

HTML

Utilização do HTML nos processos BPM

- [Dicas para montar um E-mail em HTML compatível com vários provedores](#)
- [Imprimir relatório através de link HTML \(com filtro preenchido\)](#)
- [Modelo de blocos HTML de agrupamento](#)
- [Template exemplo para notificação de e-mail](#)
- [Utilizando Bloco HTML](#)
- [Visualizador de arquivo no bloco HTML](#)

Dicas para montar um E-mail em HTML compatível com vários provedores

Olá, aqui temos dicas para montar um E-mail em HTML compatível com vários provedores (Gmail, Thunderbird, Outlook, entre outros).

Algumas vezes enfrentarmos problemas com a compatibilidade dos e-mails criados por HTML que enviamos através do Dox. Principalmente quando utilizamos o Outlook para Desktop, pois o mesmo ainda utiliza a engine do IE6 mesmo nas versões mais atuais.

Para ajudá-los reuni uma série de links que com boas práticas que eu utilizo para construir um modelo de e-mail com maior compatibilidade entre vários clientes de e-mail (Thunderbird (Desktop), outlook.com (web), mail.yahoo.com (web), outlook (Desktop)).

Começo lendo sobre as boas práticas:

- <https://webdesign.tutsplus.com/pt/articles/build-an-html-email-template-from-scratch--webdesign-12770>
- <https://tableless.com.br/boas-praticas-para-e-mail-marketing/>

Durante a criação do seu e-mail verifique a compatibilidade dos clientes:

- <https://pt.stackoverflow.com/questions/169437/email-html-5-ainda-deve-ser-feito-dentro-de-tabelas>
- <https://www.outlook-apps.com/html-ignored-by-outlook/>
- <https://www.campaignmonitor.com/css/>

Sites que realizar a validação do seu código HTML:

- https://validator.w3.org/#validate_by_upload (free)
- <https://www.htmlemailcheck.com/check/> (contém uma versão paga para recursos avançados)
- <https://putsmail.com/tests/new> (versão paga, mas é possível criar uma conta Trial)
- <https://www.google.com/webmasters/markup-tester/>

- <https://testi.at/> (versão free é limitada)

Espero que esses links ajudem vocês a resolver os problemas de compatibilidade (se houver) como me ajudou.

Imprimir relatório através de link HTML (com filtro preenchido)

Neste tópico, você verá como gerar um relatório a partir de um formulário do tipo HTML (com link) em uma atividade de processo preenchendo seus filtros automaticamente com informações/variáveis do processo.

Além de uma solução diferente para imprimir relatório na atividade, possibilita múltiplos protocolos/relatórios para impressão na mesma atividade.

Para baixar o BPM deste exemplo, [clique aqui](#).

- Crie um processo BPM simples.
- Uma atividade manual.
- Dois formulários de bloco HTML com apenas um título.

[image-1652191270536.png](#)

Image not found or type unknown

Na aba "Eventos" iremos incluir um evento PEX ao entrar na atividade. Este evento PEX estará responsável por carregar nos blocos HTML os link's para a impressão do relatório.

[image-1652191368614.png](#)

