

Kronos

1

https://kronos/gerenciarHomologacaoPonto.html

Gerenciar Homologação Ponto / Tarefa

Todos/Ativos/Inativos

Categoria

Função

Nr Registro

CPF

Nome

Limpar

Pesquisar

Dados dos Funcionários

RN001

Ações

2

[CPF] [Nr Registro] [Nome] [Data Nascimento] [Email] [Data Admissão] [Data Demissão][Função Registro]

[Folha de Ponto]

[CPF] [Nr Registro] [Nome] [Data Nascimento] [Email] [Data Admissão] [Data Demissão][Função Registro]

[Folha de Ponto]

Folha de Ponto

2

Nr Registro

CPF

Data Nasc

Nome

Data Admissão

Data Demissão

RN005

RN006

RN007

RN008

RN009

RN010

Todos

Pendentes

Ausentes

4

Dados de Pontos / Tarefas

RN002

Ações

2

[Data] [Dia Semana] [Período] [Hora Entrada] [Hora Saída] [Data Liberação] [Data Homologação] [Descrição tarefa]

[Visualizar]

[Data] [Dia Semana] [Período] [Hora Entrada] [Hora Saída] [Data Liberação] [Data Homologação] [Descrição tarefa]

[Visualizar]

Visualizar Ponto / Tarefa

3

Data

Período

Entrada Prevista

Auditoria Entrada

Hora Entrada

Saída Prevista

Auditoria Saída

Hora Saída

Horas Registrada

Data Liberação

Data Homologação

Funcionário Liberação

Funcionário Homologação

Descrição da tarefa on-line

Fechar

2

Geração Pontos Ausentes

4

Suprimir:

Sábado

Domingo

Listar

Dados de Pontos Ausentes

RN003

[Data] [Dia Semana] [Período]

[Data] [Dia Semana] [Período]

Cancelar

2

2

Gerar

RN004

RN001
select distinct f.id as id_funcionario, p.id as id_pessoa, p.CPF, p.Numero_Registro, p.Nome, p.Data_Nascimento, p.Email, f.Data_Admissao, f.Data_Demissao
from funcionario f
left join pessoa p on (f.id_pessoa = p.id)
left join funcionario_funcao ff on (f.id = ff.id_funcionario)
left join tipo_funcao tf on (ff.id_tipo_funcao = tf.id)
where f.id_instituicao = 2 --IdInstituicaoUsuarioLogado
-- and f.data_demissao is null --filtroAtivos
-- and (select count(f1.id) from funcionario f1 where f1.data_demissao is null and f1.id_pessoa = p.id) = 0 --filtroInativos
-- and tf.id_tipo_categoria_funcao = 3 --idTipoCategoriaFuncaoSelecionada
-- and ff.id_tipo_funcao = 4 --idTipoFuncaoSelecionada
-- and p.CPF = '06849869817' --filtro CPF
-- and p.Numero_Registro= 2020010001 --filtroNrRegistro
and Upper(p.Nome) like '%FRAN%' --filtroNome
order by p.nome, f.Data_Admissao

RN002
select p.data,
CASE EXTRACT(DOW FROM p.data)
WHEN 0 THEN 'Domingo'
WHEN 1 THEN 'Segunda'
WHEN 2 THEN 'Terça'
WHEN 3 THEN 'Quarta'
WHEN 4 THEN 'Quinta'
WHEN 5 THEN 'Sexta'
WHEN 6 THEN 'Sábado'
END AS DiaSemana,
t.nome as periodo,
CASE
WHEN p.id_tipo_periodo_ponto = 1 THEN c.hora_inicial_p1
WHEN p.id_tipo_periodo_ponto = 2 THEN c.hora_inicial_p2
WHEN p.id_tipo_periodo_ponto = 3 THEN c.hora_inicial_p3
ELSE Null
END as hora_inicial_prevista,
CASE
WHEN p.id_tipo_periodo_ponto = 1 THEN c.hora_final_p1
WHEN p.id_tipo_periodo_ponto = 2 THEN c.hora_final_p2
WHEN p.id_tipo_periodo_ponto = 3 THEN c.hora_final_p3
ELSE Null
END as hora_final_prevista,
p.hora_inicial_registro, p.hora_final_registro,
p.hora_inicial_informado, p.hora_final_informado, p.data_liberacao,
p.data_homologacao, p.descricao as tarefa,
pl.nome as func_liberacao, pl.nome as func_homologacao
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)
left join funcionario fl on (p.id_funcionario_liberacao = fl.id)
left join pessoa pl on (fl.id_pessoa = pl.id)
left join funcionario fh on (p.id_funcionario_homologacao = fh.id)
left join pessoa ph on (fh.id_pessoa = ph.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

RN004
- Para cada registro selecionado o sistema irá gerar o registro de um ponto com os seguintes valores:
tarefa_online = false
hora_inicial_registro = NULL
hora_final_registro = NULL
hora_final_informado = NULL
hora_inicial_informado = NULL
data = data registro selecionado
id_tipo_periodo_ponto = tipo registro selecionado
id_config_ponto = id_config_ponto vigente do funcionario selecionado
id_funcionario_liberacao = id_funcionario logado
data_liberacao = data do servidor
data_homologacao = NULL
id_funcionario_homologacao = NULL
descricao = NULL

