<% if params[:filter].blank? %>

<%= form\_tag params[:id] do %>

<ul>

<li>Sistema(s)<br />

<%= select\_tag 'filter[system]', options\_for\_select(Sistema.order(:nome).map { |s| [s.nome, s.id] }),

multiple: true, size: 5, style: "width: 400px" %></li>

<li>Organização <%= select\_tag 'filter[org]', options\_for\_select(Usuario.select(:organizacao)

.order(:organizacao).uniq.map(&:organizacao)),

include\_blank: true %></li>

<li>

Criado em:

<%= text\_field\_tag 'filter[date]', '', class: 'datepicker' %>

até

<%= text\_field\_tag 'filter[enddate]', '', class: 'datepicker' %>

</li>

</ul>

<%= submit\_tag 'Buscar' %>

<% end %>

<% else %>

<%

requisitions = UserRequisition.eager\_load(reqitems: :sistema).preload(:referencia)

requisitions = requisitions.where('sistemas.id' => params[:filter][:system])

if params[:filter][:org].present?

requisitions.select! do |req|

req.referencia.organizacao == params[:filter][:org]

end

end

from\_date = Date.parse(params[:filter][:date]) rescue nil

if from\_date

end\_date = Date.parse(params[:filter][:enddate]) rescue (from\_date + 1.day)

requisitions = requisitions.where("str\_to\_date(requisicaos.created\_at, '%Y-%m-%d') >= '#{from\_date}' and

str\_to\_date(requisicaos.created\_at, '%Y-%m-%d') <= '#{end\_date}'")

requisitions = requisitions.where(created\_at: from\_date..(end\_date+1.day))

end

@csv = [[

"Número",

"Descrição",

"Sistema",

"Login",

"Nome",

"Organização",

"Departamento",

"Status",

"Perfil Solicitado",

"Tipo Aprovação",

"Aprovação Completa",

"Aprovador/Executor Atual",

"Perfis Aprovados",

"Tipo",

"Criada em",

"Atualizada em",

"Criada por",

"Finalizada por",

"Sisusuario"

]]

%>

<table class="table table-striped table-condensed table-hover table-responsible">

<tr>

<th><%=raw @csv[0].join("</th><th>") %></th>

</tr>

<% requisitions.each do |req| %>

<%

current\_approvals = if req.status == Requisicao::STATUS\_EM\_APROVACAO

req.reqaprovacaos.joins(:approval\_rule).where('approval\_rules.nivel\_requisicao' => req.nivel\_atual)

else

[]

end

final\_executor = ""

if req.status == Requisicao::STATUS\_FECHADA

closed\_history = req.reqhists.where(descricao: "Requisição fechada").last

if closed\_history

login = closed\_history.user.try(:nome)

if login.nil? || login == User.admin.nome

final\_executor = "Autoseg"

else

final\_executor = login

end

end

end

roles = []

roles\_approval = []

req.reqitems.each do |item|

if item.referencia\_type == "Perfil"

roles << item.referencia

roles\_approval << item.referencia.to\_s if item.status == Reqitem::STATUS\_APROVADA

end

end

roles.uniq.each do |role|

@csv << [

req.id,

req.descricao,

req.reqitems.map { |i| i.sistema.nome }.uniq.join(", "),

req.usuario.login,

req.usuario.nome,

req.usuario.organizacao,

req.usuario.departamentos.first.try(:nome),

req.status,

role.to\_s,

current\_approvals.map { |r| r.descricao }.uniq.join(", "),

req.aprovada,

current\_approvals.map { |r| r.user.nome }.uniq.join(", "),

roles\_approval.uniq.join(", "),

req.class.model\_name.human,

I18n.l(req.created\_at),

I18n.l(req.updated\_at),

req.user.nome,

final\_executor,

req.usuario.sisusuarios.where(sistema\_id: role.sistema\_id).first

]

end

%>

<% end %>

<% @csv[1...@csv.size].each do |row| %>

<tr>

<td><%=raw row.join("</td><td>") %></td>

</tr>

<% end %>

</table>

<p>

<%= link\_to 'Baixar CSV', relatorio\_manual\_path(params[:id], action: 'baixar\_csv', filter: params[:filter], format: 'csv') %>

</p>

<% end %>