

HPCC Systems

Processamento e análise de big data

Alysson Oliveira e Hugo Watanuki



Bem-vindo! – Agenda do curso

✓ **HPCC Systems: Visão geral**

- ✓ O que é?
- ✓ Para que serve?

✓ **Tutorial: Machine Learning com HPCC**

- ✓ Aprendizagem supervisionada
- ✓ Previsão de preços de imóveis

✓ **Próximos passos**

- ✓ Cursos online
- ✓ Projetos de pesquisa
- ✓ Oportunidades profissionais



HPCC Systems: Visão geral

Conceito de Big Data

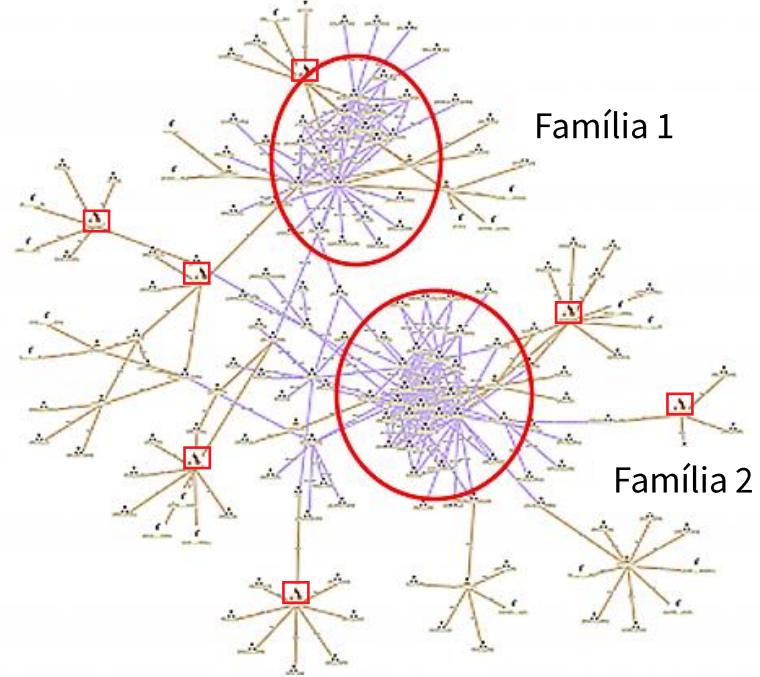
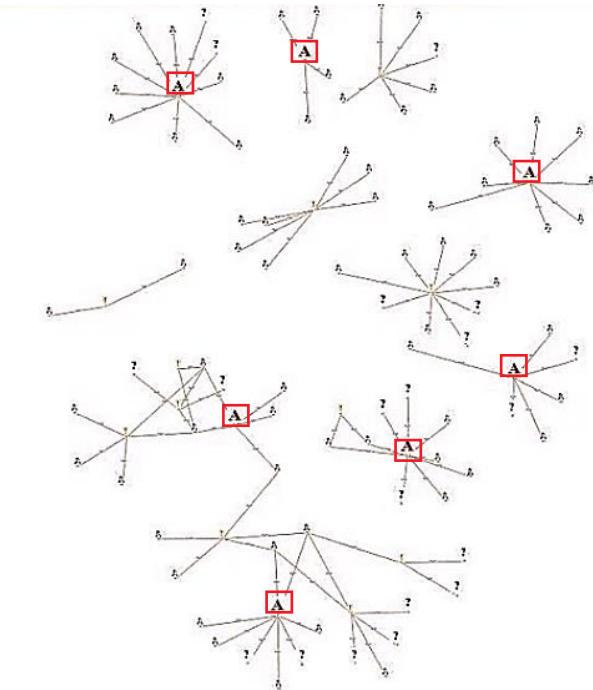
- Os cinco V's:
 - Volume
 - Variedade
 - Velocidade
 - Veracidade
 - Valor



Trade off do big data

- Problema N²
- Quantidade de dados X Recursos computacionais
- Como processar bilhões de registros em segundos?
- Como analisar multiplas fontes de dados e transformá-los em informação e conhecimento?

Exemplo: detecção de fraude



■ Acidentes com suspeitas de fraude

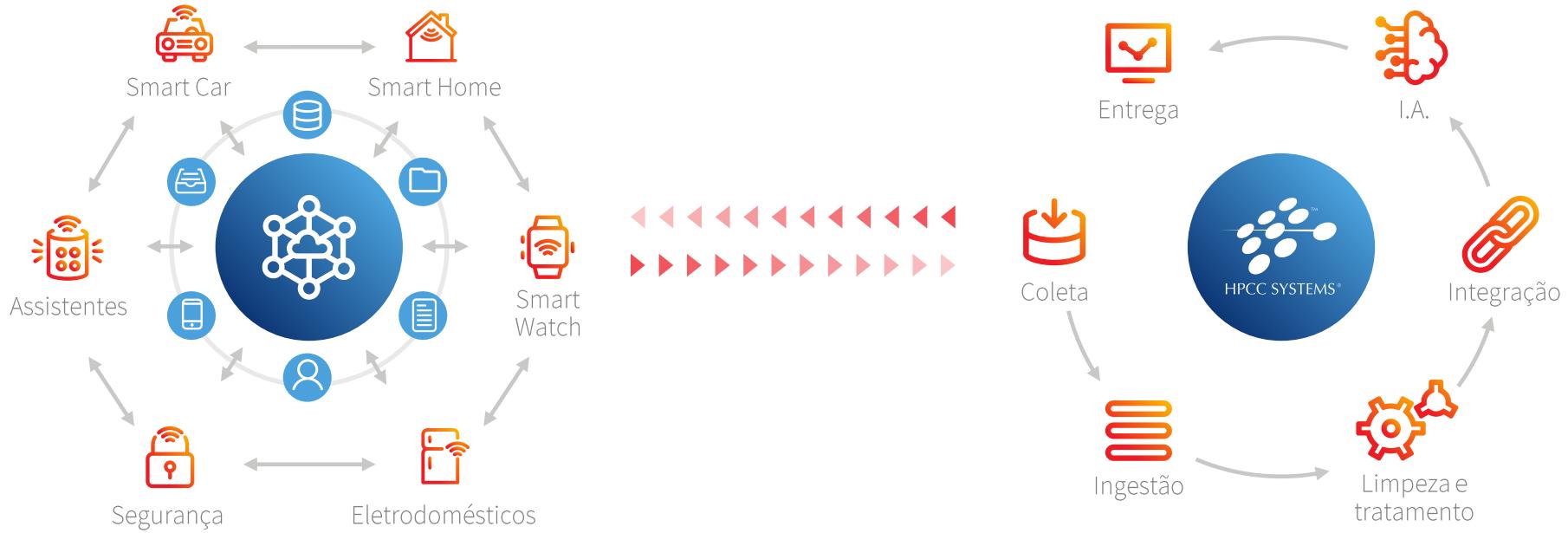
● Pessoas associadas aos acidentes



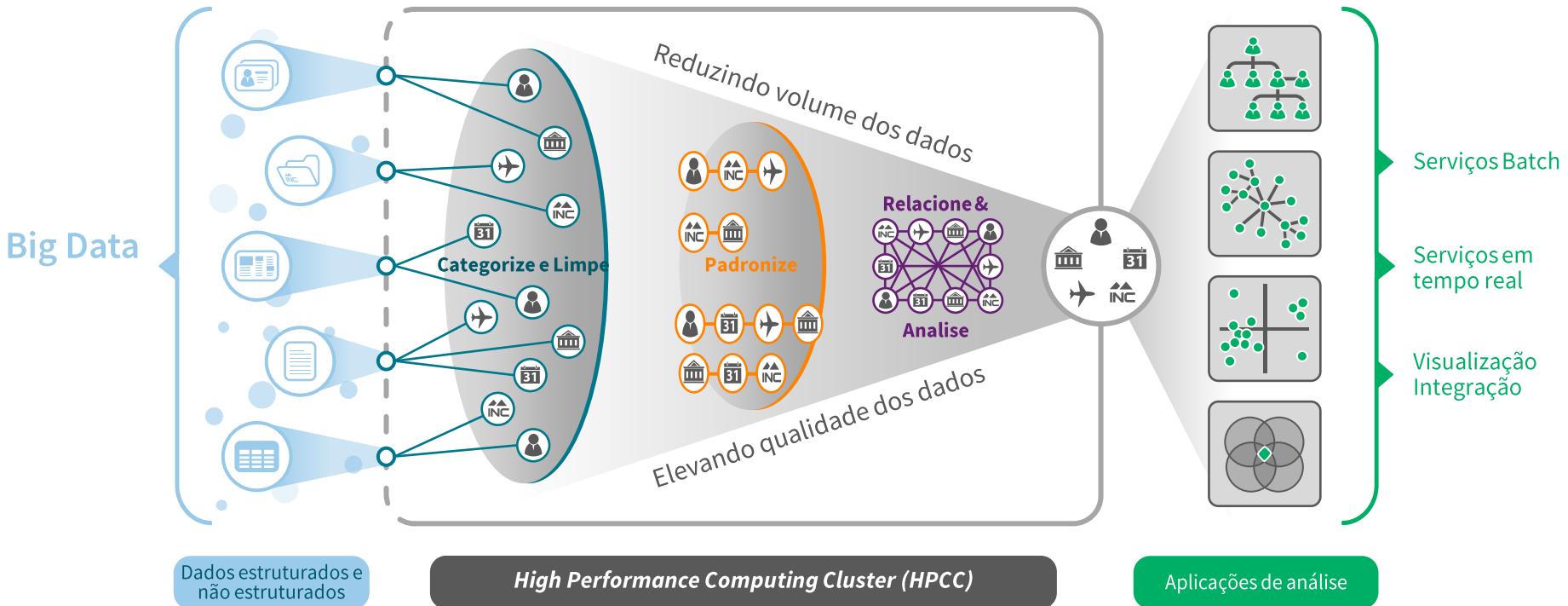
Um exemplo mais atual ...