RN007
- Registra data corrente no campo data_homologacao e idFuncionario do usuário logado no campo id_funcionario_homologacao para os registros selecionados

RN009
select distinct extract(YEAR FROM p.data)
from ponto p
left join config_ponto c on (p.id_config_ponto = c.id)
where c.id_funcionario = 2 --idFuncionarioSelecionado
order by 1 desc

RN003
select
1 as id_tipo_periodo_ponto,
d.dg as data_geracao,
cp.hora_inicial_p1 as hora_inicial_prevista
from config_ponto cp
left join (select * from (
select generate_series(
(date '2020-03-01')::date, --primeiroDiaMesSelecionado
(date '2020-03-31')::date, --ultimoDiaMesSelecionado
interval '1 day'
) dg
) as dias) d on (1=1)
where cp.hora_inicial_p1 is not null
and cp.id_funcionario = 2 --idFuncionarioSelecionado
and not exists (select p.id_tipo_periodo_ponto, p.data from ponto p
left join config_ponto x on (p.id_config_ponto = x.id)
where cp.id_funcionario = 2 --idFuncionarioSelecionado
and date_part('month', p.data) = 3 --mesSelecionado
and date_part('year', p.data) = 2020 --anoSelecionado
and p.id_tipo_periodo_ponto = 1 and p.data = d.dg)
and EXTRACT(DOW FROM d.dg) not in (0,6) --diasSemanaSuprimidos 0=Dom...6=Sab
union
select
2 as id_tipo_periodo_ponto,
d.dg as data_geracao,
cp.hora_inicial_p2 as hora_inicial_prevista
from config_ponto cp
left join (select * from (
select generate_series(
(date '2020-03-01')::date, --primeiroDiaMesSelecionado
(date '2020-03-31')::date, --ultimoDiaMesSelecionado
interval '1 day'
) dg
) as dias) d on (1=1)
where cp.hora_inicial_p2 is not null
and cp.id_funcionario = 2 --idFuncionarioSelecionado
and not exists (select p.id_tipo_periodo_ponto, p.data from ponto p
left join config_ponto x on (p.id_config_ponto = x.id)
where cp.id_funcionario = 2 --idFuncionarioSelecionado
and date_part('month', p.data) = 3 --mesSelecionado
and date_part('year', p.data) = 2020 --anoSelecionado
and p.id_tipo_periodo_ponto = 2 and p.data = d.dg)
and EXTRACT(DOW FROM d.dg) not in (0,6) --diasSemanaSuprimidos 0=Dom...6=Sab
union
select
3 as id_tipo_periodo_ponto,
d.dg as data_geracao,
cp.hora_inicial_p3 as hora_inicial_prevista
from config_ponto cp
left join (select * from (
select generate_series(
(date '2020-03-01')::date, --primeiroDiaMesSelecionado
(date '2020-03-31')::date, --ultimoDiaMesSelecionado
interval '1 day'
) dg
) as dias) d on (1=1)
where cp.hora_inicial_p3 is not null
and cp.id_funcionario = 2 --idFuncionarioSelecionado
and not exists (select p.id_tipo_periodo_ponto, p.data from ponto p
left join config_ponto x on (p.id_config_ponto = x.id)
where cp.id_funcionario = 2 --idFuncionarioSelecionado
and date_part('month', p.data) = 3 --mesSelecionado
and date_part('year', p.data) = 2020 --anoSelecionado
and p.id_tipo_periodo_ponto = 3 and p.data = d.dg)
and EXTRACT(DOW FROM d.dg) not in (0,6) --diasSemanaSuprimidos 0=Dom...6=Sab
order by 2,1

RN005
- Registra data corrente no campo data_liberacao e idFuncionario do usuário logado no campo id_funcionario_liberacao para os registros selecionados

RN006
- Registra Null nos campos data_liberacao e id_funcionario_liberacao para os registros selecionados

RN008
- Registra Null nos campos data_homologacao e id_funcionario_omologacao para os registros selecionados

RN010
select distinct extract(MONTH FROM p.data) as nrMes,
CASE EXTRACT(MONTH FROM p.data)
WHEN 1 THEN 'Janeiro'
WHEN 2 THEN 'Fevereiro'
WHEN 3 THEN 'Março'
WHEN 4 THEN 'Abril'
WHEN 5 THEN 'Maio'
WHEN 6 THEN 'Junho'
WHEN 7 THEN 'Julho'
WHEN 8 THEN 'Agosto'
WHEN 9 THEN 'Setembro'
WHEN 10 THEN 'Outubro'
WHEN 11 THEN 'Novembro'
WHEN 12 THEN 'Dezembro'
END AS nomeMes
from ponto p
left join config_ponto c on (p.id_config_ponto = c.id)
where extract(YEAR FROM p.data) = 2020 --anoSelecionado
and c.id_funcionario = 2 --idFuncionarioSelecionado
order by 1