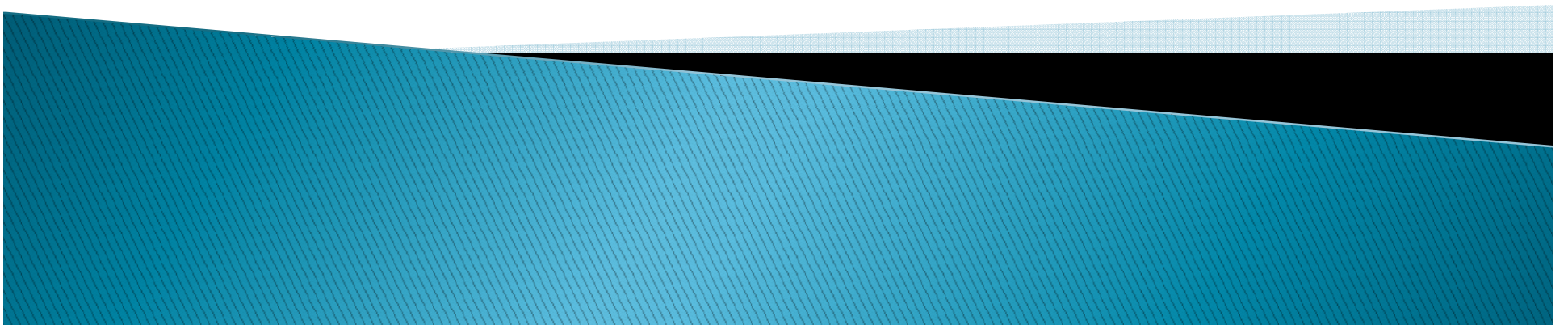


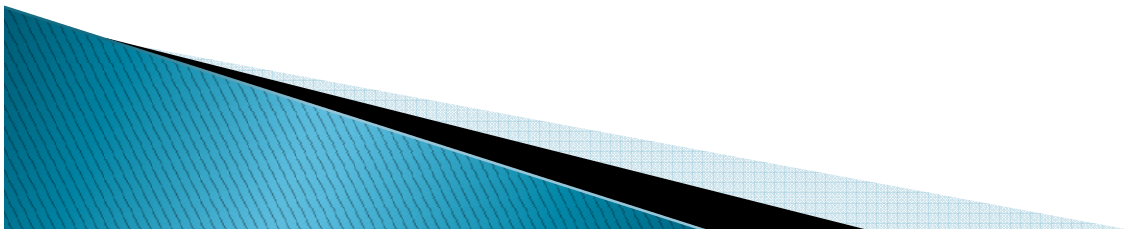
Suporte a banco de dados PostgreSQL no TerraME

