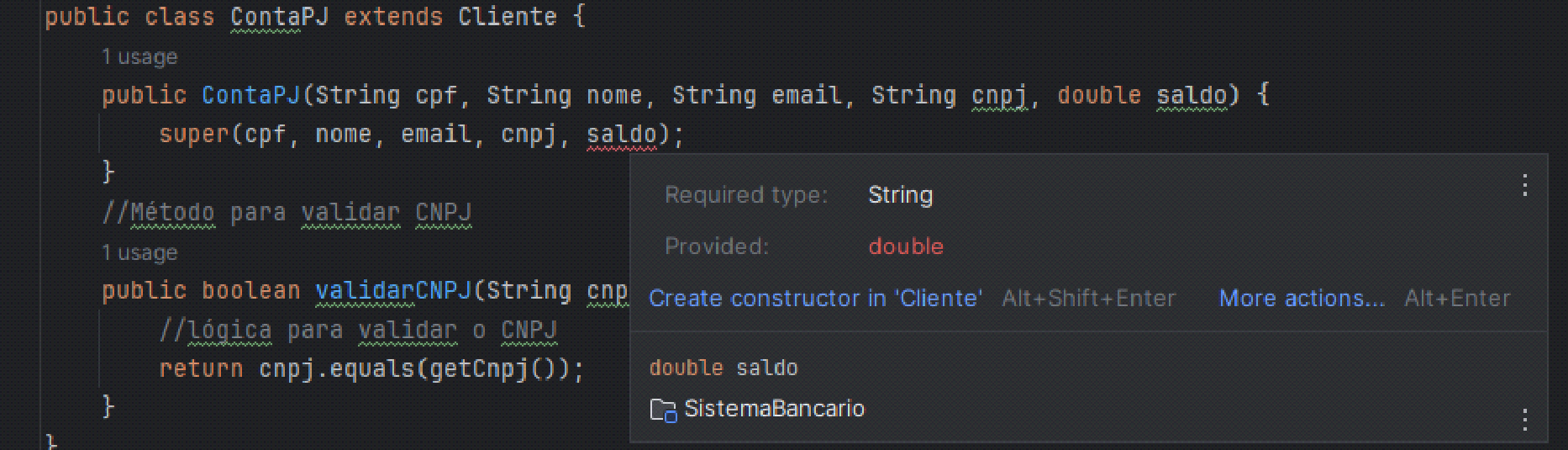
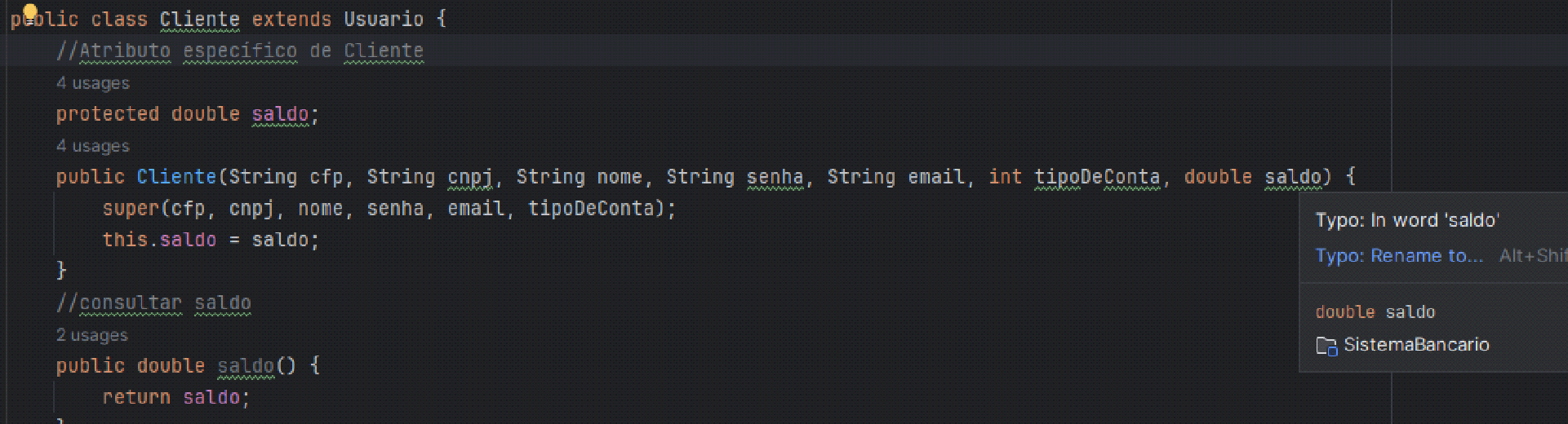
Tem um erro nas classes "contaCorrente", "contaPJ" e "contaPoupanca" que não consegui resolver de jeito nenhum.



Esse erro aparece nessas 3 como se solicitasse o tipo string pra saldo, sendo que onde ela foi criada, tem o tipo double.

O Intellij mostra uma solução, que é criar um construtor na classe cliente e colocar o double lá, mas o construtor já está criado.



Tentei de MUITAS formas diferentes ajeitar esse problema mas não consegui.

O código não está funcional por conta desse problema, mas creio eu que fora isso, não tenha mas nenhum outro problema que impeça a compilação do código.

Há alguns métodos que não estão sendo utilizados mesmo criados, porque vamos precisar da api mais pra frente para utilizar eles de forma funcional e útil.

