

Entendendo o funcionamento dos tipos data e hora

Trabalhar com data e hora em sistemas modernos requer entender completamente o uso de fuso horários. Uma aplicação *web* ou *mobile* é composta, em sua maioria das vezes, de banco de dados, servidor e cliente (navegador *web* ou aplicativo). É necessário entender bem que o fuso de cada uma dessas camadas pode ter especificações diferentes que podem gerar confusões ao implementar seu sistema.

Banco de dados

A maioria dos bancos de dados fornecem alguns tipos de dados para armazenamento de data (**DATE**), data e hora (**DATETIME**) ou apenas hora (**TIME**). Tais tipos podem considerar as informações de fuso ou não. A grande confusão ocorre quando tais tipos não levam em consideração o fuso, o que acabam ficando dependente de uma configuração correta do servidor e cliente para que não haja diferença de horários.

Os tipos mais comuns são:

- **DATE**: armazena apenas a data sem informações do fuso. Contém apenas uma data civil, sem qualquer consideração de fuso horário. Por exemplo, data de nascimento, ou a data de vencimento de uma conta, ou um prazo legal, é **DATE**.
- **DATETIME**: armazena data e hora sem informações de fuso. Contém data e hora civis, novamente sem considerar o fuso horário. Se, por exemplo, o prazo de pagamento é "dia tal até 13h00", é responsabilidade do pagador saber se estamos no horário de verão ou não, qual o fuso horário do local em que ele se encontra, etc.
- **TIME**: armazena apenas hora sem informações de fuso. É o mesmo que os campos acima, porém não especifica a data.
- **TIMESTAMP**: armazena informações de data e hora levando em consideração o momento em algo aconteceu e, por isso, consegue diferenciar os fuso horários. É um número que determina um momento específico. Tipicamente é expresso como o "número de segundos desde 1/1/1970 00:00 em Londres" (conhecido como [Era Unix](#)), mas poderia ser qualquer outra base. A ideia do timestamp é que ele vale no mundo todo, ou seja, ele identifica o momento exato em que algo aconteceu. Um acontecimento com timestamp "0" aconteceu em 31/12/1969 às 21:00 no Brasil e 1/1/1970 00:00 em Londres.

O timestamp é útil para registrar *log*, e para determinar se A aconteceu antes ou depois de B, mesmo que A e B tenham acontecido em lados opostos do planeta. Por outro lado, o timestamp é inadequado para registrar datas e horas "civis" porque a hora e até a data muda conforme o fuso horário em que o timestamp é interpretado.

Nem sempre a melhor opção é óbvia. Por exemplo, registrar a data de nascimento com **DATETIME** ou **TIMESTAMP**? De um ponto de vista matemático, o **TIMESTAMP** seria ideal porque um bebê nasce num momento bem determinado. Por outro lado, a data e hora de nascimento têm efeitos civis - colocar a data de nascimento diferente da carteira de identidade pode causar um monte de aborrecimentos - então é melhor usar **DATETIME** e o local de nascimento, já que o fuso horário é de conhecimento público.

Resumindo, ao usar **DATE**, **DATETIME** ou **TIME**, seu servidor tem que conhecer em qual fuso horário você deseja obter e gravar esses valores, para esses casos, **recomendamos que o banco e o servidor estejam no mesmo fuso horário**.

Servidor

O servidor normalmente tem apenas uma configuração global de fuso horário. Ao gravar ou obter campos do tipo **DATE**, **DATETIME** ou **TIME**, ele vai usar a informação de fuso configurada para suas conversões internas. Ao gravar ou obter campos do tipo **TIMESTAMP**, ele vai simplesmente respeitar o que está informado no valor, ou seja, as informações do fuso horário do servidor não interferem. O Cronapp usa o **JAVA** como plataforma base do servidor e o tipo `java.util.Date` para representação de todos os tipos internos: data, data e hora, hora e timestamp. O uso de um único tipo é um padrão especificado pela *Java Persistence Interface* (JPA). Para "entender" o que significa cada tipo, o JPA exige anotações acima da declaração do tipo do atributo na classe que representa a tabela do banco, mas isso não será entrado em detalhes, visto que é um assunto mais *high code* que fica abstraído para quem usa a IDE apenas como *low code*.

Cliente (navegador ou aplicativo)

O fuso horário do aplicativo é o que pode sofrer mais variações. Na maioria das vezes em sistemas modernos, os usuários estão ao redor do mundo e, sendo assim, é preciso aceitar entradas e exibições de datas e horas considerando o fuso horário de quem está digitando ou lendo.