Luiz Gustavo Diniz de Oliveira Veras
Banco de Dados Geográfico CAP-349



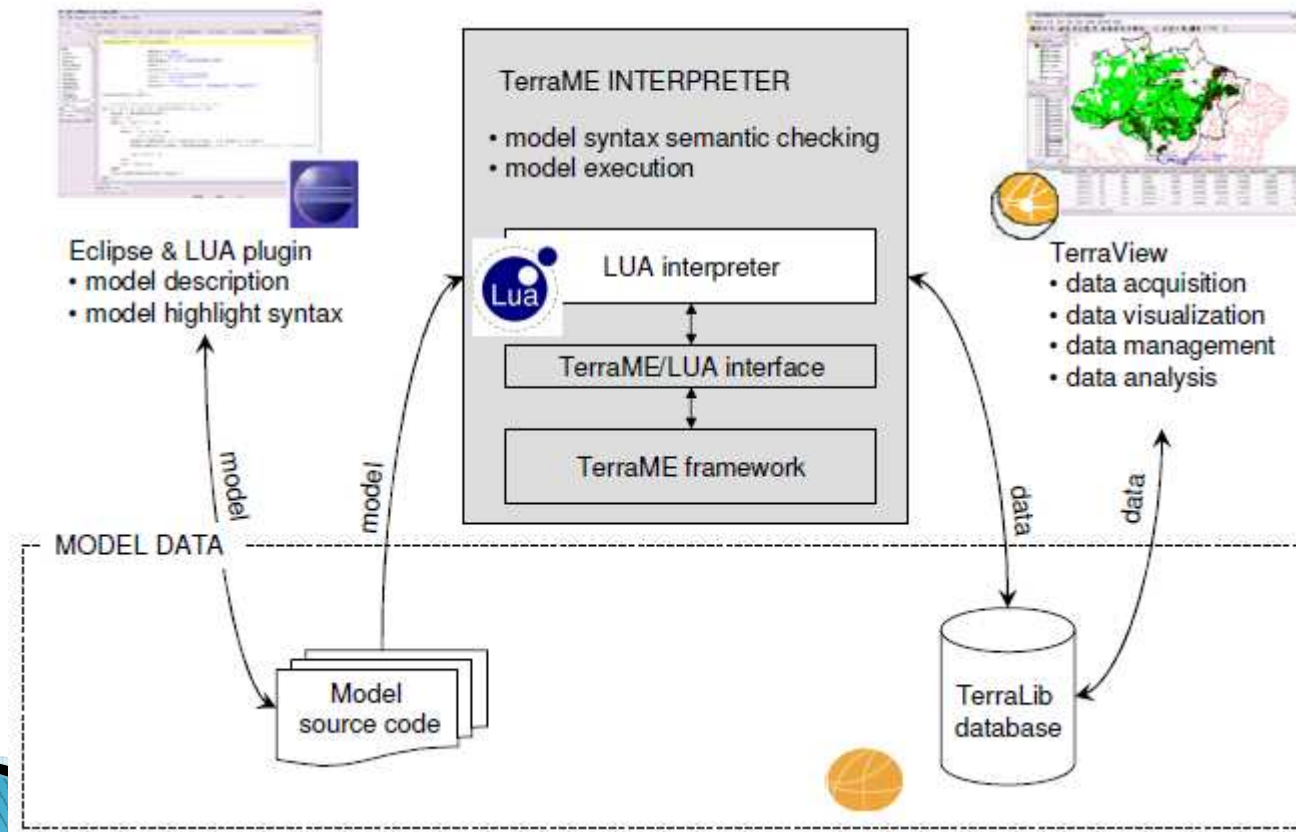
OBJETIVOS

- ▶ Implementar suporte para PostgreSQL no TerraME;
- ▶ Executar um modelo dinâmico carregando dados de um banco PostgreSQL no TerraME;
- ▶ Avaliar a velocidade entre os bancos Access, MySQL e PostgreSQL em ambiente TerraME.