Image not found or type unknown

```
const
    csLinkProtocolo1 = '1'; // codigo do formulario HTML
    csLinkProtocolo2 = '2'; // codigo do formulario HTML
var
    lsFiltro, lsHash, lsTemplate, lsRelatorio, lsJson, lsLink, lsUrlPortal : String;
    loCDS : TLibCDS; // declaracao das variaveis
begin
    loCDS := of_CriaCDSporSQL(' SELECT valor URL FROM ini where secao = '' CRM '' and chave =
```

```

'' DOXPORTAL_LINK_PORTAL'' );

    try
        lsUrlPortal := loCDS.GetStr(['URL']);
    finally
        loCDS.Free;
    end;

    lsFiltro :=
TS0.Encode64String('{"IDPROCESSO":""+aoVariaveis.GetStr('/*IDPROCESSO*/')+ '}'');
    lsHash    := TS0.Encode64String( aoVariaveis.GetStr('/*HASHSESSAO*/') );

    //Relatório 1
    lsRelatorio := '398'; //alterar valor pelo código do relatório desejado
    lsTemplate := '434'; //alterar valor pelo código do template do relatório
desejado

    lsJson    := TS0.Encode64String('{"IDRELTEMPLATE":"" + lsTemplate + '","IDRELATORIO":"" +
lsRelatorio + '}"');

    lsLink := '<a style="font-family: Roboto, sans-serif; font-size: 0.8rem;" href="' +
lsUrlPortal +
        '/api/downloadrelatoriocadastrado?aoJson=' + lsJson + '&aoFiltrosJson=' +
LsFiltro +
        '&asHash=' + lsHash + '" target="_blank">IMPRIMIR RELATÓRIO</a>';

    aoFormularios.GetJSON(csLinkProtocolo1).SetStr('TEXT0', lsLink);

    //Relatório 2
    lsRelatorio := '399'; //alterar valor pelo código do relatório desejado
    lsTemplate := '435'; //alterar valor pelo código do template do relatório
desejado

    lsJson    :=
TS0.Encode64String('{"IDRELTEMPLATE":""+lsTemplate+"","IDRELATORIO":""+lsRelatorio+"}");

    lsLink := '<a style="font-family: Roboto, sans-serif; font-size: 0.8rem;" href="' +
lsUrlPortal +
        '/api/downloadrelatoriocadastrado?aoJson=' + lsJson + '&aoFiltrosJson=' +
LsFiltro +
        '&asHash=' + lsHash + '" target="_blank">IMPRIMIR RELATÓRIO</a>';

```

```
aoFormularios.GetJSON(csLinkProtocolo2).SetStr('TEXT0', lsLink);  
end;
```

Alterar os valores atribuídos às variáveis "IsRelatorio" e "IsTemplate" de acordo com os relatórios cadastrados que deseja imprimir. Caso tenha dificuldade para entender qual o template ou idrelatório específico acessar:

- Dox estúdio
- Módulo de inteligência
- Relatórios

[image-1652191559753.png](#)

Image not found or type unknown

Ou pesquisar pelo banco de dados:

```
SELECT *  
FROM RELATORIOGERADOR
```

Exemplo do processo no portal:

[image-1652191726594.gif](#)

Image not found or type unknown

Modelo de blocos HTML de agrupamento

Código fonte HTML:

```
<div style="width: 100%; border-width: 0px 0px 3px 0px;border-style: groove ; border-color:
#000000;" >
  <table style="width: 100%;background-color:transparent " >
    <tbody>
      <tr>
        <td style="text-align: center; vertical-align: middle; background-color: rgb(102,
0, 170); padding: 3px;border-radius: 20px 20px 0px 0px;"><big><span style="color: #fdfdfd;
font-weight: bold;">CABEÇALHO</span></big></td>
      </tr>
    </tbody>
  </table>
</div>
```

Exemplo de uso aplicação:

[image-1637948774882.png](#)

Image not found or type unknown

Outra forma pensando em boas praticas e facilidade em manutenção do código CSS seria a seguinte forma:

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;700&family=Roboto:wght@400;
rel="stylesheet">
<style>
  * {
    /*font-family: 'Poppins', sans-serif;*/
    font-family: 'Roboto', sans-serif;
    padding: 0;
```