Nesta página

Conteúdo complementar

- [Formatação de máscaras na camada cliente](#)
- [Formatação de máscaras na camada servidor](#)

Exemplo: Imagine um sistema de cadastro de eventos. Nesse sistema, usuários do mundo inteiro poderão cadastrar eventos que devem acontecer segundo o contexto de fuso de quem cadastra, porém, para outros usuários ao redor do mundo, tais horários podem ser exibidos com diferenças. Um evento às 10:00 de Londres ocorre às 07:00 do Brasil ou um evento 1:00 de Londres do dia 10/11/2019 ocorre às 22:00 do dia 09/11/2019 no Brasil.

Como a IDE trata Data e Hora

O Cronapp IDE procurou deixar tudo mais simples para seus usuários, inclusive aqueles que não têm pretensão de que seus sistemas funcionem em diferentes fusos horários. Para isso, vale a pena entender bem os recursos e configurações que o Cronapp oferece, para que não haja desperdício de tempo com tentativas, erros ou suporte.

Configuração do fuso horário

Clique no menu do sistema **Projeto > Configurações** para abrir a janela de configurações, clique na aba **Configurações de Projeto**, em seguida, expanda a subaba **Banco de Dados** para encontrar dois campos importantes (Figura 1):

- **Fuso horário:** configura o fuso horário do servidor (da aplicação). Os campos **DATE**, **DATETIME** e **TIME** terão o fuso da opção selecionada. O campo **TIMESTAMP** não é afetado por essa opção (a não ser no formato textual) e seu valor padrão é (GMT00:00) UTC.
- **Converter Fuso Horário do Cliente:** marcando essa opção o Cronapp se comportará de forma bem simples. Independente do fuso que o cliente (navegador web ou aplicativo) estiver, ele será desconsiderado e será utilizado o fuso informado na opção **Fuso Horário**.

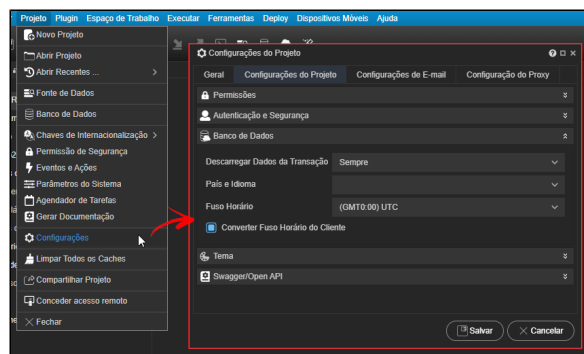


Figura 1 - Campos de configuração de fuso horário na IDE

Tipos do Diagrama de dados

O [Diagrama de dados](#) oferece os 4 tipos mais comuns de data de bancos de dados:

1. **Data:** criará ou considerará o campo do banco como **DATE** e o fuso será considerado como informado nas configurações do projeto;
2. **Data e Hora:** criará ou considerará o campo do banco como **DATETIME** e o fuso será considerado como informado nas configurações do projeto;
3. **Hora:** criará ou considerará o campo do banco como **TIME** e o fuso será considerado como informado nas configurações do projeto;
4. **Carimbo de Data e Hora:** criará ou considerará o campo do banco como **TIMESTAMP**.

Editor HTML

Em vários campos é possível exibir ou editar uma data, data e hora ou hora e possui algumas opções em sua propriedade **Tipo** que devem ser consideradas. Eles terão comportamentos diferentes a depender da configuração feita nas **Configurações do Projeto** (Figura 1). Vários campos podem receber datas.

Para entender mais sobre máscaras no cliente, leia o documento [Formatação de máscaras na camada cliente](#).

Em componentes visuais como **Entrada de texto**, **Título**, **Lista** e outros, é possível usar as propriedades **Tipo** e **Máscara** (Figura 2.1).

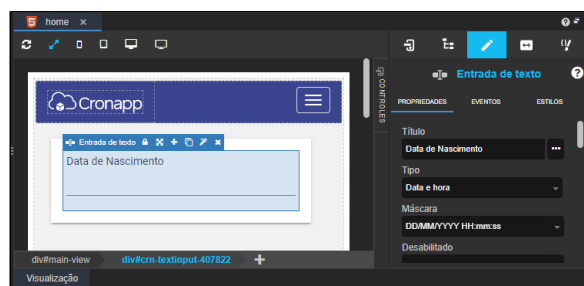


Figura 2.1 - Campos Tipo e Máscara nas propriedades de uma entrada de texto

Também é possível acessar a janela da propriedade **Conteúdo** ou a janela da opção **Editar expressão** para qualquer propriedade, nela podemos definir o **Tipo** e a **Máscaras** que será aplicada (Figura 2.2).