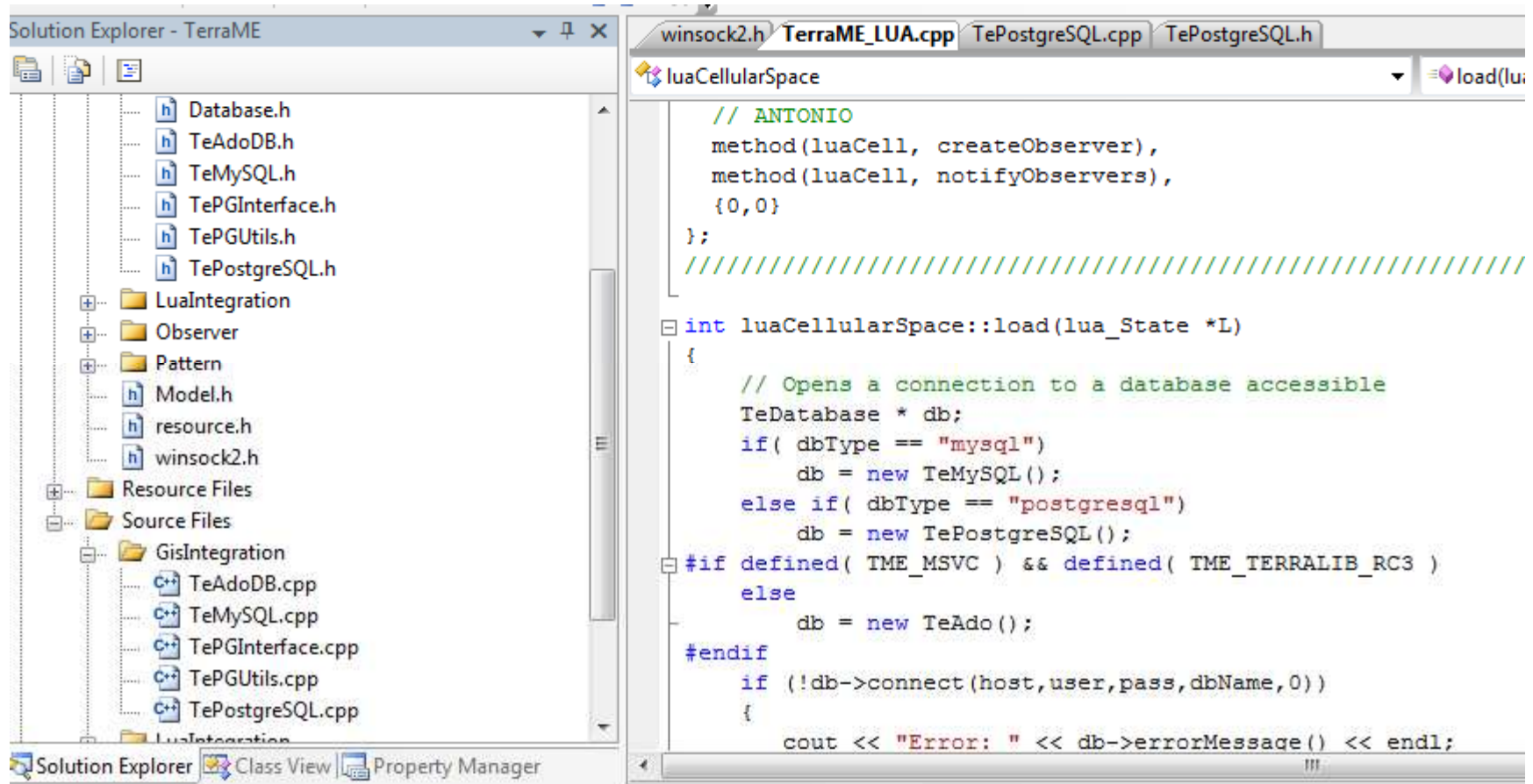


TerraMe

- ▶ Ambiente de desenvolvimento para modelagem espacial dinâmica.



Código TerraMe



The image shows a screenshot of the Visual Studio IDE. On the left, the Solution Explorer displays the project structure for 'TerraME'. The project is organized into several folders: 'Database.h' (containing TeAdoDB.h, TeMySQL.h, TePGInterface.h, TePGUtils.h, and TePostgreSQL.h), 'LuaIntegration', 'Observer', 'Pattern', 'Model.h', 'resource.h', 'winsock2.h', 'Resource Files', 'Source Files', and 'GisIntegration' (containing TeAdoDB.cpp, TeMySQL.cpp, TePGInterface.cpp, TePGUtils.cpp, and TePostgreSQL.cpp). The main editor window shows the 'luaCellularSpace.cpp' file. The code defines a class 'luaCellularSpace' with a 'load' method. The 'load' method opens a connection to a database based on the 'dbType' parameter. It handles 'mysql' and 'postgresql' types, and falls back to 'TeAdo()' if the database type is not recognized. The code also includes error handling for connection failures.

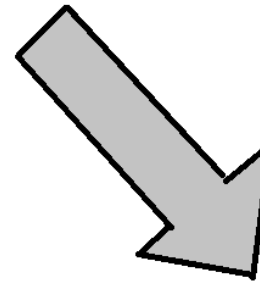
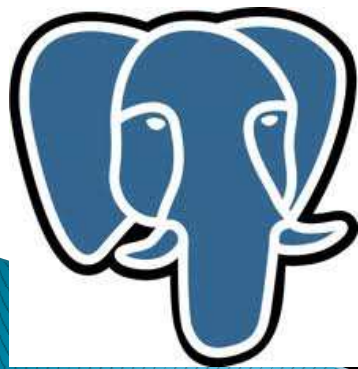
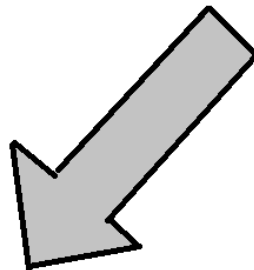
```
winsock2.h TerraME_LUA.cpp TePostgreSQL.cpp TePostgreSQL.h
luaCellularSpace
// ANTONIO
method(luaCell, createObserver),
method(luaCell, notifyObservers),
{0,0}
};
////////////////////////////////////

int luaCellularSpace::load(lua_State *L)
{
    // Opens a connection to a database accessible
    TeDatabase * db;
    if( dbType == "mysql")
        db = new TeMySQL();
    else if( dbType == "postgresql")
        db = new TePostgreSQL();
    #if defined( TME_MSVC ) && defined( TME_TERRALIB_RC3 )
    else
        db = new TeAdo();
    #endif
    if (!db->connect(host,user,pass,dbName,0))
    {
        cout << "Error: " << db->errorMessage() << endl;
    }
}
```