<https://covid19.hpccsystems.com/>

O que é o HPCC Systems?



Pra que serve o HPCC Systems?



Visão geral da plataforma



Cluster Thor

Extração, transformação e carregamento de dados



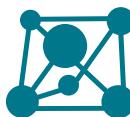
Cluster ROXIE

Entrega online de consultas em big data



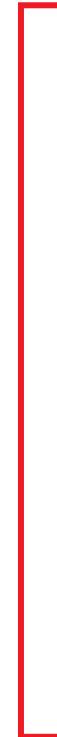
Ferramentas para manipulação de dados

Perfilamento, limpeza, consolidação de dados



Bibliotecas de Machine Learning

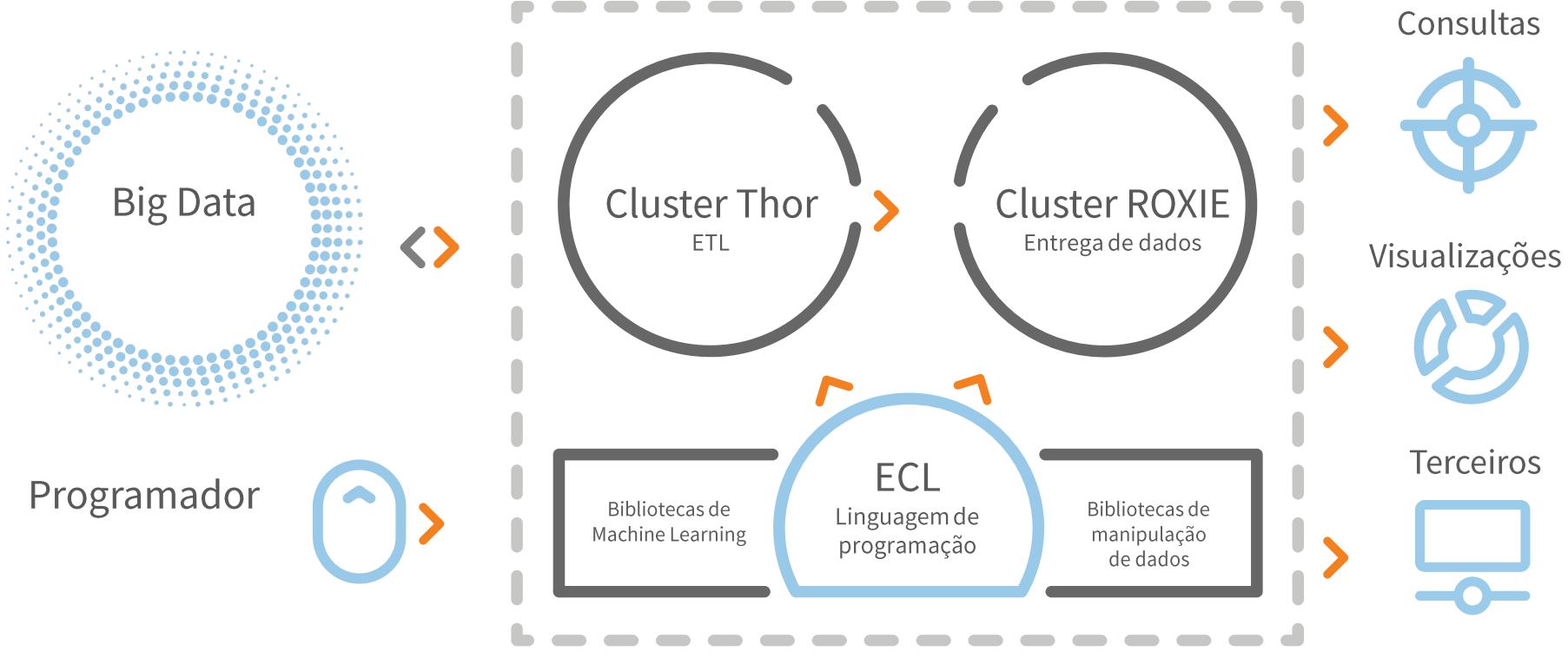
Supervisionado, não-supervisionado, aprendizagem profunda