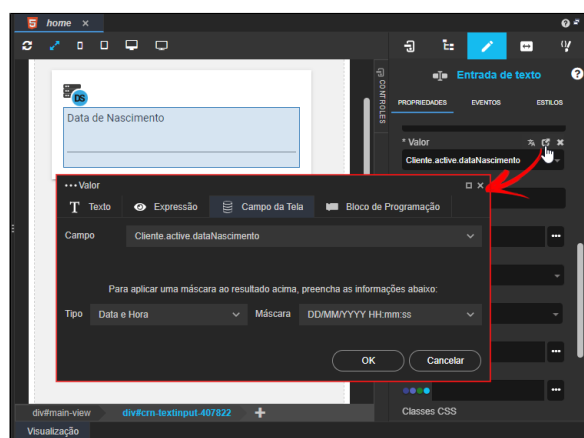


Figura 2.2 - Janela da opção Editar expressão das propriedades

A exibição desses campos será influenciada pelo Tipo e pela Máscara que serão escolhidas. As opções para o Tipo podem ter os seguintes valores para uma data, data e hora ou hora:

- **Data (Date):** exibe a data desconsiderando a hora, mesmo que no diagrama tenha sido usado algum tipo que considere a hora. Se o tipo no diagrama tiver hora, ela será armazenada como 00:00:00. Se a opção **Converter Fuso Horário do Cliente** estiver marcada, esse campo irá exibir o fuso da opção, caso contrário usará o fuso do navegador web ou aplicativo.
- **Data e Hora (Date and Time):** exibe a data e hora. Se a opção **Converter Fuso Horário do Cliente** estiver marcada, esse campo irá exibir o fuso da opção, caso contrário usará o fuso do navegador web ou aplicativo.
- **Data e Hora Locais (Date and Time Local):** exibe a data e hora. Será exibido no fuso local do navegador/aplicativo.
- **Hora (Time):** exibe apenas a hora. Se o tipo no diagrama tiver data, ela será armazenada como 01/01/1970.
- **Hora Local (Time Local):** exibe apenas a hora. Se o tipo no diagrama tiver data, ela será armazenada como 01/01/1970. Será exibido no fuso local do navegador/aplicativo.
- **Mês (Month):** exibe apenas o mês da data (Janeiro, Fevereiro etc.). Ao entrar uma nova data, usará dia, ano e hora atuais.
- **Semana (week):** exibe apenas o dia da semana da data (Segunda-Feira, Terça-Feira etc.). Ao entrar uma nova data, usará mês, ano e hora atuais.

Exemplo: Um campo com a data **22/11/2000 10:00:00 UTC (00:00)**. Se estivermos com a opção **Converter Fuso Horário do Cliente** e o fuso escolhido tiver sido UTC (00:00) e o fuso do cliente seja America/Sao_Paulo (-03:00), teremos os valores para os diferentes tipos:

Tipo	Valor
Date	22/11/2000
Date and Time	22/11/2000 10:00:00
Date and Time Local	22/11/2000 07:00:00 (Será exibido no fuso local do navegador/aplicativo)
Time	10:00:00
Time Local	07:00:00 (Será exibido no fuso local do navegador/aplicativo)
Month	Novembro
Week	Quarta-Feira

A propriedade **Tipo** irá determinar o padrão da janela de escolha rápida de datas, horas, meses, semanas, dias etc. Na Figura 2.3 temos o componente Entrada de texto configurado com o Tipo `Time`.

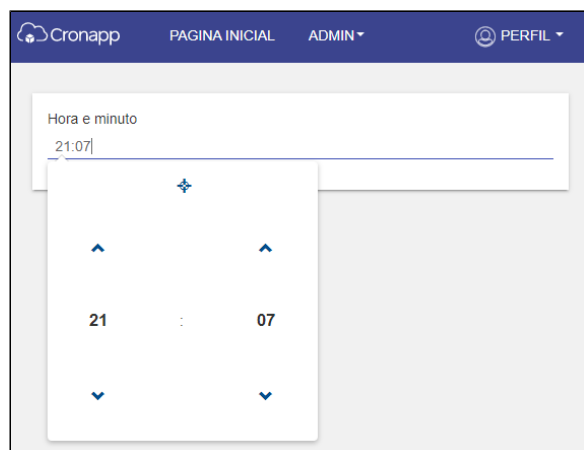


Figura 2.3 - Janela de escolha rápida para o tipo `Time`

A propriedade `Máscara` possui alguns modelos pré-definidos, mas é possível configurá-los manualmente. Saiba mais na documentação [Formatação de máscaras na camada cliente](#).

