

Kronos

https://kronos/registrarPonto.html

Registrar Ponto / Tarefa

Ano

Mês

Pesquisar

RN003

Pendentes

RN011

Registrar Tarefa OnLine

RN010

Registrar Entrada

Dados de Pontos / Tarefas	RN001	Ações
[Data] [Período] [Entrada Prevista] [Hora Entrada] [Saída Prevista] [Hora Saída] [Qtde Horas] [Data Homologação] [Descrição tarefa]		[Registrar Entrada] [Registrar Saída] <div>RN002</div>
[Data] [Período] [Entrada Prevista] [Hora Entrada] [Saída Prevista] [Hora Saída] [Qtde Horas] [Data Homologação] [Descrição tarefa]		[Registrar Entrada] [Registrar Saída]

Registrar Entrada

Data

RN004

Período

RN007

Entrada Prevista

RN012

Saída Prevista

RN012

Hora Entrada

Cancelar

1

Confirmar

RN009

RN005

Registrar Saída

Data

RN004

Período

Entrada Prevista

Hora Entrada

Saída Prevista

Hora Saída

Cancelar

1

Confirmar

RN005

Registrar Tarefa OnLine

Descrição da tarefa

Data

RN004

Hora Início

Hora Fim

Cancelar

1

Confirmar

RN008

RN005

**RN001**

```
select p.data, t.nome as periodo, p.hora_inicial_registro, p.hora_final_registro,
p.hora_inicial_informado, p.hora_final_informado, p.horas_registrada, p.data_homologacao, p.descricao as tarefa,
CASE
    WHEN id_tipo_periodo_ponto = 1 THEN c.hora_inicial_p1
    WHEN id_tipo_periodo_ponto = 2 THEN c.hora_inicial_p2
    WHEN id_tipo_periodo_ponto = 3 THEN c.hora_inicial_p3
ELSE Null
END as hora_inicial_prevista,
CASE
    WHEN id_tipo_periodo_ponto = 1 THEN c.hora_final_p1
    WHEN id_tipo_periodo_ponto = 2 THEN c.hora_final_p2
    WHEN id_tipo_periodo_ponto = 3 THEN c.hora_final_p3
ELSE Null
END as hora_final_prevista,
CASE WHEN p.hora_inicial_informado is null or p.data_liberacao is not null THEN true else false END and
CASE WHEN p.data = current_date or p.data_liberacao is not null THEN true else false END and
CASE
    WHEN c.tolerancia = '00:00:00' THEN true
    WHEN p.data_liberacao is not null and p.hora_inicial_informado is null THEN true
    WHEN id_tipo_periodo_ponto = 1 and justify_hours(cast(current_time as time) - c.hora_inicial_p1) > tolerancia THEN false
    WHEN id_tipo_periodo_ponto = 2 and justify_hours(cast(current_time as time) - c.hora_inicial_p2) > tolerancia THEN false
    WHEN id_tipo_periodo_ponto = 3 and justify_hours(cast(current_time as time) - c.hora_inicial_p3) > tolerancia THEN false
    WHEN id_tipo_periodo_ponto = 4 THEN false
ELSE true
END as tolerancia_entrada,
CASE WHEN p.hora_final_informado is null or p.data_liberacao is not null THEN true else false END and -- as hoje,
CASE WHEN p.data = current_date or p.data_liberacao is not null THEN true else false END and -- as hoje,
CASE
    WHEN c.tolerancia = '00:00:00' THEN true
    WHEN p.data_liberacao is not null and p.hora_final_informado is null THEN true
    WHEN id_tipo_periodo_ponto = 1 and justify_hours(cast(current_time as time) - c.hora_final_p1) > tolerancia THEN false
    WHEN id_tipo_periodo_ponto = 2 and justify_hours(cast(current_time as time) - c.hora_final_p2) > tolerancia THEN false
    WHEN id_tipo_periodo_ponto = 3 and justify_hours(cast(current_time as time) - c.hora_final_p3) > tolerancia THEN false
    WHEN id_tipo_periodo_ponto = 4 THEN false
ELSE true
END as tolerancia_saida
from ponto p
left join config_ponto c on (p.id_config_ponto = c.id)
left join tipo_periodo_ponto t on (p.id_tipo_periodo_ponto = t.id)
where c.id_funcionario = 2 --idFuncionarioUsuarioLogado (Padrão quando acessa funcionalidade ou Pendentes)
-- and p.hora_final_informado is null -- Padrão quando acessa funcionalidade ou Pendentes
-- and extract(YEAR FROM p.data) = 2020 -- filtroAnoSelecionado
-- and extract(MONTH FROM p.data) = 4 -- filtroMesSelecionado
order by p.data desc, p.id_tipo_periodo_ponto desc
```

**RN008**

- Registra automaticamente id\_tipo\_ponto\_periodo = 0
- Registra id\_config\_ponto da seguinte consulta:

```
select id from config_ponto c
where current_date between c.data_inicial_vigencia and c.data_final_vigencia
and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
```

