

Tutorial

1. Sobre o tutorial

O objetivo deste tutorial é incluir máscaras e validações automáticas em campos dos sistemas criados APEX utilizando javascript.

2. Funcionamento

A ideia é criar um arquivo *util.js* e incluí-lo em todas as páginas da aplicação. Utilizaremos a *jquery-maskedinput.js* (<https://github.com/digitalBush/jquery.maskedinput>) para estabelecer, no *util.js*, máscaras para determinadas classes CSS, como, por exemplo, *mascara-cpf*. Após isso, sempre que houver a necessidade de utilizá-las colocando-as na definição do item, no campo "Classes CSS", como se pode observar na imagem abaixo:

Aparência

Interface do Usuário	Desktop
Modo de Página	Normal
Modelo de Página	STN - One Level Tabs - Left Sidebar
Opções de Modelo	Usar Padrões de Modelo
Classes CSS	mascara-cpf
Tipo de Mídia	

3. Procedimento

Faça o download da *jquery-maskedinput.js* (<https://github.com/digitalBush/jquery.maskedinput>) e crie um arquivo *util.js*. Faça o upload destes arquivos nos Componentes Compartilhados (no diretório js). Inclua, no tema, os dois js, no campo "URLs de Arquivo JavaScript" das configurações do Tema:

```
#WORKSPACE_IMAGES#js/jquery.maskedinput.js  
#WORKSPACE_IMAGES#js/util.js
```

4. Máscaras

Inclua no arquivo *util.js* as seguintes funções para máscaras:

```
$( document ).ready(function() {

    // Definições de Máscaras
    $(".mascara-data").mask("99/99/9999");
    $(".mascara-cpf").mask("999.999.999-99");
    $(".mascara-cnpj").mask("99.999.999/9999-99");
    $(".mascara-telefone").mask("(99)9999-9999?9");
    $(".mascara-cep").mask("99.999-999");

    $('.mascara-telefone').focusout(function(){
        var phone, element;
        element = $(this);
        element.unmask();
        phone = element.val().replace(/\D/g, '');
        if(phone.length > 10) {
            element.mask("(99)99999-999?9");
        } else {
            element.mask("(99)9999-9999?9");
        }
    });
});
```

Suba o arquivo util.js e, a partir de então, será possível utilizar as seguintes máscaras:

- **mascara-data:** para campos de data;
- **mascara-cpf:** para campos de CPF;
- **mascara-cnpj:** para campos de CNPJ;
- **mascara-telefone:** para campos de telefone; e
- **mascara-cep:** para campos de CEP.

Inclua, agora, as funções de verificar CPF, verificar CNPJ e e-mail:

```
/**
 * Função para validar CPF.
 *
 * @param cpf o valor do CPF a ser validado
 *
 * @author http://www.geradorcpf.com/javascript-validar-cpf.htm
 *
 * @return true, caso seja um CPF válido; false, caso seja um CPF inválido
 */

function verificarCPF(cpf) {
    // Remove os pontos/traço da expressão regular, caso exista
    cpf = cpf.replace(/[^\d]+/g, '');
    if(cpf == '') return false;

    // Elimina CPFs invalidos conhecidos
    if (cpf.length != 11 ||
        cpf == "00000000000" ||
        cpf == "11111111111" ||
        cpf == "22222222222" ||
        cpf == "33333333333" ||
        cpf == "44444444444" ||
        cpf == "55555555555" ||
        cpf == "66666666666" ||
        cpf == "77777777777" ||
        cpf == "88888888888" ||
        cpf == "99999999999")
        return false;

    // Valida 1o digito
    add = 0;

    for (i=0; i < 9; i++) {
        add += parseInt(cpf.charAt(i)) * (10 - i);
    }
    rev = 11 - (add % 11);
    if (rev == 10 || rev == 11) {
        rev = 0;
    }

    if (rev != parseInt(cpf.charAt(9))) {
        return false;
    }

    // Valida 2o digito
    add = 0;
    for (i = 0; i < 10; i++) {
        add += parseInt(cpf.charAt(i)) * (11 - i);
    }
    rev = 11 - (add % 11);
    if (rev == 10 || rev == 11) {
        rev = 0;
    }
    if (rev != parseInt(cpf.charAt(10))) {
```

```
        return false;
    }
    return true;
}

/**
 * Função para verificar CNPJ
 *
 * @param cpf o valor do CNPJ a ser validado
 *
 * @author http://www.geradorcnpj.com/javascript-validar-cnpj.htm
 *
 * @return true, caso seja um CNPJ válido; false, caso seja um CNPJ inválido
 */

function verificarCNPJ(cnpj) {
    cnpj = cnpj.replace(/[\^\\d]+/g, '');

    if(cnpj == '') return false;
    if (cnpj.length != 14)
        return false;
    // Elimina CNPJs invalidos conhecidos
    if (cnpj == "00000000000000" ||
        cnpj == "11111111111111" ||
        cnpj == "22222222222222" ||
        cnpj == "33333333333333" ||
        cnpj == "44444444444444" ||
        cnpj == "55555555555555" ||
        cnpj == "66666666666666" ||
        cnpj == "77777777777777" ||
        cnpj == "88888888888888" ||
        cnpj == "99999999999999")
        return false;

    // Valida DVs
    tamanho = cnpj.length - 2
    numeros = cnpj.substring(0,tamanho);
    digitos = cnpj.substring(tamanho);
    soma = 0;
    pos = tamanho - 7;
    for (i = tamanho; i >= 1; i--) {
        soma += numeros.charAt(tamanho - i) * pos--;
        if (pos < 2)
            pos = 9;
    }

    resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;
    if (resultado != digitos.charAt(0))
        return false;
    tamanho = tamanho + 1;
    numeros = cnpj.substring(0,tamanho);
    soma = 0;
    pos = tamanho - 7;
    for (i = tamanho; i >= 1; i--) {
```

```
soma += numeros.charAt(tamanho - i) * pos--;  
if (pos < 2)  
    pos = 9;  
}  
  
resultado = soma % 11 < 2 ? 0 : 11 - soma % 11;  
if (resultado != digitos.charAt(1))  
    return false;  
  
return true;  
  
}  
  
/**  
 * Valida E-mail.  
 *  
 * @author sectrean  
 * @author Yi Yang  
 * @author ripper234  
 * @author Peter Mortensen  
 * @author multimedixp  
 */  
  
function validaEmail(email) {  
    var re = /^[^<>()[\]\.\,\;\s@\""]+(\.[^<>()[\]\.\,\;\s@\""]+)*|(\\".+\"))@((\  
    return re.test(email);  
}
```