```

        margin: 0;
        outline: 0;
    }
    .cor-de-fundo {
        background-color: #5E2981; /* Cor do fundo do agrupador */
    }
    .borda-inferior {
        border-bottom: 3px solid #B088C9; /* Borda inferior do agrupador utilizando cor
        HEX, porém poderia ser utilizado o padrão ex: purple, white ... */
    }
    .container {
        color: #fff;
        line-height: 1.5;
        border-top-left-radius: .4rem;
        border-top-right-radius: .4rem;
        width: 100%;
        text-align: center;
    }
    .container p {
        padding: 5px 10px;
        font-weight: bold;
        font-size: 1.2em;
    }
</style>
<div class="container cor-de-fundo borda-inferior ">
    <p>
        Texto do agrupador /*Texto que irá aparecer como "titulo" no agrupador */
    </p>
</div>

```

Exemplo de uso aplicação:

[image-1637948841497.png](#)

Image not found or type unknown

***** nas versões mais recentes do software, já existe a opção de agrupador padrão do sistema, semelhante ao que está mostrado na imagem acima.**

</div>

Resultado DeskTop:

[image-1637949370757.png](#)

Image not found or type unknown

Resultado Mobile:

[image-1637949397227.png](#)

Image not found or type unknown

Utilizando Bloco HTML

Existe possibilidades em nosso sistema DOX que permitem o uso de HTML no campo de títulos de formulários e também nos formulários de **Bloco HTML**.

O Bloco HTML é um recurso muito útil quando usado da forma correto e bem feito, pois nele é possível criar inúmeras coisas, porém vale ressaltar que o bloco lê todo código HTML, CSS, Java Script entretanto esses códigos devem estar em uma única pagina para que ele consiga ler ou referencia-los de uma pagina já na web, como a imagem abaixo mostra:

[image-1637949012641.png](#)

Image not found or type unknown

Um exemplo dentro de um processo BPM foi criar um navbar que iria direcionar para outras paginas, contendo listas e vários componentes.

[image-1637949048324.png](#)

Image not found or type unknown

Código fonte:

```
<!DOCTYPE html>
<html lang="PT-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link
href="https://fonts.googleapis.com/css?family=Gruppo|Roboto+Condensed|Rubik+Mono+One&display=sw
rel="stylesheet">
  <title>Forum Ema</title>
<style>
  *{
    margin: 0;
    padding: 0;
    font-family: 'Roboto Condensed', sans-serif;

  }
  .menu {
```

```
width: 100%;
height: 50px;
background-color: #671176;

}

.centralizar{
display: flex;
flex-direction: row;
flex-wrap: wrap;
justify-content: center;

}

/*Estilizando as listas */
.menu ul {
list-style: none;
position: relative;
}

.menu ul li {
width: 150px;
float: left;
}

.menu a {
padding: 15px;
display: block;
text-decoration: none;
text-align: center;
background-color: #671176;
color: #fff;
font-family: 'Roboto Condensed', sans-serif;
}

.menu ul ul {
position: absolute;
visibility: hidden;
}

.menu ul li: hover ul {
```

```
        visibility: visible;
    }

    .menu a:hover {
        background-color: #E48AF4;
        color: white;
    }

    .menu ul ul li a:hover{
        background-color: #E48AF4;
        color: white;
    }

    .menu ul ul li {
        float: none;
        border-bottom: solid 1px #671176;
    }

    .menu ul ul li a {
        background-color: #580268;
    }

    .corpo img{
        width: 45px;
        min-width: 80px;
        margin: 30px 47%;
    }

    .imagem{
        display: flex;
        align-items: center;
        justify-content: center;
    }

    .corpo{
        width: 100%;
    }

    /*Deixando o Menu Responsivo */
    @media(max-width: 800px) {

        .menu{
            transition: all .4s;
            width: 100%;
        }
    }
}
```

```

        height: auto;
    }

    .menu ul li {
        width: 100%;
        float: none;
    }

    .menu ul ul {
        position: static;
        overflow: hidden;
        max-height: 0;
        transition: all 1s;
    }

    .menu ul li:hover ul {
        height: auto;
        max-height: 200px;
    }

    .corpo{
        width: 100%;
    }

    .corpo img{
        margin: 10px 38%;
    }
}

/*TEXT0 MAIN**/

.corpo{
    text-align: center;
}

.back-texto{
    margin: 5%;
    border-radius: 2px;
    -webkit-box-shadow: 4px 7px 3px rgba(50, 50, 50, 0.77);
    -moz-box-shadow: 4px 7px 3px rgba(50, 50, 50, 0.77);
    box-shadow: 4px 7px 3px rgba(50, 50, 50, 0.77);
    border-top: solid rgba(203, 202, 202, 0.77);
    border-left: solid rgba(203, 202, 202, 0.77);
}

.back-texto li{

```

```

    list-style: none;
}
/*RODAPE*/

.rodape{
    margin: 0;
    padding: 0;
    border: 0;
    background-color: #580268;
    vertical-align: baseline;
}
</style>
</head>

