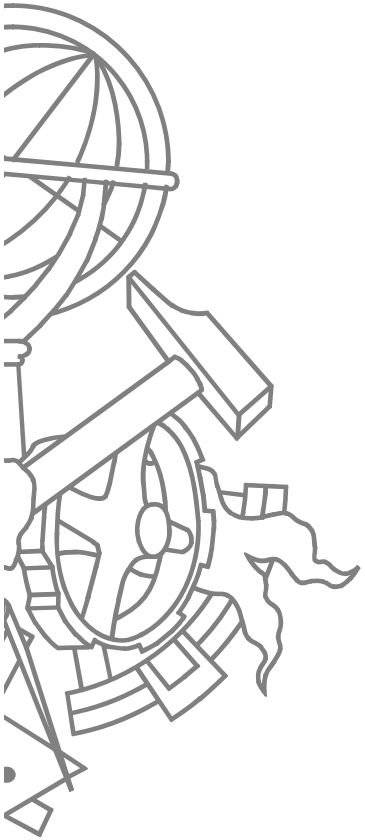
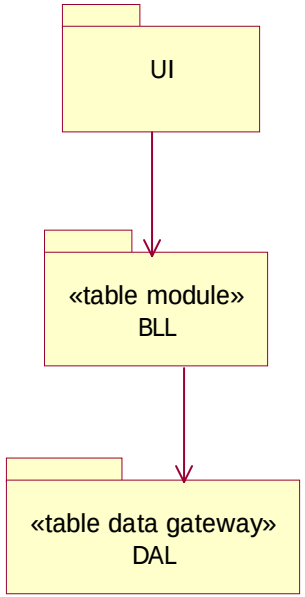
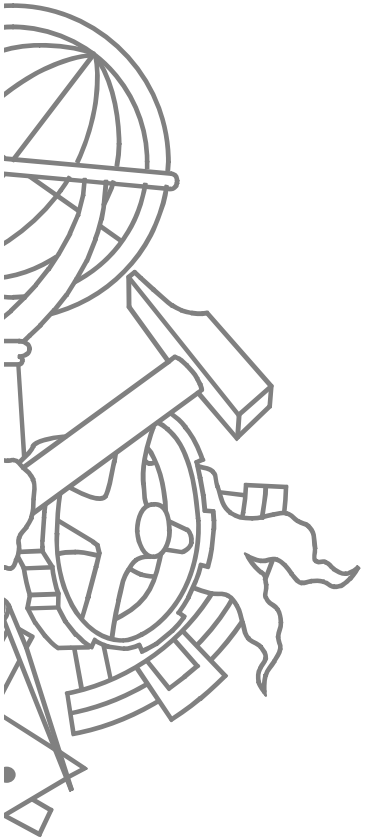
Table Module com Table Data Gateway usando Result Set

Packages





Classes BLL



Contract

+ Contract ( )
+ RecognizedRevenue ([in] contractID : int , [in] asOf : DateTime) : Money
+ CalculateRevenueRecognitions ([in] contractID : int)
+ GetContracts ( ) : DataSet
+ GetContractsByProduct ([in] productID : int) : DataSet
+ GetContractsByCustomer ([in] customerID : int) : DataSet

