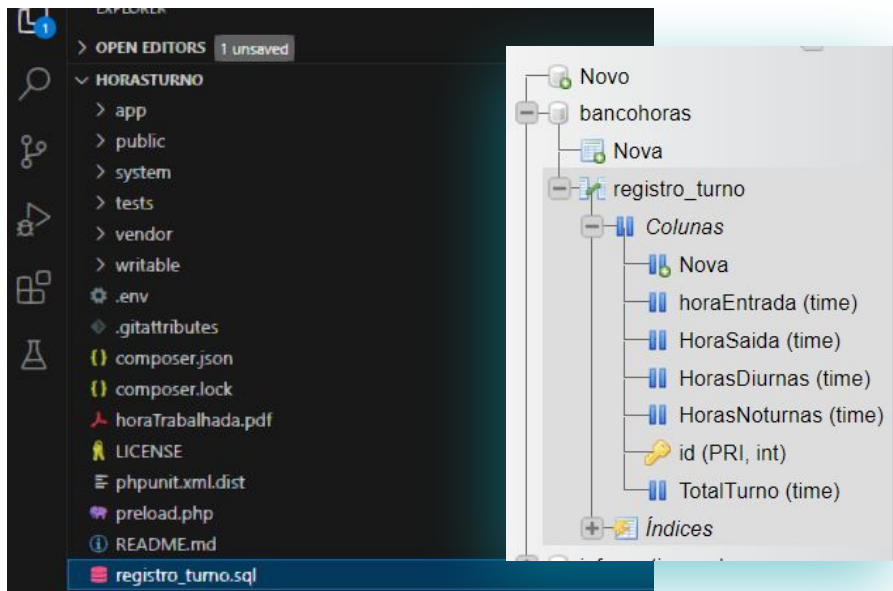


C:\xampp\apache\conf\extra\httpd-vhosts.conf adicionar:

3.

```
<VirtualHost *:80>
    ServerAdmin webmaster@local.horatrabalhada.com
    DocumentRoot "C:/xampp/htdocs/horasTurno/public"
    ServerName local.horatrabalhada.com
    ErrorLog "logs/horatrabalhada.com-error.log"
    CustomLog "logs/horatrabalhada.com-access.log" common
</VirtualHost>
```

MySQL - Opcional



diretório raiz do projeto & tabela bancohoras

Caso queira utilizar banco de dados na sua implementação do projeto, é possível acessar o método de leitura em:

model: MTurno::read_registros

Por meio do controller:

Horatrabalhada->read_registros

Importar também o SQL que está na raiz do projeto,

-Caso não aconteça a importação realize a criação de um banco chamado bancodeHora e tente novamente .

Acessar em <http://local.horatrabalhada.com>

[Guia de Instalação máquina bitnami codeigniter 4](#)

[Guia instalação codeigniter 4 site oficial](#)

/app/Controllers/Horatrabalhada.php

```
1  <?php
2  namespace App\Models;
3
4  use CodeIgniter\Model;
5  use CodeIgniter\I18n\Time;
6  class MTurno extends Model
7  {
8      public Time $horaEntrada;
9      public Time $horaSaida;
10     public Time $periodoDiurno;
11     public Time $periodoNoturno;
12     var $turno;
13
14     protected $table = 'registro_turno';
15     protected $allowedFields = ['id', ...
16                                     'TotalTurno'];
17     public function montaTurno(){...
```

horasTurno/app/Controllers/Horatrabalhada.php

Responsável por acessar o model podendo criar objetos do tipo Turno que são usados nas operações de comparação que ocorrem nos models.

HoraTrabalhada->montaTurnoCall()

Implementa a comparação de horas verificando os valores de da hora de entrada e classificando como horas diurnas ou noturnas, adicionando neste momento os minutos referentes a diferença do período até a primeira hora de trabalho concluída. Será explicado no slide 8.

Existem também algumas funções que não estão sendo utilizadas que serviram como base para o aprendizado necessário para a resolução do problema..

HoraTrabalhada/index

Página inicial, onde é possível receber os valores de entrada e saída do turno para o cálculo que será realizado.

