# LAB 3 – Orientação a Objetos

Neste laboratório vamos criar uma aplicação que utiliza os recursos de orientação a objetos.

## Exercício 1 – Utilizando Construtores

Neste exercício iremos criar uma classe com diferentes construtores.

1. Adicione um projeto do tipo ***Class Library***à *Solution* com o nome “Lab3.ClassLibrary”.
2. Adicione um novo item ao projeto, do tipo ***Class*** e com o nome “ContaCorrente”.
3. Adicione dois atributos à classe “ContaCorrente”:
   * Atributo *privativo* chamado ***nome*** do tipo ***string*;**
   * Atributo *privativo* chamado ***saldo*** do tipo ***decimal*;**
4. Adicione dois construtores à classe “ContaCorrente”:
   * O primeiro construtor recebe apenas o *nome do correntista*;
   * O segundo construtor deve receber dois parâmetros: *nome* e *saldo inicial* da conta
5. Compile o projeto.

## Exercício 2 – Utilizando Métodos e Propriedades

Nesta etapa iremos criar métodos na classe “ContaCorrente”.

1. Acrescente um método para *depósito* de valores na conta:

public void Depositar(decimal val)

{

saldo = saldo + val;

}

1. Acrescente um método para *saque* de valores da conta:

public void Sacar(decimal val)

{

if (val <= saldo)

saldo = saldo - val;

}

1. Adicione as propriedades abaixo:

public string Nome

{

get { return nome; }

}

public decimal Saldo

{

get { return saldo; }

}

1. No projeto adicione uma *referência* ao projeto “Lab3.ClassLibrary”.
2. Encontre o método ***Page\_Load*** dentro do formulário web (“Default.aspx.cs”) e adicione o seguinte código:

ContaCorrente minhaConta = new ContaCorrente("Gerson");

Response.Write("Movimentações conta: " + minhaConta.Nome + "<br/>");

Response.Write(minhaConta.Saldo + "<br/>");

minhaConta.Depositar(100);

Response.Write(minhaConta.Saldo + "<br/>");

minhaConta.Sacar(50);

Response.Write(minhaConta.Saldo + "<br/>");

1. Compile e execute a aplicação.

## Exercício 3 – Adicionando atributos

1. Altere a classe “ContaCorrente” de forma que ela possua atributos para armazenar a *data de criação* e *data da última movimentação*.
2. A *data de criação* deve ser gerada no momento da *inicialização* da conta (“DateTime.Now”) e a atualização sempre que houver movimentação da conta.
3. Crie também *métodos de* *acesso* a estes atributos e altere o método ***Page\_Load*** para testar as alterações.