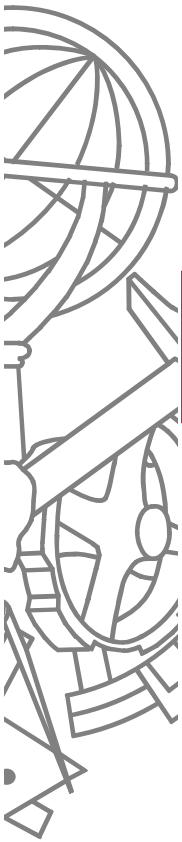
Customer

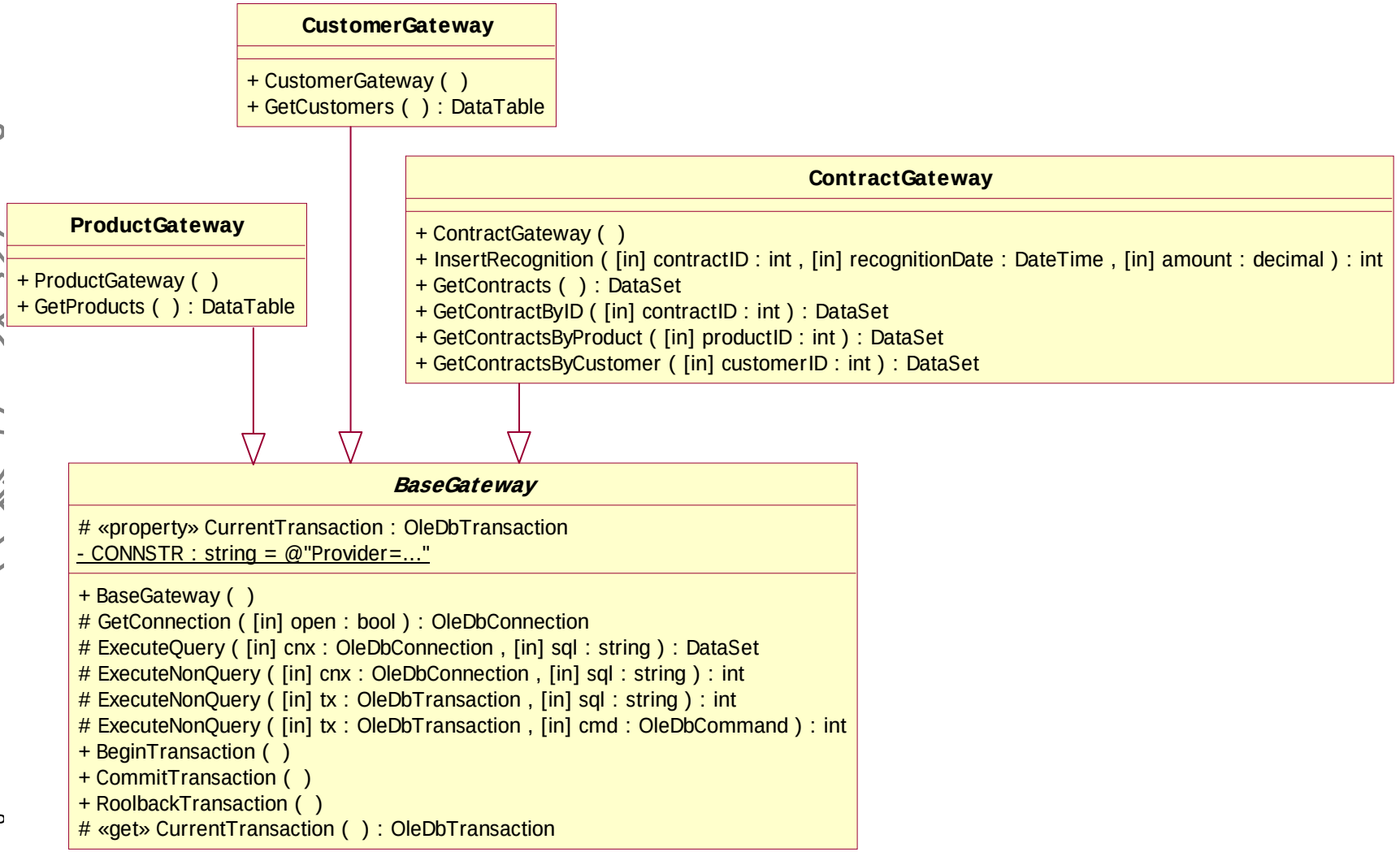
+ Customer ( )
+ GetCustomers ( ) : DataTable