Conversão de banco

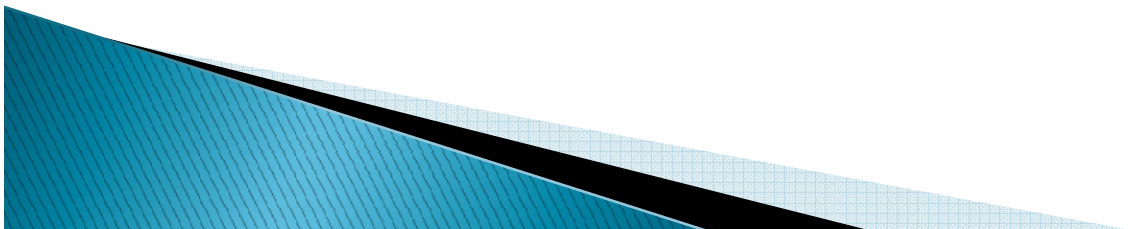


Terraview

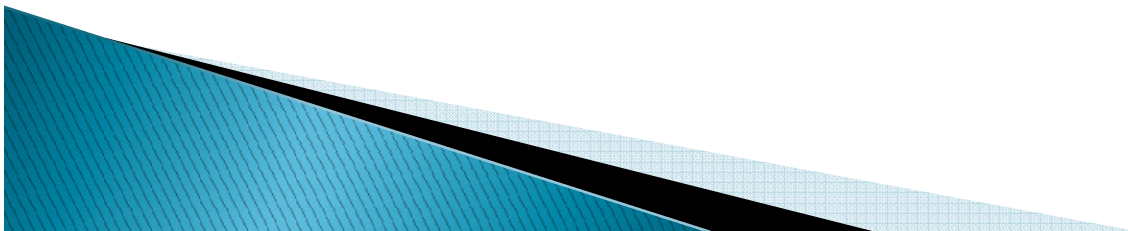
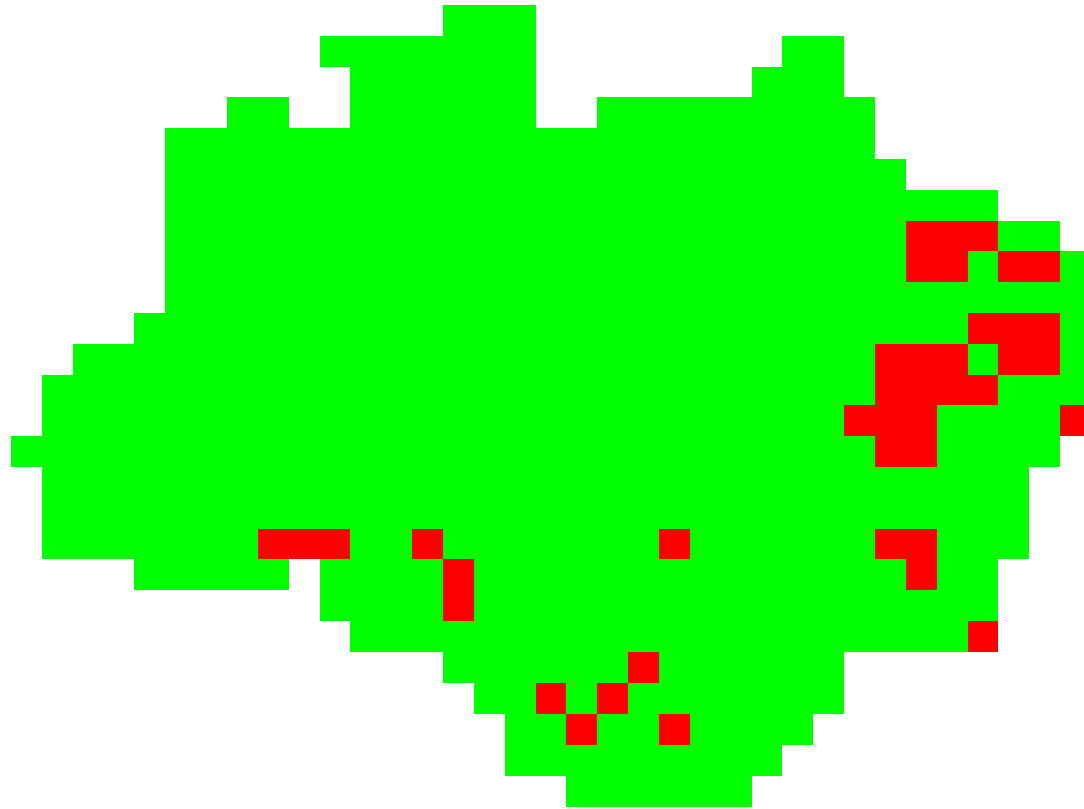