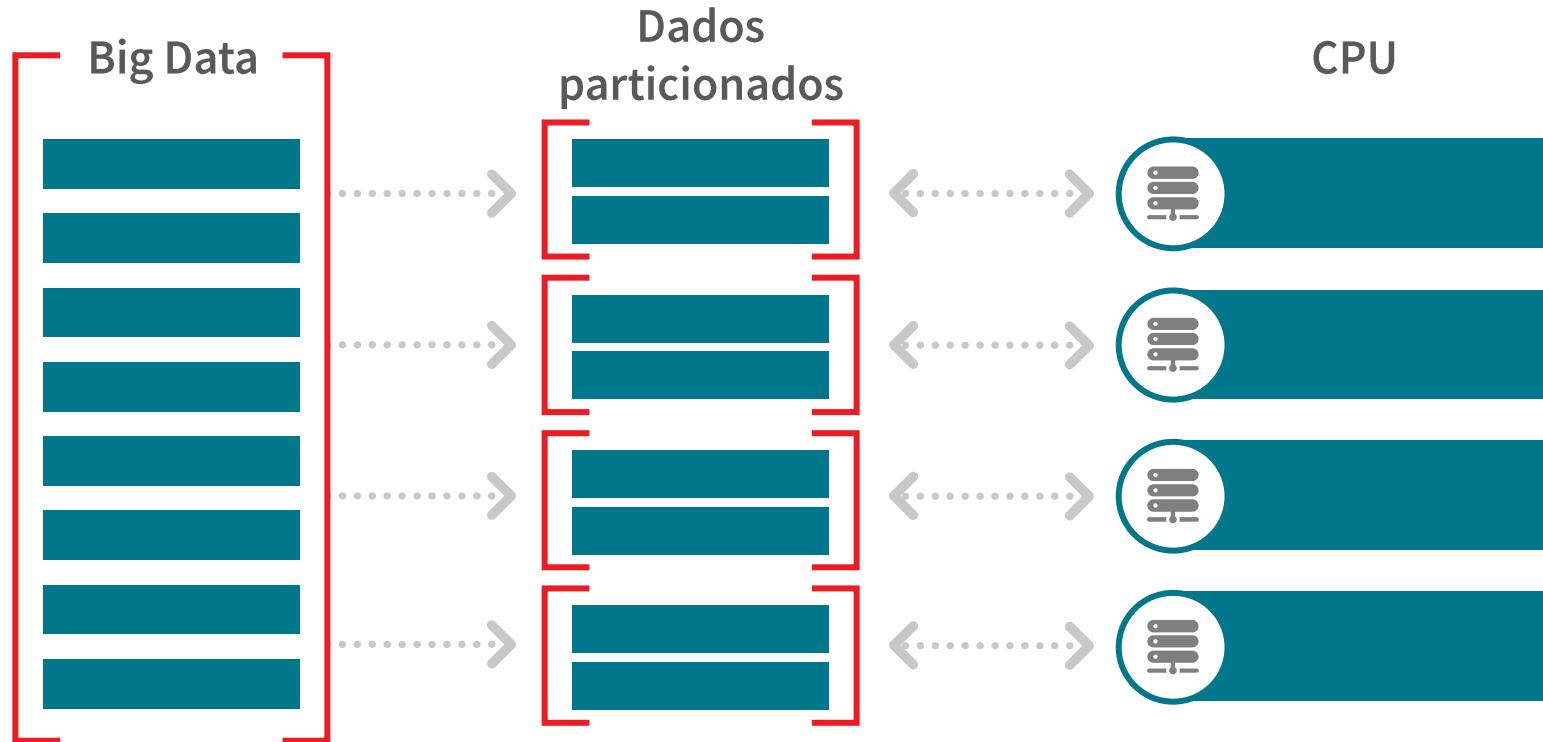
Conectividade

Plugins de integração com outros sistemas

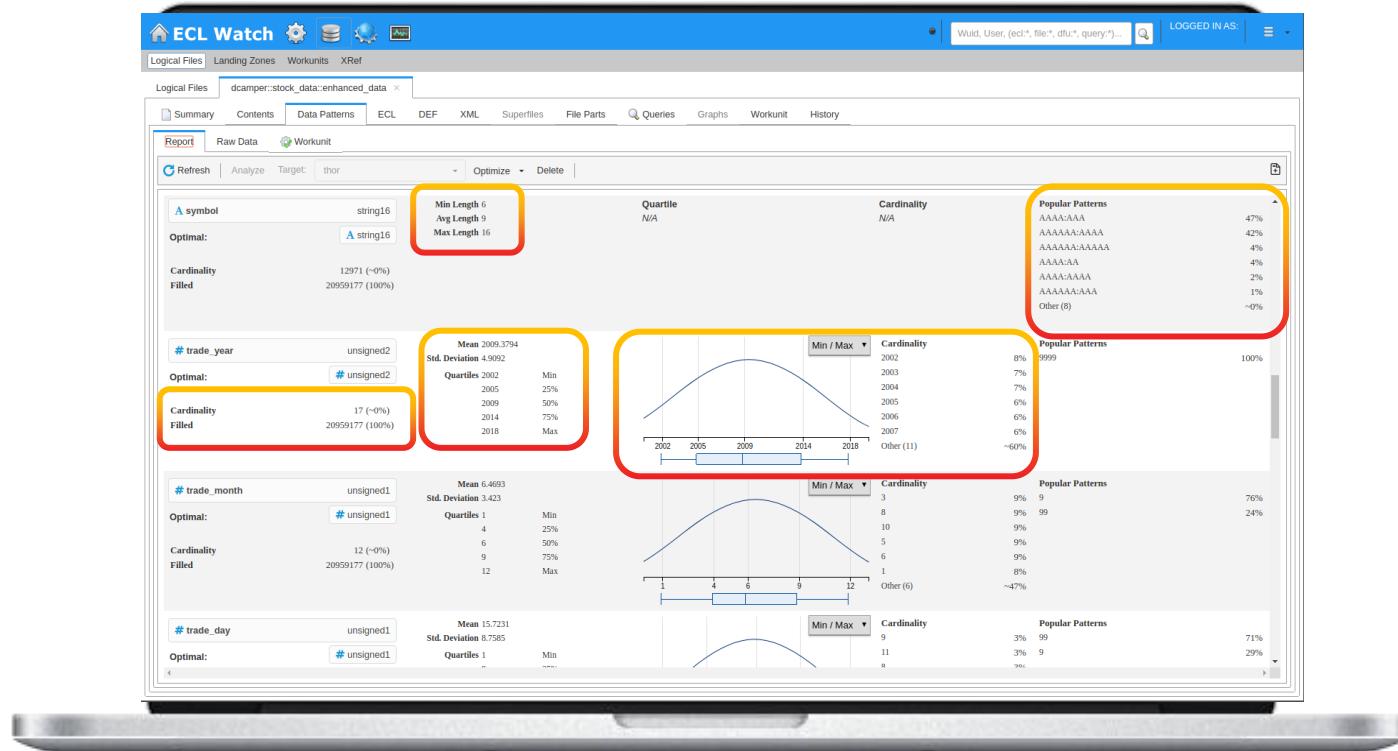
Arquitetura do HPCC Systems



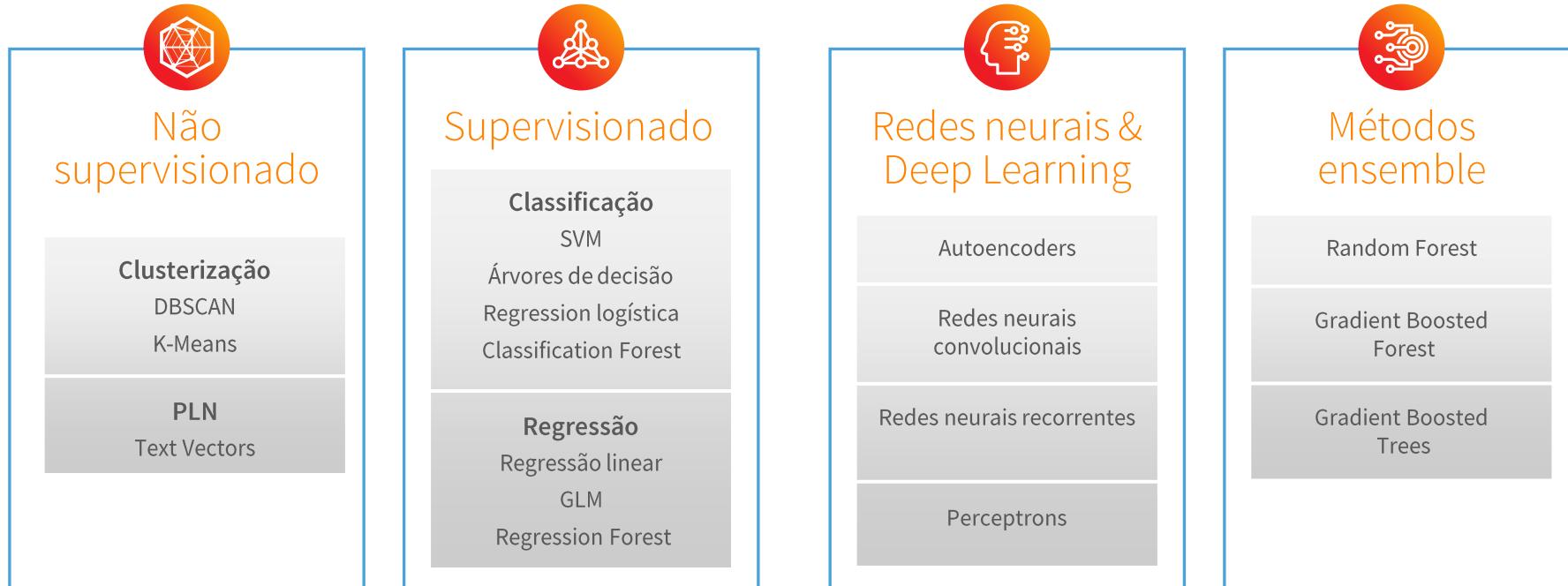
Dados distribuídos e processamento paralelo



Bibliotecas de profiling de dados



Bibliotecas de machine learning



Plugins para conectividade

HPCC

PENTAHO

Tableau

Java API

REDIS

SPARK

KAFKA

Couchbase

SQS

MEMCACHED

Linguagens suportadas

- C++
- R
- Python
- Java
- CQL
- SQL

CODE: SELECT ALL

```
IMPORT java;
STRING jcat(STRING a, STRING b) :=
  IMPORT(java,
    'JavaCat.cat:(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;' :
  classpath('/opt/HPCCSystems/classes'));

jcat('Hello ', 'world!');
```

CODE: SELECT ALL

```
IMPORT python;
SET OF STRING split(STRING text) := EMBED(python)
  return text.split()
ENDEMBED;
split('Once upon a time');
```

CODE: SELECT ALL

```
IMPORT python;
r := RECORD
  STRING word;
  UTF8 tags;
END;
DATASET(R) tag(STRING text) := IMPORT(python, './ex2.tag');
tag('Once upon a time there was a boy called Richard');
```

CODE: SELECT ALL

```
IMPORT MySQL;
stringrec := RECORD
  string name
END;
sqlrec := RECORD
  string ssn;
  string address;
END;
DATASET(sqlrec) MySQLJoin(dataset(stringrec) inrecs) := EMBED(mysql)
  SELECT * from tbl1 where name = ?;
ENDEMBED;
MySQLJoin(indata);
```



Tutorial: Machine Learning com HPCC

Objetivo do tutorial

Serviço web de consulta de preço de imóveis

roxie
fn_getprice_roxiequery_web.1 Dynamic Form ▾

FN_GETPRICE_ROXIEQUERY_WEB_1REQUEST

assess_val:	1188720
bedrooms:	3
full_baths:	2
half_baths:	1
land_sq_ft:	14774
living_sq_ft:	1437
year_acq:	2011
year_built:	1968
zip:	95451

Capture Log Info. Trace Level: No Timeout

Call Query ▾ Output Tables ▾ FORM POST ▾ Submit Clear All

fn getprice roxiequery web.1 Response

Dataset: Result 1

	preco
1	626353

(1.662.959 registros de propriedades)

#	propertyid	house_number	house_number_suffix	predir	street	streettype	postdir	apt	city	state	zip	total_value	assessed_value
1	828195	144			MCKIERNAN	DR			WALNUT CREEK	CA	94597	62614	62614
2	1144455	281			CENTER	ST			BALTIMORE	MD	21136	105500	10550
3	1494347	483			NEWTON	RD			FLAGSTAFF	AZ	86011	2220	2220
4	1910847	802			HATCHERY	CT			WOODLAND	WA	98674	356000	356000
5	4267562	5007		E	ROY ROGERS	RD			TROY	MI	48085	327253	327253
6	4888602	7607			PEBBLESTONE	DR	000009		KERNVILLE	CA	93238	732179	732179
7	54135	4			WAINWRIGHT	DR			NORTH FORT MYERS	FL	33917	159724	87848
8	762012	125			SHIPYARD	DR	000150		MELBOURNE VILLAGE	FL	32904	96300	96300
9	2331721	1190			LITTLEOAK	DR			HOUSTON	TX	77011	238854	217810
10	3276109	2506			MEADOW	DR			LA QUINTA	CA	92253	30977	30660