Product

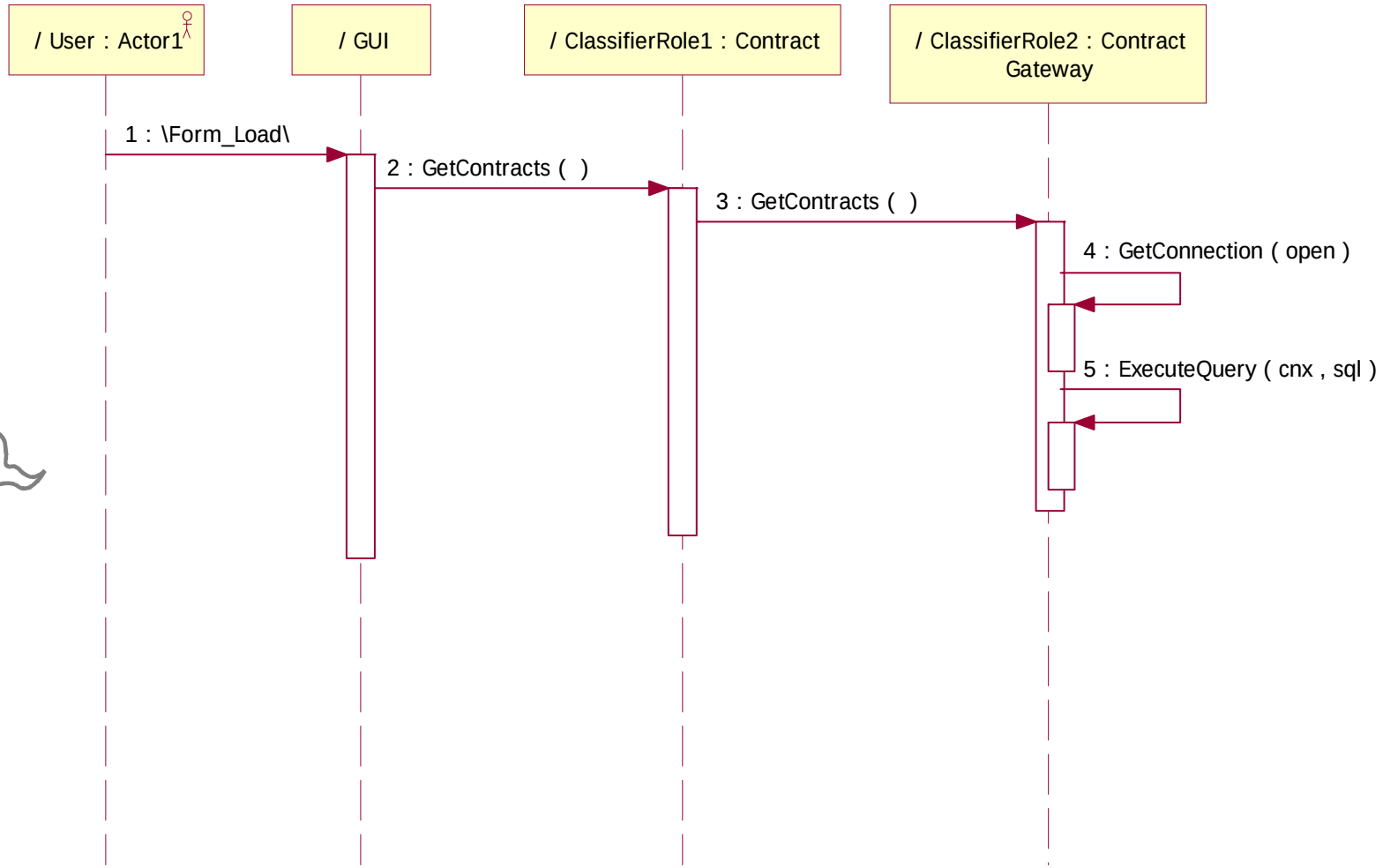
+ Product ( )
+ GetProducts ( ) : DataTable