Conexão do banco no TerraME

```
cs = CellularSpace{  
  >> dbType = "postgresql",  
  >> database = "modelo",  
  >> theme = "dinamica",  
  >> host = "localhost",  
  >> user = "postgres",  
  >> password = "123",  
  >> select = {"defor", "dist_urban_areas", "dist_roads"}  
}
```

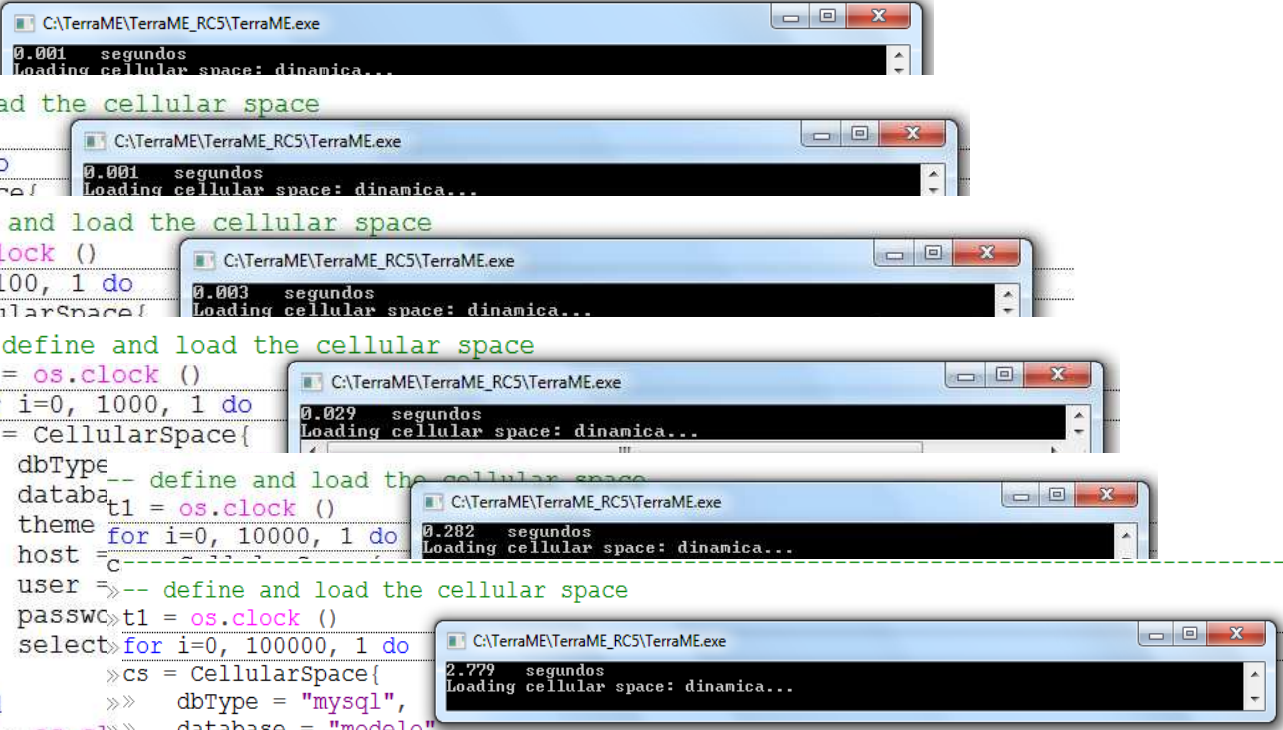


Execução do Modelo



Testes com MySql

```
-- define and load the cellular space
t1 = os.clock ()
--for i=0, 1000, 1 do
cs = CellularSpace{
  >> dbType = "mysql"
  >> da-- define and load the cellular space
  >> tht1 = os.clock ()
  >> ho-- define and load the cellular space
  >> for i=0, 10, 1 do
  >> uscs = CellularSpace{
  >> pa-- define and load the cellular space
  >> se-- define and load the cellular space
  >> datt1 = os.clock ()
  >> thefor i=0, 100, 1 do
--end >> hoscs = CellularSpace{
t2 = os>> use-- define and load the cellular space
print(>> pas-- define and load the cellular space
  >> sel-- define and load the cellular space
  >> thefor i=0, 1000, 1 do
  >> hoscs = CellularSpace{
end >> use-- define and load the cellular space
t2 = os>> pas-- define and load the cellular space
print(t>> sel-- define and load the cellular space
  >> thefor i=0, 10000, 1 do
  >> host = "localhost",
  >> user = "root",
  >> password = "mago2009",
  >> select = {"defor", "dist_urban_areas", "dist_roads"}
  >> }
end >> use-- define and load the cellular space
t2 = os.c>> pas-- define and load the cellular space
print(t2-t>> sel-- define and load the cellular space
  >> thefor i=0, 100000, 1 do
  >> cs = CellularSpace{