Integração com WEB

SIEEL 2021

Property value query

Assessed Value

Bedrooms

Full Baths

Half Baths

Land Square ft

Living Square ft

Year Acquired

Year Built

ZIP

Search

This page says

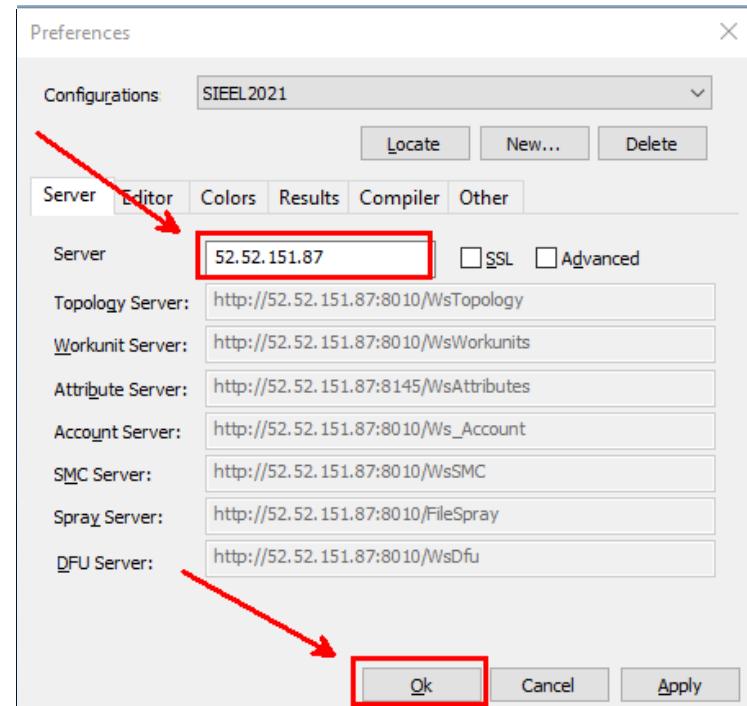
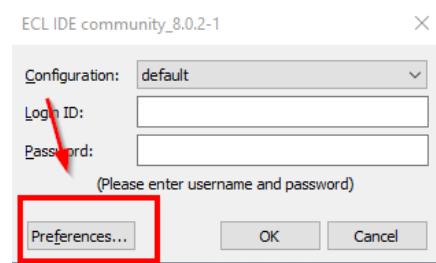
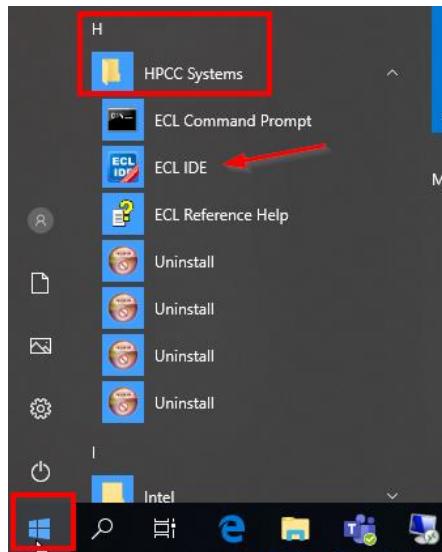
O preço estimado do imóvel é: 626353

OK

Preparação do ambiente

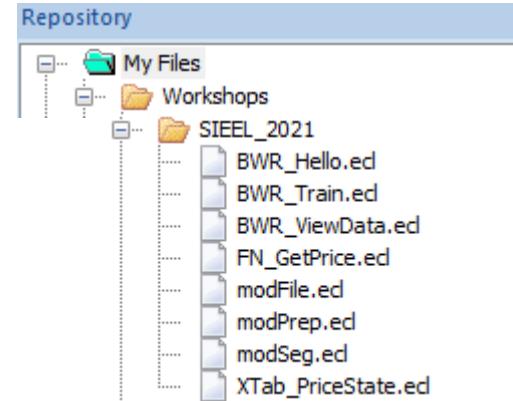
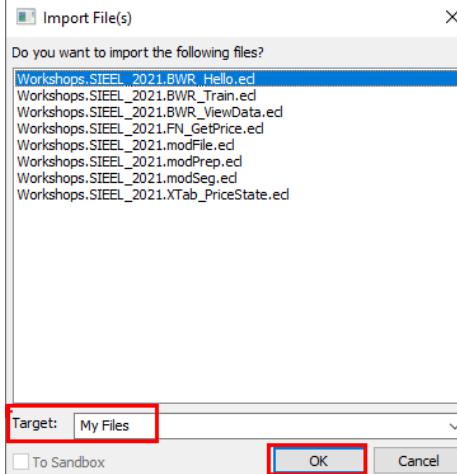
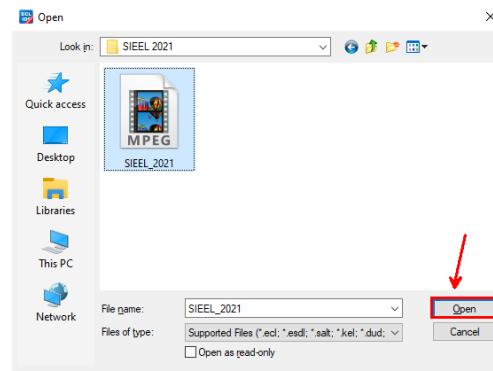
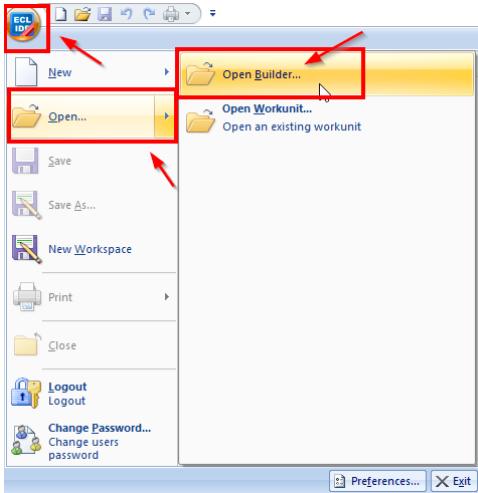
Cluster de treinamento: <http://52.52.151.87:8010/>

ECL IDE:

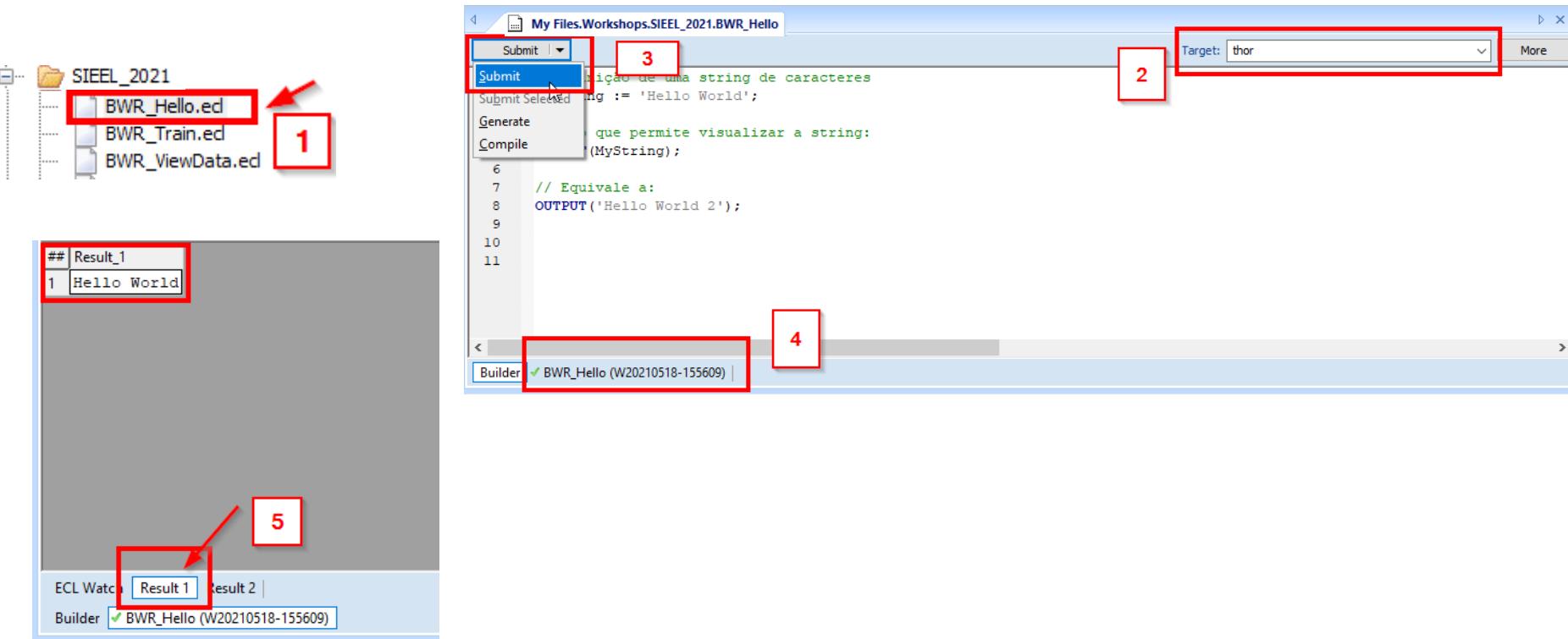


Preparação do ambiente (cont.)