Classes DAL

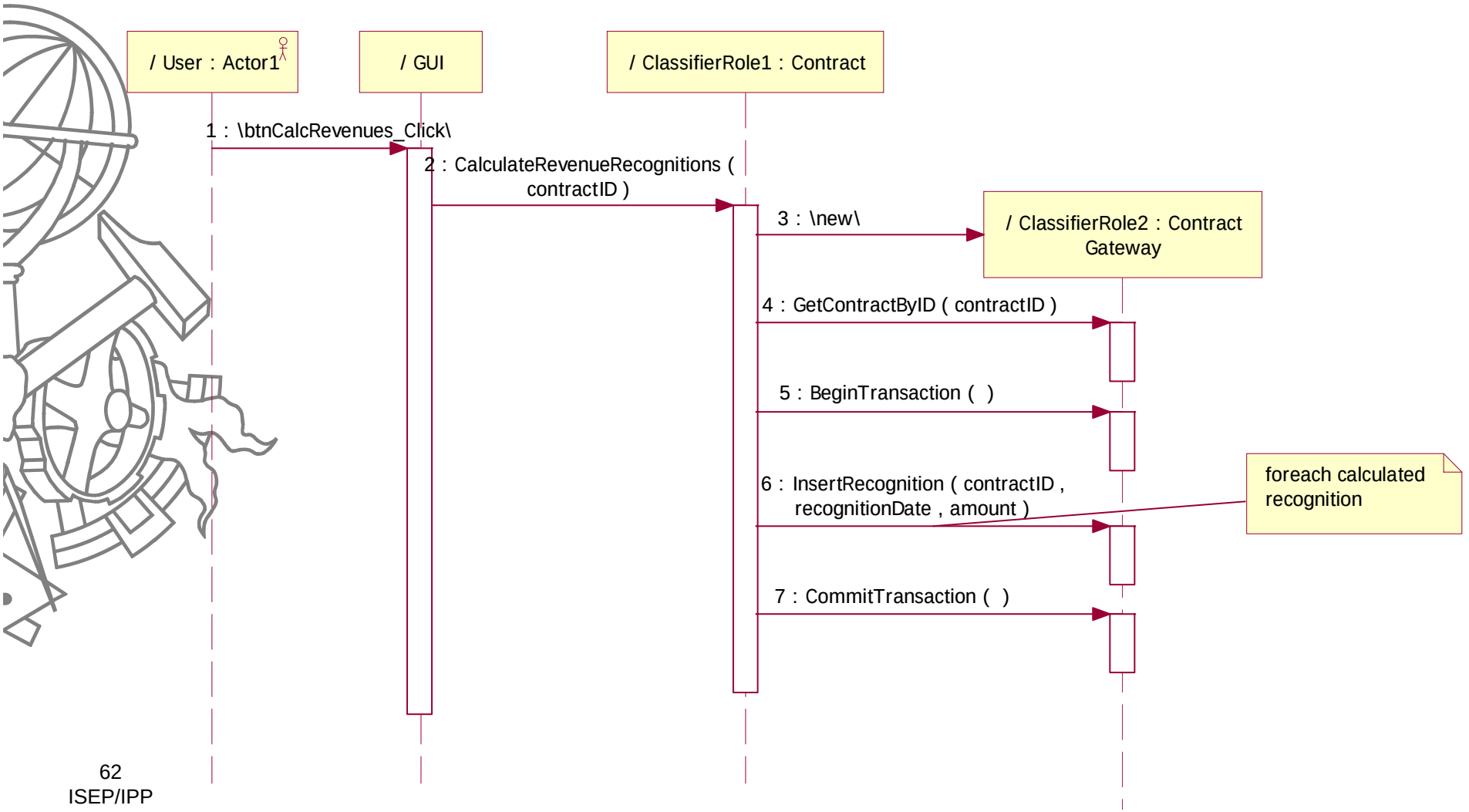



CustomerGateway
+ CustomerGateway ( )
+ GetCustomers ( ) : DataTable
ProductGateway
+ ProductGateway ( )
+ GetProducts ( ) : DataTable
ContractGateway
+ ContractGateway ( )
+ InsertRecognition ([in] contractID : int , [in] recognitionDate : DateTime , [in] amount : decimal) : int
+ GetContracts ( ) : DataSet
+ GetContractByID ([in] contractID : int) : DataSet
+ GetContractsByProduct ([in] productID : int) : DataSet
+ GetContractsByCustomer ([in] customerID : int) : DataSet
BaseGateway
«property» CurrentTransaction : OleDbTransaction
- CONNSTR : string = @"Provider=..."
+ BaseGateway ( )
# GetConnection ([in] open : bool) : OleDbConnection
# ExecuteQuery ([in] cnx : OleDbConnection , [in] sql : string) : DataSet
# ExecuteNonQuery ([in] cnx : OleDbConnection , [in] sql : string) : int
# ExecuteNonQuery ([in] tx : OleDbTransaction , [in] sql : string) : int
# ExecuteNonQuery ([in] tx : OleDbTransaction , [in] cmd : OleDbCommand) : int
+ BeginTransaction ( )
+ CommitTransaction ( )
+ RoolbackTransaction ( )
# «get» CurrentTransaction ( ) : OleDbTransaction

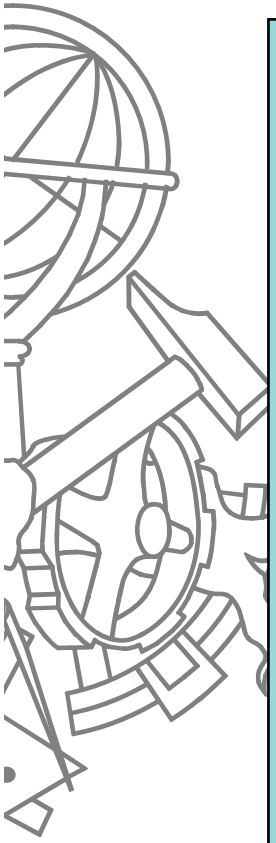
Sequência GetContracts



Sequência CalculateRevenues



CalculateRevenues()