  >>>> dbType = "mysql",
  >>>> database = "modelo",
  >>>> theme = "dinamica",
  >>>> host = "localhost",
  >>>> user = "root",
  >>>> password = "mago2009",
  >>>> select = {"defor", "dist_urban_areas", "dist_roads"}
  >>>> }
end >> use-- define and load the cellular space
t2 = os.clock ()
print(t2-t1,"segundos")
```



Teste com Access(mdb)

```
-- define and load the c
t1 = os.clock ()
--for i=0, 100000, 1 do
cs = CellularSpace{
  >> database = INPUT_PATH.."amazonia.mdb",
  >> dbType = -- define and load
  >> theme = "t1 = os.clock ()
  >> select =
  for i=0, 10, 1 do
  cs = CellularSpace{
    >> database = INPUT_PATH.."amazonia.mdb",
    >> dbType = "ADO"
    >> theme = "dinar
    >> select = {"def
  }
  end
  t2 = os.clock ()
  print(t2-t1,"segun

  >> t1 = os.clock ()
  >> for i=0, 100, 1 do
  cs = CellularSpace{
    >> database = INPUT_PATH.."amazonia.mdb",
    >> dbType = "ADO"
    >> theme = "dinar
    >> select = {"def
  }
  end
  t2 = os.c
  print(t2-t1,"s

  >> t1 = os.clock ()
  >> for i=0, 10000, 1 do
  cs = CellularSpace{
    >> database = INPUT_PATH.."amazonia.mdb",
    >> dbType = -- define and load th
    >> theme = "d
    >> select = {"defor", "dist_urban_areas", "dist_roads"
  }
  end
  t2 = os.cloc
  print(t2-t1,"s

  >> t1 = os.clock ()
  >> for i=0, 100000, 1 do
  cs = CellularSpace{
    >> database = INPUT_PATH.."amazonia.mdb",
    >> dbType = "ADO",
    >> theme = "dinamica",
    >> select = {"defor", "dist_urban_areas", "dist_roads"
  }
  end
  t2 = os.clock ()
  print(t2-t1,"segundos")
```