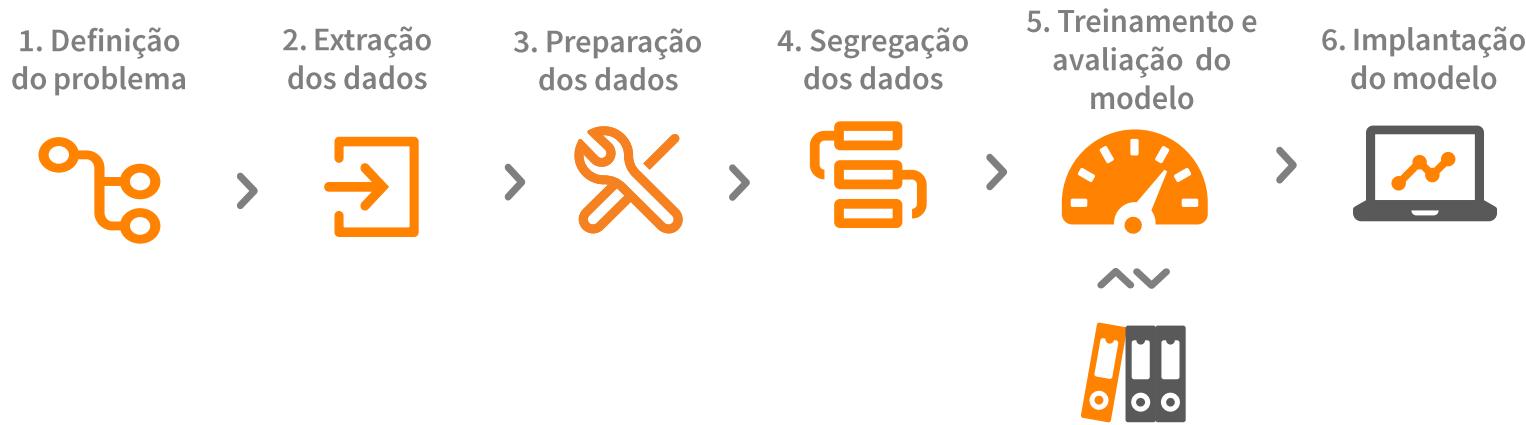
SIEEL_2021.mod



Teste do ambiente



Fluxo de aprendizado supervisionado

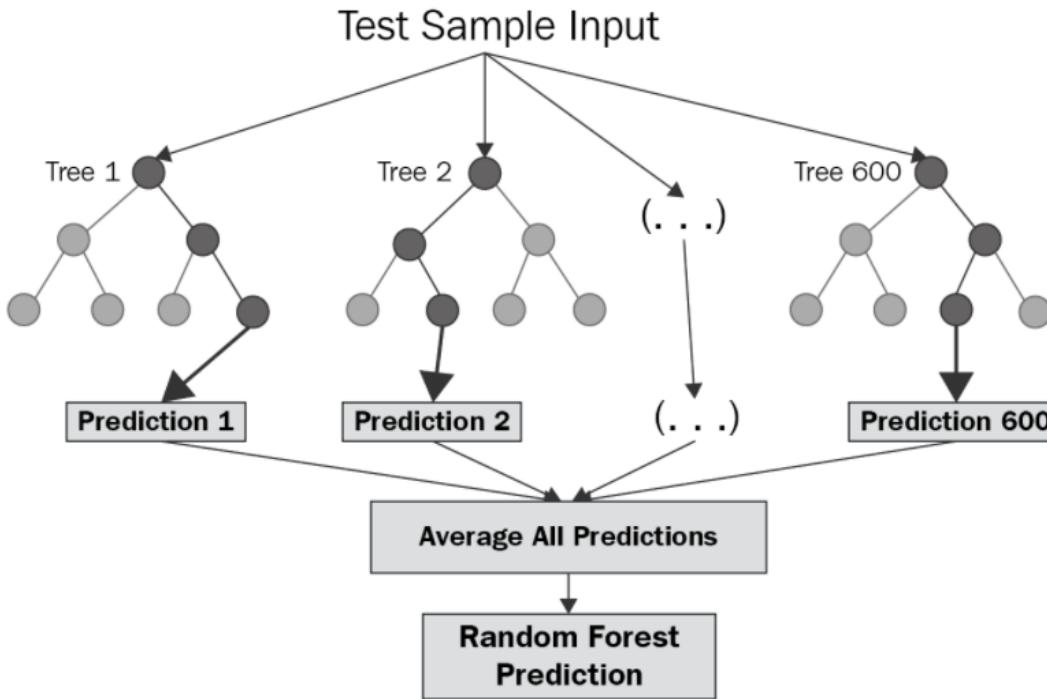


1. Definição do problema

“Dado um conjunto de atributos de uma propriedade (localização, metragem, ano de construção), como predizer o seu valor real de venda?”

propertyid	house_number	house_number_predir	street	street_type	postdir	apt	city	state	zip	total_value	assessed_value	year_acquired	land_square_foot	living_square_foot	bedrooms	full_baths
828195	144		MCKIERNAN	DR			WALNUT CREEK	CA	94597	62614	52614	2006	20418	2485	3	2
1144455	281		CENTER	ST			BALTIMORE	MD	21136	105500	10550	2007	4807	1368	0	0
1494347	483		NEWTON	RD			FLAGSTAFF	AZ	86011	2220	2220	0	5654	1011	3	1
1910847	802		HATCHERY	CT			WOODLAND	WA	98674	356000	356000	0	6094	0	2	1
4267562	5007	E	ROY ROGERS	RD			TROY	MI	48085	327253	327253	2007	3484	0	3	0
4888602	7607		PEBBLESTONE	DR		000009	KERNVILLE	CA	93238	732179	732179	2010	19597	6132	6	6
48725	4		LONG	AVE			SUNRISE	FL	33323	271000	271000	2008	6880	2392	4	2
83528	6		TRILLUM	LN			WAYLAND	MA	02193	79889	79889	2007	7657	1657	4	1
94604	7		PARMENTER	AVE			PLYMOUTH	MN	55441	23800	23800	2005	19994	1754	3	2
220326	17		TIMBER	RD			LOS ANGELES	CA	90063	89000	89000	2008	7840	954	3	1
994609	212		FREYER	DR	NE		PHILOMONT	VA	20131	59800	59800	2009	11199	1241	3	0
1836173	724		EASTER	ST			ALLENTOWN	PA	18102	191600	191600	0	9100	2534	4	2
2910797	1903		SADDLE BROOK	DR			CLIO	CA	96106	61610	61610	2007	0	0	0	0
3083959	2158		RIVERSIDE	DR			UPPER MOREL...	PA	19006	90300	0	0	0	1235	3	2
3952189	4040		GRAND VIEW	BLVD		000054	RIO LINDA	CA	95673	0	0	0	2700720	0	0	0
4186238	4726		LAS PALMAS	CT			WAELDER	TX	78959	18816	18816	2009	2159	1320	0	0
4597143	6213		WILSON	RD			ZOLFO SPRINGS	FL	33890	72600	0	0	8496	0	3	1
4624905	6321		STONEWALL	LN			PATERSON	NJ	07514	139880	139880	2008	10454	1391	4	2
92326	7		KNOLLCREST	DR			NARANJA	FL	33032	76214	76214	2008	4800	930	2	0
1792852	704		ERIN	DR			TRABUCO	CA	92678	28010	28010	2007	5200	0	3	1
1843977	728	S	ARLINGTON H...	RD			BLOOMING GRO...	TX	76626	130400	130400	2007	36154	1629	3	1
4214R72	4R21		MURTI F OAK	DR		0000075	SAN BERNARD	CA	02376	22250	22250	2007	03654	0	0	0

1. Definição do problema (cont.)



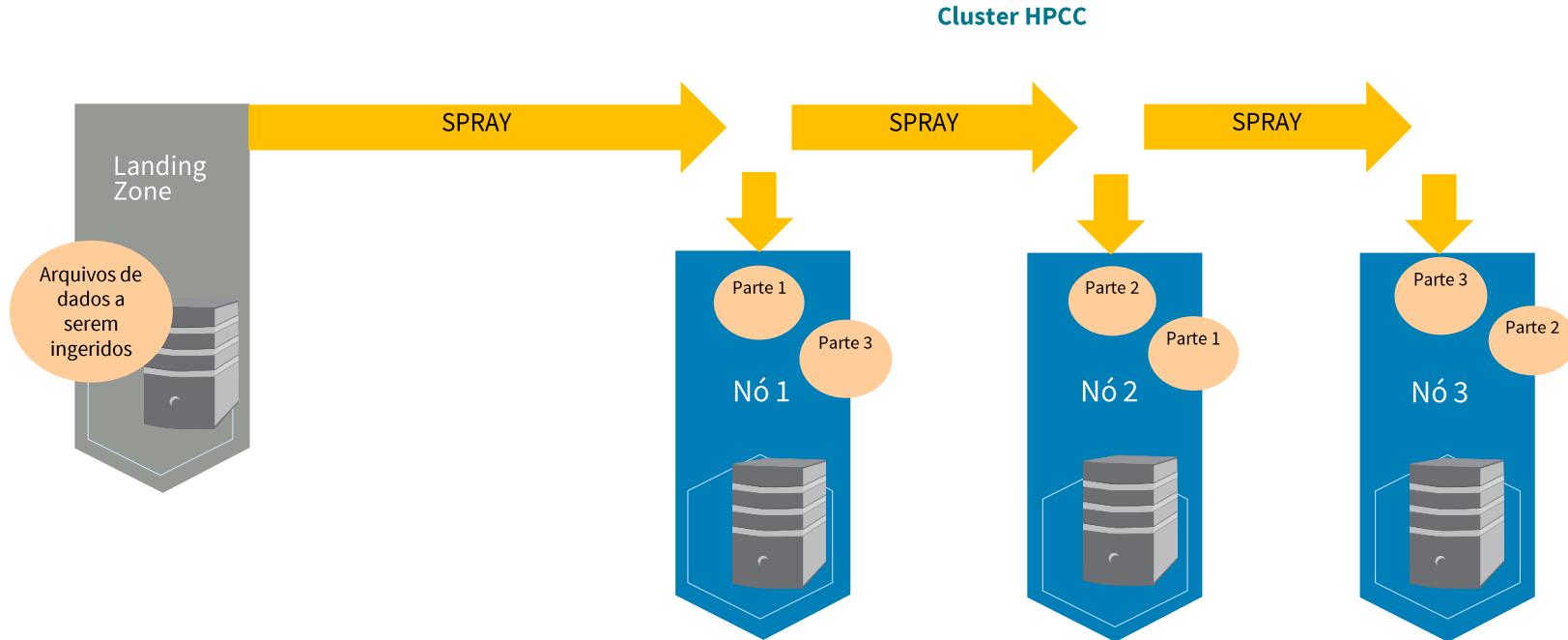
2. Extração dos dados

“Importação e análise de dados brutos provenientes de diferentes fontes”