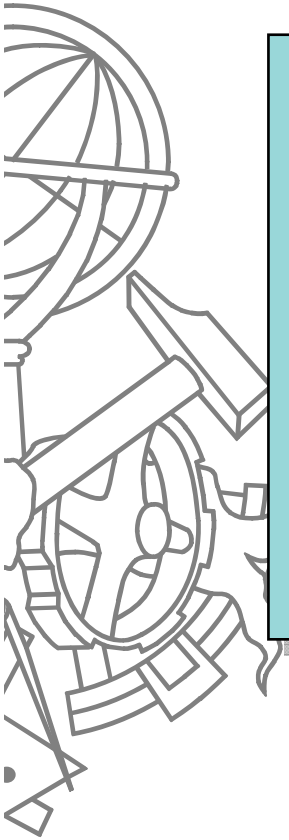


```
public void CalculateRevenueRecognitions(int contractID) {
    DAL.ContractGateway dal = new DAL.ContractGateway();
    DataSet ds = dal.GetContractByID(contractID);

    string prodType = (string)ds.Tables[0].Rows[0]["type"];
    decimal totalRevenue = (decimal)ds.Tables[0].Rows[0]["revenue"];
    DateTime recDate = (DateTime)ds.Tables[0].Rows[0]["dateSigned"];

    dal.BeginTransaction();
    switch (prodType) {
    case "PT":
          dal.InsertRecognition(contractID, recognitionDate, totalRevenue);
          break;
    case "FC":
          decimal[] allocs = Money.Allocate(totalRevenue, 3);
          dal.InsertRecognition(contractID, recDate, allocs[0]);
          dal.InsertRecognition(contractID, recDate.AddDays(60), allocs[1]);
          dal.InsertRecognition(contractID, recDate.AddDays(90), allocs[2]);
          break;
    case "BD":
          decimal[] allocs = Money.Allocate(totalRevenue, 3);
          dal.InsertRecognition(contractID, recDate, allocs[0]);
          dal.InsertRecognition(contractID, recDate.AddDays(30), allocs[1]);
          dal.InsertRecognition(contractID, recDate.AddDays(60), allocs[2]);
          break;
    }
    dal.CommitTransaction();
}
```

CalculateRevenues()

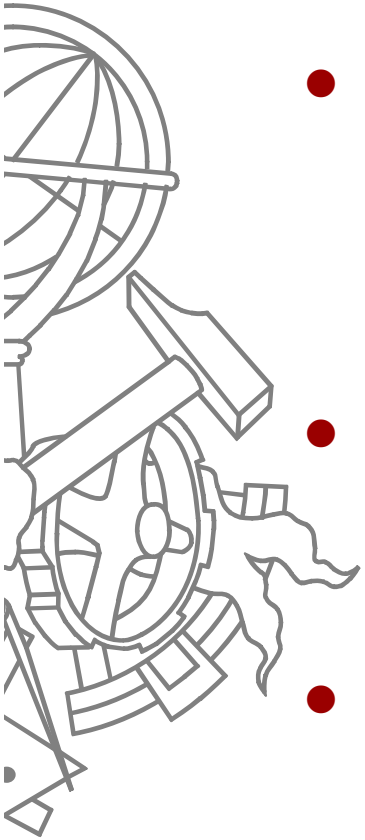


```
public int InsertRecognition(int contractID,
                             DateTime recognitionDate,
                             decimal amount)
{
    OleDbCommand sqlcmd = new OleDbCommand("INSERT INTO TRevenueRecognitions
    (contractID, dateRecognition, amount) VALUES (?, ?, ?)",
    CurrentTransation.Connection, CurrentTransation);

    sqlcmd.Parameters.Add("@cid", contractID);
    sqlcmd.Parameters.Add("@dt", recognitionDate);
    sqlcmd.Parameters.Add("@amt", amount);

    return ExecuteNonQuery(CurrentTransation, sqlcmd);
}
```

Notas



- Método **`ContractGateway.GetContractByID`** retorna um DataSet resultante do *join* da tabela **`TContracts`** com **`TProducts`**
- Para garantir coerência e consistência das alterações na BD, a classe Gateway tem que guardar estado relativo à transação a usar
- Controlo de transacções dá uma discussão separada...