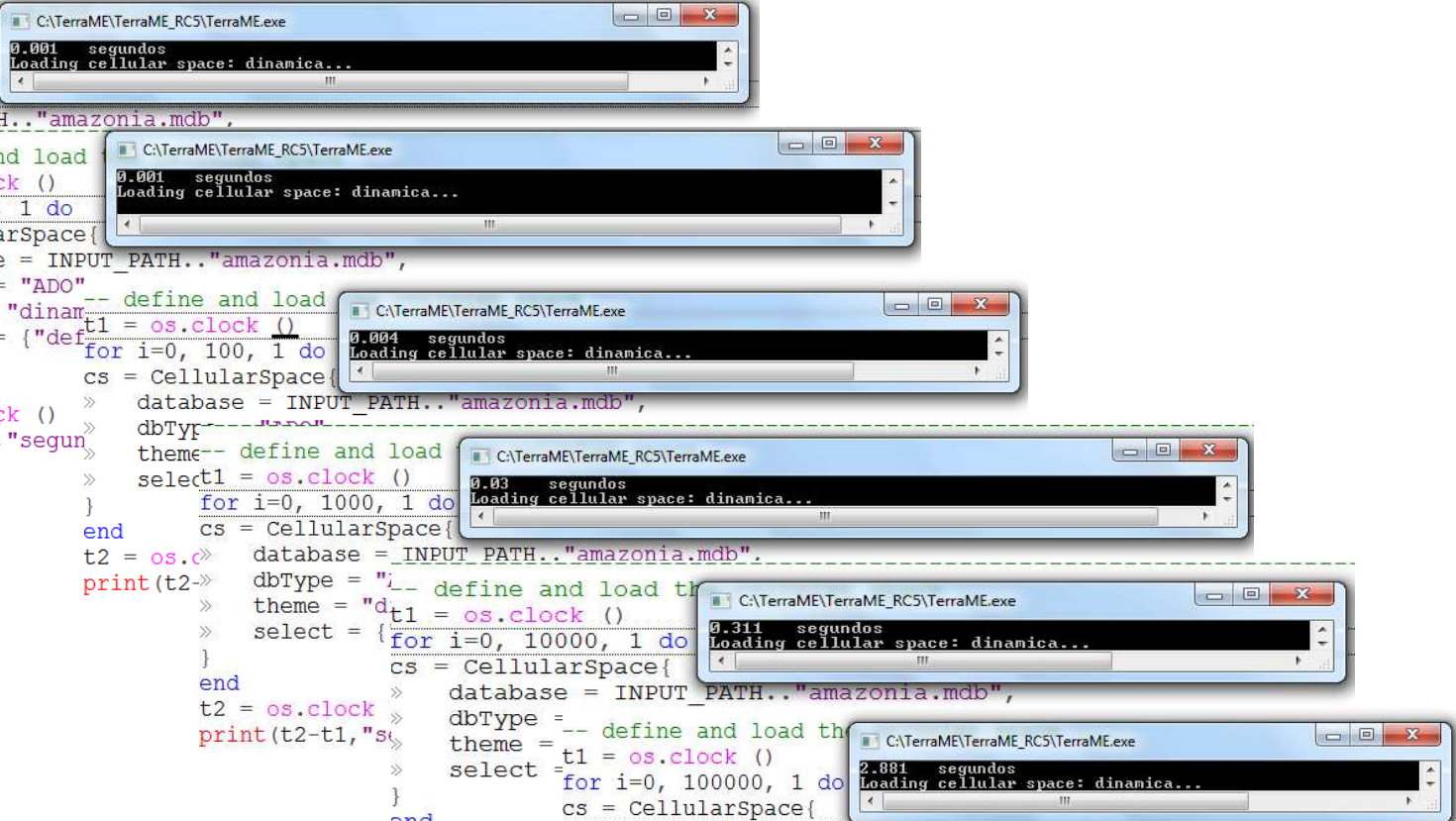
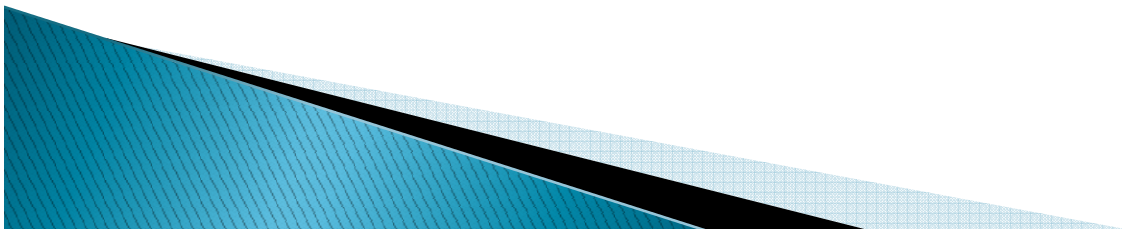


Gráfico de Tempo x Iterações



Tempo de inicialização do espaço celular por banco de dados



Obrigado!