##	personid	propertyid	house_number	house_number_suffix	predir	street	streettype	postdir	apt	city	state	zip	total_value
1	187522928604396	828195	144			MCKIERNAN	DR			WALNUT CREEK	CA	94597	62614
2	187522928604396	1144455	281			CENTER	ST			BALTIMORE	MD	21136	105500
3	187522928604396	1494347	483			NEWTON	RD			FLAGSTAFF	AZ	86011	2220
4	187522928604396	1910847	802			HATCHERY	CT			WOODLAND	WA	98674	356000
5	187522928604396	4267562	5007		E	ROY ROGERS	RD			TROY	MI	48085	327253
6	187522928604396	4888602	7607			PEBBLESTONE	DR		000009	KERNVILLE	CA	93238	732179
7	1258313199446079	48725	4			LONG	AVE			SUNRISE	FL	33323	271000
8	1258313199446079	83528	6			TRILLUM	LN			WAYLAND	MA	02193	79889
9	1258313199446079	94604	7			PARMENTER	AVE			PLYMOUTH	MN	55441	23800
10	1258313199446079	220326	17			TIMBER	RD			LOS ANGELES	CA	90063	89000
11	1258313199446079	994609	212			FREYER	DR	NE		PHILOMONT	VA	20131	59800
12	1258313199446079	1836173	724			EASTER	ST			ALLENTOWN	PA	18102	191600
13	1258313199446079	2910797	1903			SADDLE BROOK	DR			CLIO	CA	96106	61610
14	1258313199446079	3083959	2158			RIVERSIDE	DR			UPPER MORELAND	PA	19006	90300
15	1258313199446079	3952189	4040			GRAND VIEW	BLVD		000054	RIO LINDA	CA	95673	0

##	Result_2
1	1662959

2. Extração dos dados



As partes do arquivo são referenciadas em ECL como um único arquivo lógico...

2. Extração dos dados (cont.)

The screenshot shows the ECL Watch application interface. On the left, there is a tree view of logical files under 'Landing Zones'. A specific file named 'propriedades' is selected and highlighted with a red box and arrow (labeled 1). In the main panel, there are several tabs at the top: 'Delimited' (selected), 'XML', 'JSON', 'Variable', and 'BLOB'. Below these tabs, the 'Target Name' field contains the value 'propriedadesXXX' and is also highlighted with a red box and arrow (labeled 3). On the right side, there is an 'Options' panel with various configuration settings. One setting, 'Replicate: No Common:' with the checkbox checked, is highlighted with a red box and arrow (labeled 4). At the bottom right of the main panel, there is a 'Spray' button, which is also highlighted with a red box and arrow (labeled 2).

<http://52.52.151.87:8010/>
(ECL Watch)

2. Extração dos dados (cont.)

ECL Watch interface showing the extraction of data from logical files.

The top navigation bar includes icons for Home, Logical Files, Landing Zones, Workunits, and XRef. The "Logical Files" tab is selected.

The main content area displays a table of logical files. One row is highlighted with a red box and contains the logical name "...propriedadesxxx". An arrow points to this row.

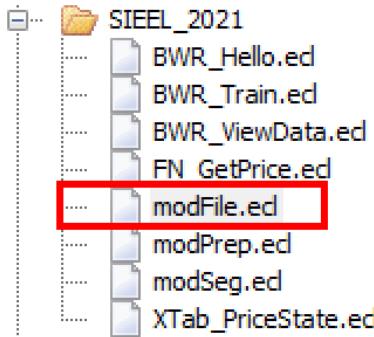
The bottom section shows a detailed view of the "...propriedadesxxx" file. The "Data Patterns" tab is selected, and the "Analyze" button is highlighted with a red box.

The analysis results for the "field1" column include:

- Optimal:** # unsigned8
- Cardinality Filled:** 279256 (~17%) / 1662959 (100%)
- Mean:** 9215555893612636000
- Std. Deviation:** 5324435205711618000
- Quartiles:** 187522928604396 (Min), 4609083185180437000 (25%), 9212310153083255000 (50%), 13816564114144750000 (75%), 18446714708963650000 (Max)
- Cardinality:** N/A
- Popular Patterns:** 999999999999999999 (49%), 999999999999999999 (46%), 999999999999999999 (5%), 999999999999999999 (0%), 999999999999999999 (0%), 999999999999999999 (0%)

A histogram plot shows the distribution of values for "field1". The x-axis ranges from 1.88e+14 to 1.84e+19. The y-axis represents frequency. A blue line represents the distribution, and a blue box highlights a specific bin.

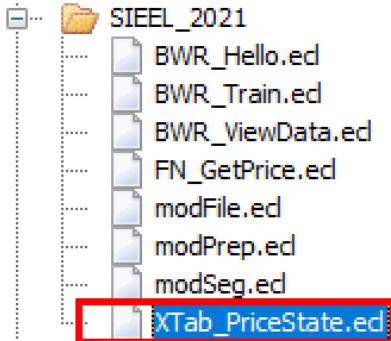
2. Extração dos dados (cont.)



personid	propertyid	house_number	house_nui	predir	street	streettype
18752292...	828195	144			MCKIERNAN	DR
18752292...	1144455	281			CENTER	ST
18752292...	1494347	483			NEWTON	RD
18752292...	1910847	802			HATCHERY	CT
18752292...	4267562	5007	E		ROY ROGERS	RD
18752292...	4888602	7607			PEBBLESTONE	DR
12583131...	48725	4			LONG	AVE
12583131...	83528	6			TRILLUM	LN
12583131...	94604	7			PARMENTER	AVE
12583131...	220326	17			TIMBER	RD

```
EXPORT modFile := MODULE
  EXPORT Layout := RECORD
    UNSIGNED8 personid;
    UNSIGNED4 propertyid;
    UNSIGNED2 house_number;
    STRING8 house_number_suffix;
    STRING2 predir;
    STRING29 street;
    STRING5 streettype;
    STRING2 postdir;
    STRING6 apt;
    STRING27 city;
    STRING2 state;
    STRING5 zip;
    UNSIGNED4 total_value;
    UNSIGNED4 assessed_value;
    UNSIGNED3 year_acquired;
    UNSIGNED4 land_square_footage;
    UNSIGNED3 living_square_feet;
    UNSIGNED2 bedrooms;
    UNSIGNED2 full_baths;
    UNSIGNED2 half_baths;
    UNSIGNED3 year_built;
  END;
  EXPORT File := DATASET('~propriedadesXXX',Layout,CSV);
```

Bônus: Visualize os dados brutos

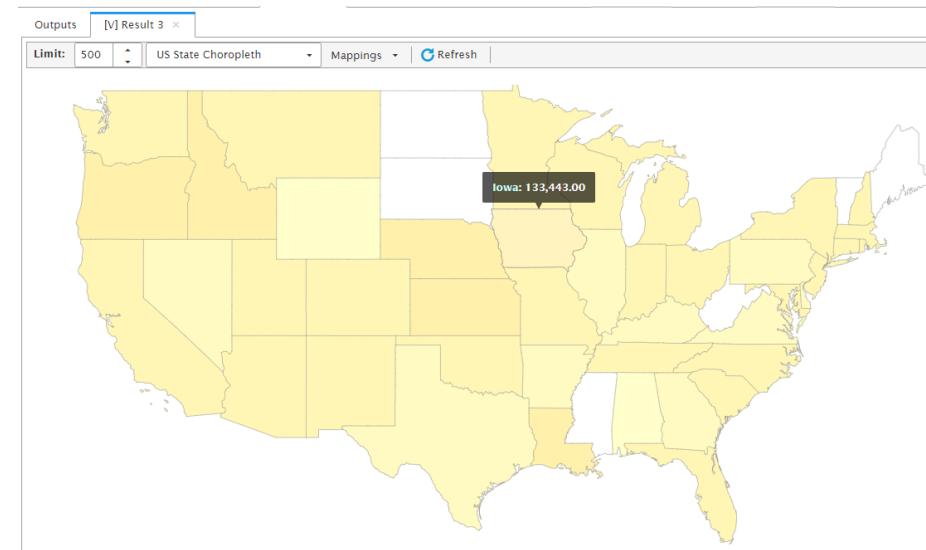


```
IMPORT $;

Property := $.modFile.File;

OutRec := RECORD
    Property.state;
    UNSIGNED4 avg_value := AVE(GROUP,Property.total_value);
END;

EXPORT XTAB_PriceState := TABLE(Property,OutRec,state);
```



3. Preparação dos dados

“Limpeza, padronização e consolidação de registros ”

##	propertyid	zip	assessed_value	year_acquired	land_square_footage	living_square_feet	bedrooms	full_baths	half_baths	year_built	total_value
1	79784	33424	76440	2015	4299	1255	3	2	0	2010	76440
2	3924129	20601	95900	2013	11224	1468	3	2	1	2007	95900
3	413843	8803	76000	2015	57000	1858	3	2	0	1970	76000
4	608224	98370	39340	2012	7405	1066	3	1	1	1967	39340
5	942963	72032	278400	2008	9600	2459	3	2	0	1963	278400
6	2237271	79935	143600	2011	8430	1008	2	1	1	1961	143600
7	4443742	84065	166934	2013	9317	1700	4	2	0	1991	166934
8	3834707	66227	348350	2012	15300	2663	4	2	1	2002	348350
9	3592739	19606	54000	2015	15060	2292	4	2	1	1980	90000
10	2916349	34639	119050	2015	6947	1709	3	2	0	2009	140950