----------------------------------------------------------------------------------------------------------------

Alguns comentários e "checks" que não foram feitos no código, vou fazer aqui, explicações e motivos, pois se for fazer todas essas explicações em javadoc no código, ficaria muito poluído. (Se isso for um problema, desculpa, na próxima faremos de um jeito diferente).

Definição de classes, atributos e métodos:  
Contém em todas as classes feitas.

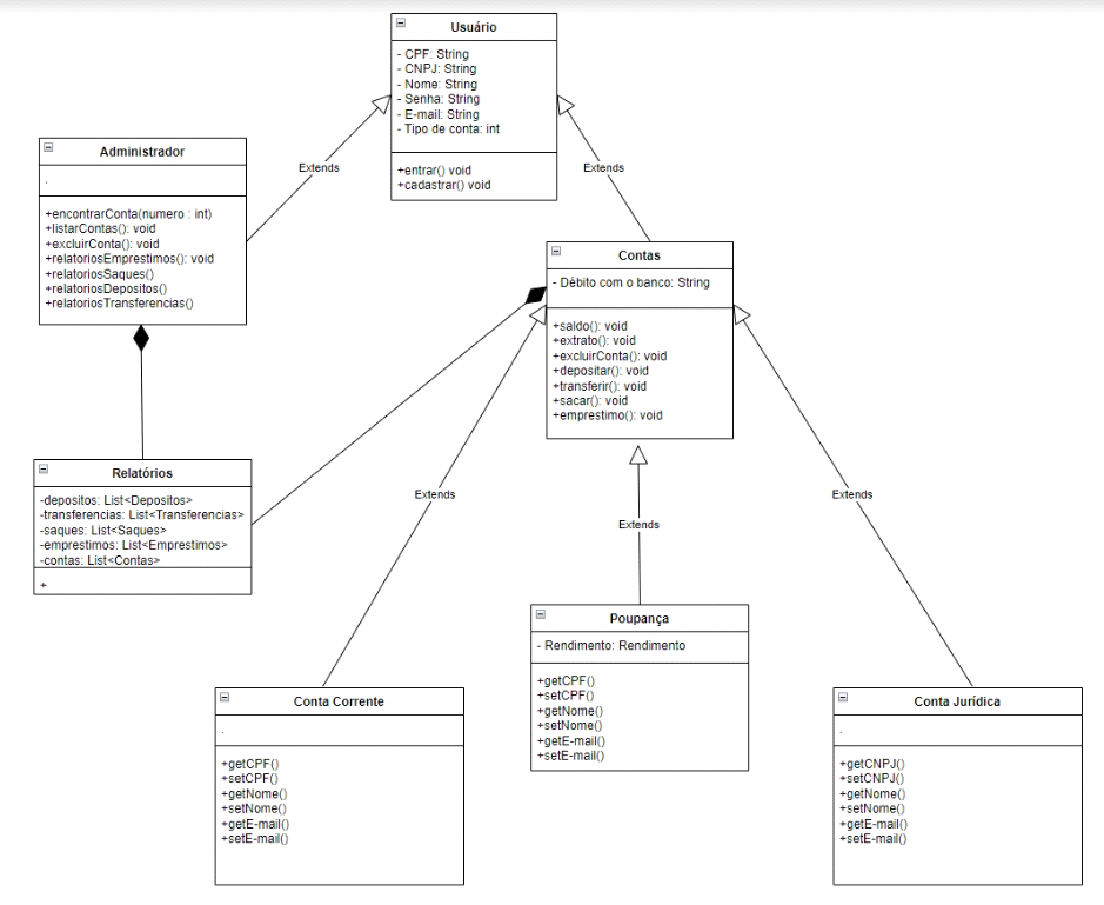
Controle de acesso aos atributos e métodos:  
Há modificadores de acesso onde necessário. Utilizando publico para métodos de acesso onde quero que seja acessado por todos, private onde não quero que seja acessível em outras classes e protected para um meio tempo onde quero acessar em subclasses.

Sobrecarga (overload) de método:  
Há mais de um overload sendo utilizado no código, como por exemplo, na classe "Cliente".  
Essa classe possui dois métodos com o mesmo nome mas com parâmetros diferentes.  
Método "saldo()", construtor "Cliente". Ambos tem o nome "saldo", mas um é o método que retorna o valor de saldo sendo double e o outro é um construtor.

Atributos e métodos static:  
Foi utilizado método static na classe "ContaPoupanca", para declarar a variável "taxaRendimento" como uma variável de classe, para ela pertencer a ContaPoupanca e não a instâncias individuais.  
Nesse caso foi utilizado static final, para que ela não possa mais ser alterada e que o valor seja constante, pois a taxa de rendimento foi feita para não ser alterada.

Utilização de coleções (e.g. ArrayList):  
Foi utilizado arraylist para armazenar os dados de saques, transferências, depósitos e empréstimos. Utilizado na classe Administrador que "extends" de usuário, contém o acúmulo de dados de forma dinâmica em arraylists.

Herança:  
Todas as classes contém alguma hierarquia. Essa hierarquia está contida no diagrama de classes a seguir.



Alguns métodos são utilizados em mais de uma classe, assim, é utilizado a hierarquia para pegar os valores de uma classe superior. Um exemplo disso é a utilização de dados contidos em "ContaCorrente" e "ContaPoupanca" ao mesmo tempo, assim, esses dados são contidos em "Contas" e acessados com "super()" nas subclasses.