Controller

-HoraTrabalhada.php/index()

View

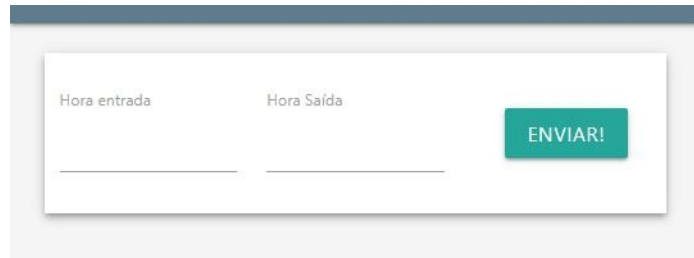
-inputHora.php

Call

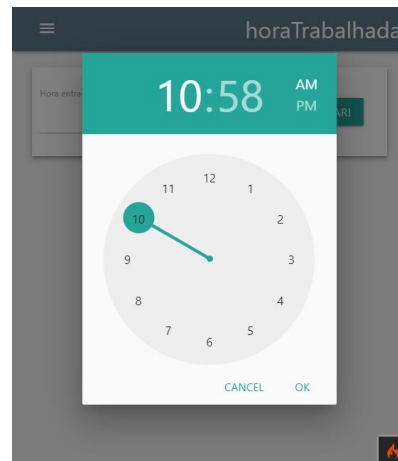
-horaTrabalhada/montaTurnoCall()

Recebendo valores de

`$_POST['$horaEntrada', '$horaSaida'];`

A screenshot of a web form titled 'HoraTrabalhada/index'. It features two input fields: 'Hora entrada' and 'Hora Saída'. To the right of these fields is a green button with the text 'ENVIAR!'. The form is set against a light gray background with a subtle shadow.

horasTurno/app/Views/input_hora.php



Relógio utiliza JQuery na implementação

HoraTrabalhada/montaTurnoCall()

Página inicial, onde é possível receber os valores de entrada e saída do turno para o cálculo que será realizado.

Controller

HoraTrabalhada.php/montaTurnoCall()

Acessa o model MTurno, possibilitando a criação do objeto.

```
0 references | 0 overriden
48 public function montaTurnoCall(){
49     $turno = model(MTurno::Class);
50     $turno->montaTurno();
51     $data = [
52         'horarioEntrada' => $turno->horaEntrada,
53         'horarioSaida' => $turno->horaSaida,
54         'periodoTurno' => $turno->turno,
55         'periodoDiurno' => $turno->periodoDiurno,
56         'periodoNoturno' => $turno->periodoNoturno
57     ];
58     // ($turno);
59     return view('navBar') . view('output_comparaHora', $data);
60     // var_dump($turno->turno->h);
61 }
```

horasTurno/app/Controllers/Horatrabalhada.php

View

navbar

output_comparaHora.php

Call

-FUTURAMENTE

horasTurno/app/Views/output_comparaHora.php

/app/Models/MTurno.php

Responsável por implementar o objeto turno, por se tratar de um Model, já existe. Uma

Atributos:

Time horaEntrada

Time horaSaida

Time PeriodoDiurno

Time PeriodoNoturno

Var turno

Métodos:

montaTurno(){}:

comparaHoras(){}:

save_turno(){};*

loadAllTurnos(){};

analiseHora(){};

classificaHora();

pegaHora();*

```
namespace App\Models;

use CodeIgniter\Model;
use CodeIgniter\I18n\Time;
4 references | 0 implementations
class MTurno extends Model
{
    12 references
    public Time $horaEntrada;
    7 references
    public Time $horaSaida;
    9 references
    public Time $periodoDiurno;
    8 references
    public Time $periodoNoturno;
    5 references
    var $turno;

    18 references
    protected $table = 'registro_turno';
    7 references
    > protected $allowedFields = ['id', ...
    'TotalTurno'];

    1 reference | 0 overrides
    > public function montaTurno(){...
    }

    1 reference | 0 overrides
    > public function comparaHoras(){...
    }

    0 references | 0 overrides
    > protected function save_turno(){...
    }

    1 reference | 0 overrides
    public function loadAllTurnos($id = false)
    > {
    }
}
```