3. Preparação dos dados



The screenshot shows a file explorer window with a tree view on the left and a code editor on the right. The tree view lists several files under a folder named 'SIEEL_2021'. The file 'modPrep.ed' is highlighted with a red border.

```
IMPORT $;
Property := $.modFile.File;

EXPORT modPrep := MODULE
    // Limpando os dados
    CleanFilter := Property.zip <> '' AND Property.assessed_value <> 0 AND Property.year_acquired <> 0 AND
        Property.land_square_footage <> 0 AND Property.living_square_feet <> 0 AND
        Property.bedrooms <> 0 AND Property.full_baths <> 0 AND Property.year_Built <> 0;

    EXPORT CleanProperty := Property(CleanFilter);

    EXPORT STD_Layout := RECORD
        UNSIGNED8 PropertyID;
        UNSIGNED3 zip;
        UNSIGNED4 assessed_value;
        UNSIGNED4 year_acquired;
        UNSIGNED4 land_square_footage;
        UNSIGNED4 living_square_feet;
        UNSIGNED2 bedrooms;
        UNSIGNED2 full_baths;
        UNSIGNED2 half_baths;
        UNSIGNED2 year_built;
        UNSIGNED4 total_value;                                // variável dependente - a ser determinada
        UNSIGNED4 rnd;                                       // número aleatório
    END;

    EXPORT myDataP := PROJECT(CleanProperty, TRANSFORM(STD_Layout,
        SELF.rnd := RANDOM(),
        SELF.Zip := (UNSIGNED3)LEFT.Zip,
        SELF := LEFT))
        :PERSIST('~/SIEEL::XXX::PrepProp');

    // Aleatorize os dados ordenando o campo com número aleatório
    EXPORT myDataPS := SORT(myDataP, rnd);
    EXPORT myDataPrep := PROJECT(myDataPS,STD_Layout and NOT rnd);

END;
```

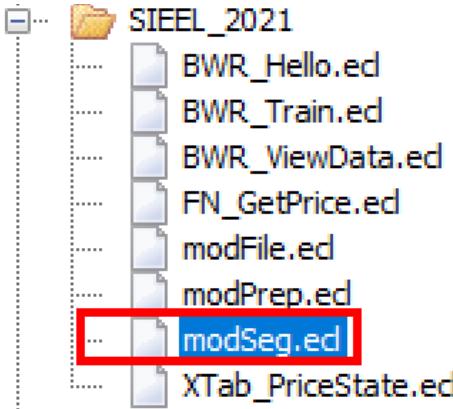
4. Segregação dos dados

“Selecionar aleatoriamente amostras de treinamento e validação com distinção de variáveis dependentes e independentes”

##	wi	id	number	value
1	1	79784	1	76440.0
2	1	3924129	1	95900.0
3	1	413843	1	76000.0
4	1	608224	1	39340.0
5	1	942963	1	278400.0
6	1	2237271	1	143600.0
7	1	4443742	1	166934.0
8	1	3834707	1	348350.0
9	1	3592739	1	90000.0
10	1	2916349	1	140950.0

##	wi	id	number	value
1	1	79784	1	33424.0
2	1	79784	2	76440.0
3	1	79784	3	2015.0
4	1	79784	4	4299.0
5	1	79784	5	1255.0
6	1	79784	6	3.0
7	1	79784	7	2.0
8	1	79784	8	0.0
9	1	79784	9	2010.0
10	1	3924129	1	20601.0

4. Segregação dos dados



```
IMPORT $,ML_Core;

// Considere os primeiros 5000 registros como amostra de treinamento
myTrainData := $.modPrep.myDataPrep[1..5000];

// Considere os 2000 registros seguintes como amostra de teste
myTestData := $.modPrep.myDataPrep[5001..7000];

// Conversão matricial dos campos numéricos
ML_Core.ToField(myTrainData, myTrainDataNF);
ML_Core.ToField(myTestData, myTestDataNF);
// OUTPUT(myTrainDataNF);
// OUTPUT(myTestDataNF);

EXPORT modSeg := MODULE;

EXPORT myIndTrainDataNF := myTrainDataNF(number < 10);

EXPORT myDepTrainDataNF := PROJECT(myTrainDataNF(number = 10),
                                     TRANSFORM(RECORDOF(LEFT),
                                                SELF.number := 1,
                                                SELF := LEFT));

EXPORT myIndTestDataNF := myTestDataNF(number < 10);

EXPORT myDepTestDataNF := PROJECT(myTestDataNF(number = 10),
                                     TRANSFORM(RECORDOF(LEFT),
                                                SELF.number := 1,
                                                SELF := LEFT));

END;
```

5. Treinamento e avaliação do modelo

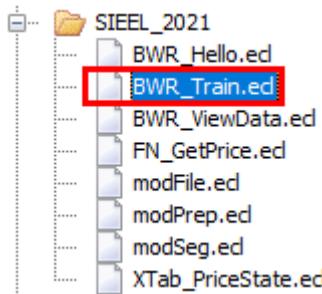
“Obtenção de modelo a partir da amostra de treinamento e validação na amostra de teste”

	wi	value	indexes	fileposition
			Item	
1	0	4356.0	3 10 1	0
2	0	2812.0	3 10 2	27
3	0	2476.0	3 10 3	54
4	0	1244.0	3 10 4	81
5	0	1082.0	3 10 5	108
6	0	4085.0	3 10 6	135

##	wi	id	number	value
1	1	3634	1	59055.31318837311
2	1	5840	1	126151.3283316611
3	1	12721	1	150876.4676173128
4	1	47045	1	233897.4086392291
5	1	91757	1	111950.2604939628
6	1	117238	1	81157.13156934927
7	1	149746	1	75868.58107175257
8	1	239046	1	39961.17077444747
9	1	246517	1	128203.9088547347
10	1	252615	1	69009.47259550788

##	wi	regressor	r2	mse	rmse
1	1	1	0.7304899830671003	7982069594.129144	89342.4288573416

5. Treinamento e avaliação do modelo



```
IMPORT $;
IMPORT ML_Core;
IMPORT LearningTrees AS LT;

// Selecione o algoritmo
myLearnerR      := LT.RegressionForest(10,,10,[1]);

// Obtenha o modelo treinado
myModelR        := myLearnerR.GetModel($.modSeg.myIndTrainDataNF,$.modSeg.myDepTrainDataNF);
OUTPUT(myModelR,'~SIEEL::XXX::mymodelR',NAMED('ModeloTreinado'),overwrite);

// Teste o modelo
predictedDeps := myLearnerR.Predict(myModelR, $.modSeg.myIndTestDataNF);
OUTPUT(predictedDeps,NAMED('ValoresPrevistos'));

// Avalie o modelo
assessmentR    := ML_Core.Analysis.Regression.Accuracy(predictedDeps,$.modSeg.myDepTestDataNF)
OUTPUT(assessmentR,NAMED('AvaliacaodoModelo'));
```

6. Implantação do modelo

“Carregamento de dados e disponibilização de consulta web”

roxie

fn_getprice_aros Dynamic Form ▾

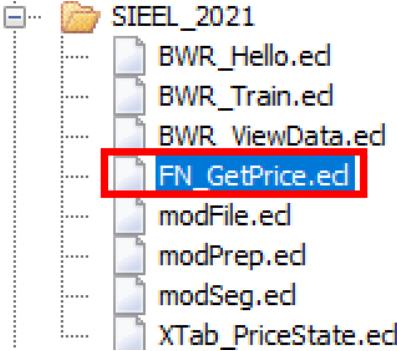
FN_GETPRICE_AROREQUEST

assess_val:	<input type="text"/>
bedrooms:	<input type="text"/>
full_baths:	<input type="text"/>
half_baths:	<input type="text"/>
land_sq_ft:	<input type="text"/>
living_sq_ft:	<input type="text"/>
year_acq:	<input type="text"/>
year_built:	<input type="text"/>
zip:	<input type="text"/>

Capture Log Info. Trace Level: No Timeout

Call Query ▾ Output Tables ▾ FORM POST ▾ Submit Clear All

6. Implantação do modelo



```
IMPORT $;
IMPORT ML_Core;
IMPORT LearningTrees as LT;

EXPORT FN_GetPrice(Zip, Assess_val, Year_acq,
                    Land_sq_ft, Living_sq_ft, Bedrooms,
                    Full_baths, Half_baths, Year_built) := FUNCTION

    myInSet := [zip, assess_val, year_acq, land_sq_ft, living_sq_ft,
                bedrooms, full_baths, half_baths, year_built];

    myInDs := DATASET(myInSet, {REAL8 myInValue});

    ML_Core.Types.NumericField PrepData(RECORDOF(myInDs) Le, INTEGER C) := TRANSFORM
        SELF.wi           := 1,
        SELF.id          := 1,
        SELF.number      := C,
        SELF.value       := Le.myInValue;
    END;

    myIndepData := PROJECT(myInDs, PrepData(LEFT,COUNTER));

    mymodel := DATASET('~mymodelXXX',ML_Core.Types.Layout_Model2,FLAT,PRELOAD);

    myLearner := LT.RegressionForest(10,,10,[1]);

    myPredictDeps := MyLearner.Predict(mymodel, myIndepData);

    RETURN OUTPUT(myPredictDeps,{preco:=ROUND(value)});

END;

END;
```

6. Implantação do modelo

The screenshot shows the HPCC Systems IDE interface during the deployment of a model named 'FN_GetPrice.edl'. The left sidebar lists files in the 'SIEEL_2021' folder, with 'FN_GetPrice.edl' selected. The main window displays the ECL code for the function. A context menu is open over the code, with 'Compile' highlighted. A red box labeled '1' points to the 'Target: roxie' dropdown at the top right of the code editor. A red box labeled '2' points to the 'Compile' option in the context menu. Below the code editor is a job submission dialog with 'Job Name: FN_GetPrice_XXX' and a 'Submit' button. The bottom section shows the job details for 'W20211006-160253' and a log table.