**RN009**

- Registra id\_config\_ponto da seguinte consulta:

```
select id from config_ponto c
where current_date between c.data_inicial_vigencia and c.data_final_vigencia
and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
```

**RN010**

- Somente habilitado se a consulta abaixo retornar true

```
select
CASE
    WHEN ((select count(*) from ponto p where p.id_tipo_periodo_ponto = 1 and p.id_config_ponto = c.id and p.data = current_date) = 0)
    and (c.hora_inicial_p1 is not null)
    and (c.tolerancia = '00:00:00' or justify_hours(cast(current_time as time) - c.hora_inicial_p1) <= tolerancia) THEN true
    WHEN ((select count(*) from ponto p where p.id_tipo_periodo_ponto = 2 and p.id_config_ponto = c.id and p.data = current_date) = 0)
    and (c.hora_inicial_p2 is not null)
    and (c.tolerancia = '00:00:00' or justify_hours(cast(current_time as time) - c.hora_inicial_p2) <= tolerancia) THEN true
    WHEN ((select count(*) from ponto p where p.id_tipo_periodo_ponto = 3 and p.id_config_ponto = c.id and p.data = current_date) = 0)
    and (c.hora_inicial_p3 is not null)
    and (c.tolerancia = '00:00:00' or justify_hours(cast(current_time as time) - c.hora_inicial_p3) <= tolerancia) THEN true
    ELSE false
END as habilitado
from config_ponto c
where current_date between c.data_inicial_vigencia and c.data_final_vigencia
and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
```

**RN002**

- [Registrar Entrada] -> tolerancia\_entrada de RN001 = true
- [Registrar Saída] -> tolerancia\_saída de RN001 = true

**RN003**

- Lista somente os pendentes
- condição "and p.hora\_final\_informado is null" de RN001

**RN004**

- Captura data do servidor, somente na inclusão

**RN005**

- Captura e registra hora do servidor no campo hora\_inicial\_registro para entrada e hora\_finnal\_registro para saída quando usuário informa os respectivos horários

**RN007**

- Quando o registro vier oriundo de [Registrar Entrada] de Ações o período deve ser pré-fixado em função do registro corrente da chamada, sem opção de edição como ocorre no registrar saída
- Quando o registro vier oriundo do botão "Registrar Entrada"

```
select t.id, t.nome
from tipo_periodo_ponto t
where t.id in
    (select 1 from config_ponto c
    where c.hora_inicial_p1 is not null
    and current_date between c.data_inicial_vigencia and c.data_final_vigencia
    and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
union
select 2 from config_ponto c
where c.hora_inicial_p2 is not null
    and current_date between c.data_inicial_vigencia and c.data_final_vigencia
    and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
union
select 3 from config_ponto c
where c.hora_inicial_p3 is not null
    and current_date between c.data_inicial_vigencia and c.data_final_vigencia
    and c.id_funcionario = 2 ) --idFuncionarioUsuarioLogado
and not exists
(select p.id_tipo_periodo_ponto
from ponto p
left join config_ponto c on (p.id_config_ponto = c.id)
where p.data = current_date
    and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
    and p.id_tipo_periodo_ponto = t.id)
```

**RN011**

- Somente visível se a consulta baixo retornar um id :

```
select id from config_ponto c
where current_date between c.data_inicial_vigencia and c.data_final_vigencia
and c.tarefa_online = true
and c.id_funcionario = 2 --idFuncionarioUsuarioLogado
```

**RN012**

- Criar função passando id\_tipo\_ponto periodo com retorno hora\_inicial\_prevista e hora\_final\_prevista
- Para id\_tipo\_perido\_ponto = 1

```
select c.hora_inicial_p1 as hora_inicial_prevista, c.hora_final_p1 as hora_final_prevista
from config_ponto c
where c.id = (select id from config_ponto cp
where current_date between cp.data_inicial_vigencia and cp.data_final_vigencia
and cp.id_funcionario = 2 ) --idFuncionarioUsuarioLogado
-- Para id_tipo_perido_ponto = 2
select c.hora_inicial_p2 as hora_inicial_prevista, c.hora_final_p2 as hora_final_prevista
from config_ponto c
where c.id = (select id from config_ponto cp
where current_date between cp.data_inicial_vigencia and cp.data_final_vigencia
and cp.id_funcionario = 2 ) --idFuncionarioUsuarioLogado
-- Para id_tipo_perido_ponto = 3
select c.hora_inicial_p3 as hora_inicial_prevista, c.hora_final_p3 as hora_final_prevista
from config_ponto c
where c.id = (select id from config_ponto cp
where current_date between cp.data_inicial_vigencia and cp.data_final_vigencia
and cp.id_funcionario = 2 ) --idFuncionarioUsuarioLogado
```

**RN013**

- select distinct extract('year' from data) from ponto;