Bloco de programação

O bloco de programação fornece algumas funções para tratamento de datas. Algumas das mais importantes são:

- **Obter valor do campo:** obtém o valor de qualquer campo da tela cliente no servidor. Todos os campos data e hora trafegam entre o cliente e servidor no formato [ISO 8601](#) que inclui todas as informações de data, hora com minutos, segundos, milissegundos e fuso. Ao trafegar do cliente para o servidor, ele leva as devidas conversões de fuso.
- **Obter campo (Banco de Dados):** essa função obtém o campo do banco de dados que sempre estará no fuso especificado para a aplicação.
- **Exibir Notificação:** essa função bem comum merece uma observação importante, ao exibir uma data. Caso você não formate a data, ela será exibida no fuso horário do cliente.
- **Imprimir:** essa função bem comum merece uma observação importante, ao imprimir uma data. Caso você não formate a data, ela será exibida no formato ISO 8601 no fuso horário do servidor.
- **Converter Texto para Data:** essa função converterá um texto (representação textual de uma data) para um objeto data. Caso uma máscara seja informada, ela a usará para tentar fazer a conversão, caso contrário, algumas máscaras predefinidas serão usadas - levando em consideração o idioma do cliente. Caso o formato textual não considere o fuso na sua escrita (ex.: 12/03/1910), o fuso padrão do servidor será usado. Caso o fuso tenha sido especificado (ex.: 2019-09-22T02:30Z), ele será usado.

- **Formatar Data:** formata uma data para um formato textual - ou seja, para um campo do tipo texto. Caso a máscara seja informada, ele a usará (Figura 4), caso contrário, usará o padrão de máscara do idioma escolhido pelo cliente. Ao formatar para um texto, o Cronapp irá usar o fuso especificado para a aplicação.
- **Formatar Data com Fuso Horário:** tem o mesmo comportamento da função acima, porém o fuso pode ser especificado através do parâmetro da função.

Máscaras

É possível tratar as máscaras no lado Servidor ou Cliente, porém existem diferenças.

Camada Servidor

Na camada Servidor, a especificação de máscaras é diferente do cliente, inserimos o formato "MM-dd-yyyy", onde "MM" é o mês em 2 dígitos, "dd" é o dia em 2 dígitos e "yyyy" é o ano em 4 dígitos. Segue um exemplo do uso da máscara no servidor:

Acesse a documentação [Formatação de máscaras na camada servidor](#) para mais detalhes sobre o uso da máscara na camada Servidor.

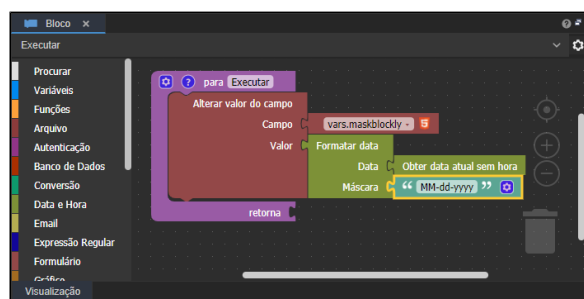


Figura 3.1 - Configurando a máscara através do bloco Formatar data Servidor

Camada Cliente

Na camada Cliente, a forma que é escrita a máscara é diferente do servidor, aqui, inserimos o formato "MM-DD-YYYY", onde "MM" é o mês em 2 dígitos, "DD" é o dia em 2 dígitos e "YYYY" é o ano em 4 dígitos. Segue um exemplo do uso da máscara no servidor:

Acesse a documentação [Formatação de máscaras na camada cliente](#) para mais detalhes sobre o uso da máscara na camada Cliente.

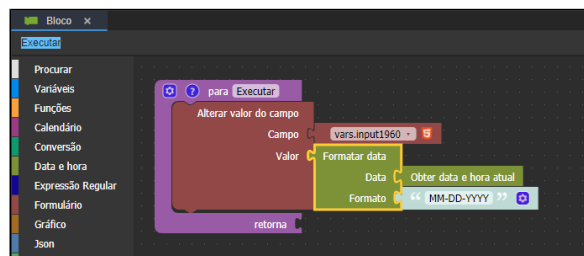


Figura 3.2 - Configurando a máscara através do bloco Formatar data Cliente

Observações

Caso um objeto do tipo data seja enviado para qualquer função que espere um texto, ele será representado pelo formato ISO 8601 no fuso horário especificado para a aplicação.

Ex.: JSON enviado para um serviço REST.

Tenham atenção ao visualizar o formato ISO 8601 das datas. Datas com textos diferentes podem representar o mesmo valor, pois o fuso pode estar escrito de forma diferente:

Ex.: 2019-09-22T00:00:00Z é igual a 2019-09-21T09:00:00-03:00.