Após isso, inclua no util.js funções que verificam se um campo com classes CSS mascara-cpf, mascara-cnpj ou mascara-email foram modificados. Caso eles tenham sido modificados, executar a função de validação e, caso seja inválido, incluir uma classe "item-error" no campo e adicionar uma mensagem de erro.

```

/**
 * Verifica uma validação com a função funcaoValidacao e, caso
 * tenha um erro, inclui a classe CSS "item-error" no input
 * e insere uma mensagem
 *
 * @author Igor Augusto
 *
 * @param funcaoValidacao a função de validação a ser usada
 * @param mensagemErro a mensagem de erro a ser usada
 */

verifica_validacao = function(campo, funcaoValidacao, mensagemErro) {
    if(campo.next().hasClass("label-error")) {
        campo.next().remove();
    }
    // Verifica se é um CPF
    if(campo.val() != '' && funcaoValidacao(campo.val()) == false) {
        // Se for um CPF, incluir a label do erro e marca o item com a classe de
        campo.after("<div class='label-error'>" + mensagemErro + "</div>");
        campo.addClass("item-error");
    } else {
        // Se não for um CPF, remove a classe item-error
        campo.removeClass("item-error");
    }
}

/**
 * Aplica as validações ao vivo.
 */

$(".mascara-cpf:not(.no-validation)").change(function() {verifica_validacao($(this), funcaoValidacaoCPF, mensagemCPF);});
$(".mascara-cnpj:not(.no-validation)").change(function() {verifica_validacao($(this), funcaoValidacaoCNPJ, mensagemCNPJ);});
$(".mascara-email").change(function() {verifica_validacao($(this), validaEmail, mensagemEmail);});

```

Dessa maneira, campos com mascara-cpf, mascara-cnpj e mascara-email serão validados automaticamente caso estejam inválidos no momento em que o usuário terminar de preenchê-lo.

Caso queira incluir uma nova validação ao vivo, basta criar uma nova função de validação e executar o seguinte:

```

$(".<nome-classe>").change(verifica_validacao(<nome-funcao-validacao>, "<mensagem-erro>"));

```

Por fim, crie um arquivo util.css para incluir estilos para a classe item-error (que será uma borda vermelha para o item) e para label-error (que será de cor vermelha). O arquivo .css deverá conter o seguinte:

```
input.item-error {  
    border-color: #D61E1E;  
}  
  
.label-error {  
    color: D61E1E;  
}
```

Inclua nos Componentes Compartilhados e adicione no template o CSS, na definição do tema, no campo "URLs do Arquivo CSS":

```
#WORKSPACE_IMAGES#css/util.css
```

Fontes:

- Validador CPF: <http://www.geradorcpf.com/javascript-validar-cpf.htm>
(<http://www.geradorcpf.com/javascript-validar-cpf.htm>)
- Validador CNPJ: <http://www.geradorcnpj.com/javascript-validar-cnpj.htm>
(<http://www.geradorcnpj.com/javascript-validar-cnpj.htm>)