<body>
    <header>
        <nav class="menu">
            <ul class="centralizar">
                <li><a target="_blank">EMA</a>
                    <ul>
                        <li><a>PROCESSO MARMITA</a></li>
                        <li><a>PROCESSO FOLGA</a></li>
                        <li><a>RESERVAR CARRO</a></li>
                    </ul>
                </li>
                <li><a href="http://forum.ema.net.br/index.php" target="_blank">ACESSAR
FORUM</a></li>
                <li><a>CONTEUDOS</a>
                    <ul>
                        <li><a href="http://forum.ema.net.br/viewforum.php?f=10"
target="_blank">HTML</a></li>
                        <li><a href="http://forum.ema.net.br/viewforum.php?f=11"
target="_blank">ORACLE</a></li>
                        <li><a href="http://forum.ema.net.br/viewforum.php?f=12"
target="_blank">FIREBIRD</a></li>
                        <li><a href="http://forum.ema.net.br/viewforum.php?f=13"
target="_blank">POSTGRESQL</a></li>
                    </ul>
                </li>
                <li><a href="https://ema.net.br/contato/"

```



```

target="_blank">CONTATO</a></li>
    </ul>
</nav>
</header>

<main class="corpo">
    
    <div class="back-texto">
        <h3>Ema Software</h3>
        <p>Em algumas situações, é necessário personalizar de modo mais avançado o layout
de uma atividade de processo. Neste exemplo, você pode conferir como utilizar o tipo de
formulário Bloco HTML com um exemplo prático de utilização de vários recursos HTML, CSS e
Javascript.</p>
        <ul>
            <h4>Possibilidades do bloco:</h4>
            <li>HTML</li>
            <li>CSS</li>
            <li>JavaScript</li>
            <li>Integração Api</li>
        </ul>
    </div>
    <div id="script"></div>
</main>

<footer class="rodape"></footer>

<script>
    var btn = document.createElement('button');
    var containerElement = document.querySelector('#script');

    btn.setAttribute('type','button');
    btn.appendChild( document.createTextNode('Clique Aqui') );

    btn.onclick = function(){
        alert(' JavaScript Puro' );
    }

    containerElement.appendChild(btn);
</script>
</body>

```

```
</html>
```

Visualizador de arquivo no bloco HTML

Com o tipo de formulário **Bloco HTML**, é possível criar um formulário mais usual e visualmente elaborado. Aqui você verá que é possível incluir o visualizar de arquivos do navegador (para ver imagens e documentos sem precisar efetuar download) dentro do formulário de uma atividade do **DOX**.

[Clique aqui](#) para baixar o processo deste exemplo.

[image-1651005872601.gif](#)

Image not found or type unknown

Neste exemplo, tem duas atividades no processo. Na primeira atividade, há um campo, onde o usuário anexa o arquivo (documento ou imagem), na segunda atividade o conteúdo do arquivo anexado é mostrado na tela através do recurso de visualização de arquivos do navegador.

Na primeira atividade é configurado um formulário do tipo **dissertativo** vinculado à uma variável do tipo **Anexo - Arquivo** para salvar o conteúdo:

[image-1648821333983.png](#)

Image not found or type unknown

E uma variável do tipo **inteiro** para gravar o código do anexo:

[image-1648821483547.png](#)

Image not found or type unknown

Na segunda atividade, é configurado um campo do tipo **Bloco HTML** para mostrar o conteúdo do arquivo anexado, com o recurso **<iFrame>**.

[image-1648821688739.png](#)

Image not found or type unknown

É necessário inserir o endereço do anexo no atributo **src** da tag **<iFrame>**.

```
<iframe width="100%" height="700px"
src="http: //127. 0. 0. 1/GED/api/visualizaarquivo?asHash=/*HASHSESSAO*/&IDARQUIVO=/*CODARQUIVO*/&DI
```

Verificar esses valores:

- "127.0.0.1": Endereço de acesso ao portal, terá que alterar pelo endereço configurado para o seu portal;
- "/*CODARQUIVO*/": variável com o código do arquivo vinculada no formulário onde é anexado o mesmo.

Disponível na versão: **11+**

Versão homologada: **12.29 ++**

Navegadores homologados: **Chrome** e **Firefox**