/app/models/MTurno::montaTurno()

```
1 reference|0 overrides
21 public function montaTurno(){
22
23     //variáveis presentes no arquivo
24     $horaAtual = $_POST['horaEntrada'];
25     $horaFinal = $_POST['horaSaida'];
26     //$this->horaSaida = $_POST['horaSaida'];
27
28     $this->periodoDiurno = new Time('00:00:00');
29     $this->periodoNoturno= new Time('00:00:00');
30
31     $horaAtual = new Time($horaAtual);
32     $horaFinal = new Time($horaFinal);
33
34     $this->horaEntrada = $horaAtual;
35     $this->horaSaida = $horaFinal;
36
37     $this->comparaHoras();
38
39     //$this->horaEntrada->getHour();
40     //var_dump($this->turno->h);
41     $data = ['horarioEntrada' => $this->horaEntrada,
42             'horarioSaida' => $this->horaSaida,
43             'periodoDiurno' => $this->periodoDiurno,
44             'periodoNoturno' => $this->periodoNoturno,
45             'periodoTurno' => $this->turno];// . $this->horaSaida;
46     return;
47     //view('navBar');
48     // view('output_comparaHora', $data);
49 }
50
```

horasTurno/app/Models/MTurno.php

MTurno::comparaHoras()

Implementa a comparação de horas verificando os valores de da hora de entrada e saída classificando-as como horas diurnas ou noturnas, adicionando neste momento os minutos referentes a diferença do período até a primeira hora de trabalho concluída,

Após isso acontece um loop onde é adicionado 1 a cada hora até chegarmos no horário final e também classificando como horas noturnas e diurnas.

Por fim o algoritmo realiza algumas operações utilizando comparações entre objetos Time e objetos DateTime que são atributos da classe Turno;

Algoritmo de comparação

```
51 1 reference | 0 overrides
52 public function comparaHoras()
53     //não confundir com horaTrabalhada->comparaHora() que é uma função que será removida
54     // $horaAtual= new Time($horaInicial);
55     // $horaFinal= new Time($horaInicial);
56
57     $diferencaMinutosIniciais = 60 - $this->horaEntrada->getMinute();
58     $diferencaHoras =
59     //echo '<br>' . $horaInicial->getMinute();
60     $horaAtual= new Time($this->horaEntrada);
61     $horaFinal = new Time($this->horaSaida);
62
63
64 > if($this->horaEntrada->getHour() >= $horaFinal->getHour()){ ...
65     }
66
67
68     //seta o valor de hora trabalhadas em um turno
69     $this->turno = $horaFinal->diff($this->horaEntrada);
70     //primeira hora
71
72 > if($this->horaEntrada->getHour() > 22 || $this->horaEntrada->getHour() <=5 && $this->horaEntrada->getMinute() != 0 && $controle =0){ ...
73 > }else{ ...
74
75
76     //echo verifica valores da hora final;
77 > if($this->horaEntrada->getHour() >= 22 || $this->horaSaida->getHour() <=5 ){ ...
78 > }else{ ...
79
80
81
82
83
84
85
86
87 > }
88
89
90 }
```

Algoritmo de comparação

```
83
84 //echo verifica valores da hora final;
85 > if($this->horaEntrada->getHour() >= 22 || $this->horaSaida->getHour() <=5 ){...
87 > }else{ ...
90 }
91 //loop
92 for($i = 0; $i < $this->turno->h -1 ; $i++){
93     if($horaAtual->getHour() >=22 || $horaAtual->getHour() < 5 ){
94         $this->periodoNoturno = $this->periodoNoturno->addHours(1);
95     }
96     $horaAtual = $horaAtual->addHours(1);
97 }
98
99 $aux = $this->turno->format('%H:%i');
100 $aux = new Time($aux);
101
102 $diff = $this->periodoNoturno->toTimeString();
103
104 $diff = new Time($diff);
105 $aux = $aux->diff($diff);
106 $aux = new Time($aux->format('%H:%i'));
107 $this->periodoDiurno = $aux;
108
109 //var_dump($aux);
110 // $horasDia = $this->turno->h - $this->periodoNoturno->getHour();
111 // $this->periodoDiurno = $this->periodoDiurno->addHours($horasDia);
112 //echo $treta2->format('%H:%i');
113
114 return;
115
```

Qualquer dúvida

henriquelopes@artebodoque.com

32988055272

Henrique Lopes