Erros mais comuns durante a construção

- **<Classe> could not be resolved to a type**
- **Tradução / O que quer dizer**
- <Classe> não pôde ser resolvido para um tipo
- O compilador não encontrou a classe com o nome <Classe>
- **Soluções**
- Verifique se não digitou o nome da classe errado
- Verifique se importou a classe corretamente (sim, até nas páginas JSP precisamos importar as classes)
- Para importar as classes corretamente, é preciso que as mesmas estejam dentro de um pacote nomeado (não pode ser o "pacote padrão", sem nome).

- **NoClassDefFoundError ou ClassNotFoundException**
- **Tradução / O que quer dizer**
- A classe citada não foi encontrada após compilação
- **Solução**
- Provavelmente, a classe não foi compilada corretamente. Limpe completamente (projeto e server) e tente executar novamente.

- **NullPointerException**
- **Tradução / O que quer dizer**
- Exceção de ponteiro nulo
- Você está tentando acessar um atributo ou método de uma variável que não está apontando para objeto nenhum (isto é, apontando para null). Exemplo:
- int id = usuario.getId();
- **Soluções**
- Identifique em que linha e com qual variável isso ocorreu (a mensagem de erro pode ajudar nisso)
- Implemente mecanismos para evitar verificar atributos ou métodos da mesma se ela estiver nula. Exemplo:
- int id = 0;
- if (usuario != null) {
- id = usuario.getId();
- }

- **ParseInt ... Exception**
- **Tradução / O que quer dizer**

- Exceção (erro) durante parse (converter string em outro tipo) para inteiro
- O programa não conseguiu converter o valor recebido em inteiro - o valor recebido é null ou então a String não corresponde a um valor inteiro.
- **Soluções**
- Pode-se capturar a exceção com um bloco try... catch;
- Melhoria - implementar uma classe que execute o try...catch e retorne 0 caso não seja possível a conversão do valor.

- **SQLDateException**
- **Tradução / O que quer dizer**
- Você está tentando salvar uma String em um campo (no #BancoDeDados) que só aceita Date
- **Solução**
- Em Java, precisamos converter a String em java.util.Date e depois em java.sql.Date para poder armazenar no Banco de Dados
- Pode-se fazer a conversão da seguinte forma:
- DateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
- java.util.Date jDate = (java.util.Date) formatter.parse(<dataComoString>);
- java.sql.Date sqlDate = new java.sql.Date(jDate.getTime());
- Melhoria - implementar uma classe para converter String em java.sql.Date e vice-versa;

- **PSQLException: ERROR: column <Coluna> does not exist**
- **Tradução / O que quer dizer**
- Uma consulta SQL está tentando acessar uma coluna que não existe em uma tabela citada na mesma
- **Solução**
- Identifique a consulta SQL e compare os campos de cada tabela listada com os campos das tabelas no #BancoDeDados

- **PSQLException: A nome da coluna <Coluna> não foi encontrado neste ResultSet.**
- **Tradução / O que quer dizer**
- **Solução**
- Verifique se o nome da coluna acessado no resultSet foi realmente retornado da consulta (nome pode estar errado ou não incluído no select);