My Files.Workshops.SIEEL_2021.FN_GetPrice

Target: roxie

1

2

```
Core.Types.NumericField PrepData(RECORDOF(myInDS) Le, INTEGER C) := TRANSFORM
  SELF.wi      := 1,
  SELF.id      := 1,
  SELF.number   := C,
  SELF.value    := Le.myInValue;
END;

myIndepData := PROJECT(myInDs, PrepData(LEFT,COUNTER));

mymodel := DATASET ('~mymodelXXX',ML_Core.Types.Layout_Model2,FLAT,PRELOAD);

MULTICARRIER := T.T_RegressionForest(10..10,1111..
```

Workflow: W20211006-160253

Action: compile

State: compiled

Owner: OlivAl01

Job Name: FN_GetPrice_ARO

Description: +

Protected:

Cluster: roxie

Total Cluster Time: 0.000

Aborted by:

Aborted time:

Services:

Variables (17) Outputs (1) Inputs Timers (10) Graphs (1)

Refresh Save Delete Restore Reschedule Deschedule Set To Failed Abort Recover Resubmit Clone Publish Z.A.P Slave Logs

Job Name: FN_GetPrice_XXX

Remote Dali:

Source Process:

Comment:

Priority: None

Allow Foreign Files:

Update Super Files:

Submit

Clear Copy Download

Severity	Source	Code	Message
Warning	eclcc	4531	JOIN condition folded to constant, converting to an ALL join

Serviço disponível para uso!

ECL Watch

[Queries](#) [Package Maps](#)

[Queries](#)

Refresh | Open | Delete | Suspend | Unsuspend | Activate | Deactivate | Filter | Option

	Priority	Name
		fn_getprice_xxx.1

fn_getprice_xxx.1

[fn_getprice_xxx.1](#)

[Summary](#) [Errors/Status \(1\)](#) [Logical Files \(1\)](#) [Super Files](#) [Libraries Used \(0\)](#) [Graphs \(1\)](#) [Resources](#) [Test Pages](#) [W20211007-203343](#)

[SOAP](#) [JSON](#) [WSDL](#) [Request Schema](#) [Response Schema](#) [Sample Request](#) [Sample Response](#) [Parameter XML](#) [Legacy Form](#) [Links](#)

Reset

roxie
fn_getprice_xxx.1 Dynamic Form

FN_GETPRICE_XXX_1REQUEST

assess_val:	1188720
bedrooms:	3
full_baths:	2
half_baths:	1
land_sq_ft:	14774
living_sq_ft:	1437
year_acq:	2001
year_built:	1968
zip:	95451

Capture Log Info. Trace Level: No Timeout

[Call Query](#) [Output Tables](#) [FORM POST](#) [Submit](#) [Clear All](#)

fn_getprice_xxx.1 Response

Dataset: Result 1

preco
1 722902



Próximos passos

Cursos online: +170 aulas (learn.lexisnexis.com/hpcc)

Introdução ao ECL (parte 1)

- Conceitos e consultas

Introdução ao ECL (parte 2)

- ETL com ECL

ECL Avançado (parte 1)

- Dados relacionais

ECL Avançado (parte 2)

- Superarquivos, XML/JSON e PLN

ECL Aplicado

- Geração e automação de código ECL

ROXIE ECL (parte 1)

- Índices e consultas

ROXIE ECL (parte 2)

- Otimização de consultas

Machine Learning com HPCC Systems

- Fundamentos para uso dos plugins

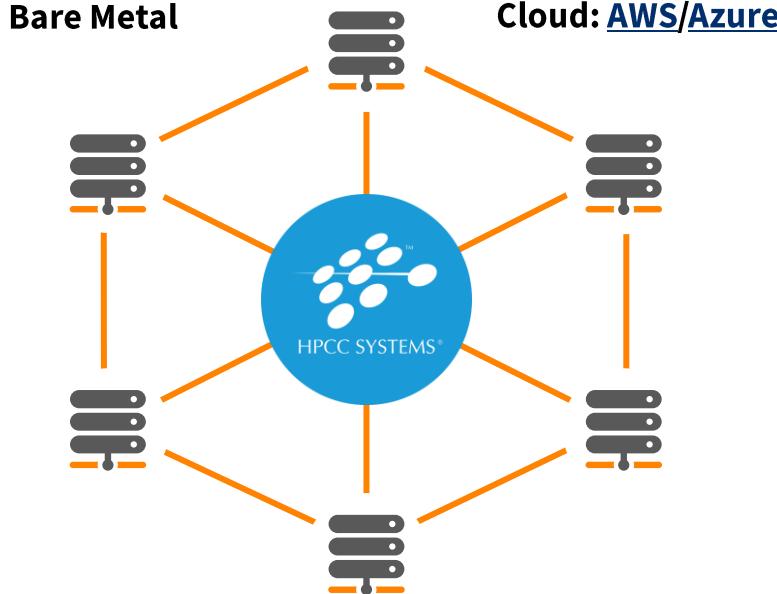
Administração de Sistemas

- Conceitos e operação básica

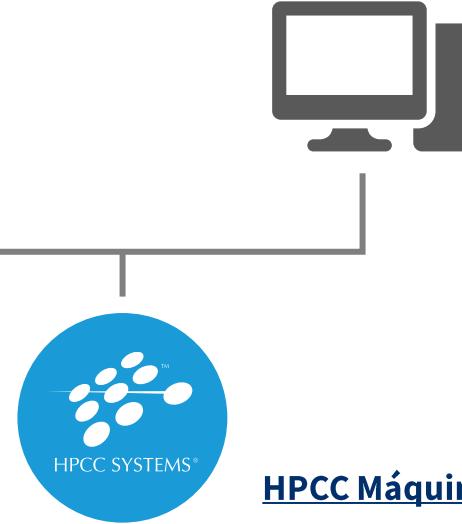
HPCC para gestores

- Visão geral e aplicações da plataforma

Opções de uso: play.hpccsystems.com



Oracle Virtual Box
HyperV
[Docker](#)



[HPCC Máquina Virtual](#)

Quem somos nós?



RELX é um provedor global de análises baseadas em informações e ferramentas de decisão para clientes profissionais e empresariais. O Grupo atende clientes em mais de 180 países e possui escritórios em cerca de 40 países.

Saiba mais em www.relx.com



Científico



Eventos



Análise de risco



Legal



Ativos e clientes



- 12 petabytes de dados públicos e privados
- 270 milhões de transações por hora
- Clientes em mais de **100** países
- **76%** de todas as empresas Fortune 500
- **7** dos 10 maiores bancos do mundo
- **100%** dos 50 maiores bancos americanos
- **95 das 100** maiores seguradoras
- **Mais de 7.500** orgãos governamentais locais, estaduais e federais

Estrutura no Brasil



Área de atuação

Análise de dados para organizações que buscam gerenciar riscos, encontrar oportunidades e melhorar seus resultados. Sediada em Atlanta, Geórgia, a LexisNexis Risk Solutions tem mais de 5.400 funcionários ao redor do mundo.

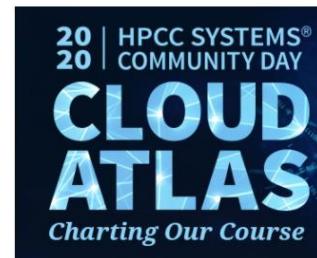
Tecnologia de código aberto

Plataforma de computação de Big Data de código aberto chamada HPCC Systems com vastos ativos de dados para proporcionar inteligência de decisão para clientes.

<https://github.com/hpcc-systems>

Relacionamento com Academia

<https://hpccsystems.com/community/academics>



Universidade de São Paulo
Brasil



UNIVERSIDADE FEDERAL
DE SANTA CATARINA



Universidades Brasileiras

Universidade de São Paulo
Brasil



- Disciplina Optativa na Poli/USP ([Link](#))
- Cursos de extensão ([Link](#))
- Co-orientação de IC's (PIBIC [Link1](#) [Link2](#) [Link3](#))



- Co-Orientação de TCC's ([Link](#) [Link2](#))
- Co-autoria de artigos científicos ([Link](#))
- Auxílio para aquisição de equipamentos

Projetos de Pesquisa



<https://wiki.hpccsystems.com/display/hpcc/Available+Projects>

Oportunidades profissionais

#ExploreMore

<https://risk.lexisnexis.com/about-us/careers>

<https://www.linkedin.com/company/lexisnexis-risk-solutions/>

<https://www.vagas.com.br/v2273659>

Ana Cristina Vieira

Senior Talent Acquisition

LexisNexis Risk Solutions Group

📱 +55.11.97075.5659

ana.vieira@lexisnexisrisk.com

Links úteis

- Site principal: hpccsystems.com
- Primeiros passos: hpccsystems.com/Why-HPCC-Systems
- Canal do youtube: youtube.com/user/HPCCSystems
- Fórum da Comunidade: hpccsystems.com/forums
- Poster Competition: [Link](#)



Faça parte da
Comunidade

Registre-se em hpccsystems.com

Considerações Finais



Contato

- Alysson.Oliveira@lexisnexisrisk.com
- Hugo.Watanuki@lexisnexisrisk.com



Backup

Enterprise Control Language (ECL)

Linguagem de programação centrada em dados (Data flow)

- Declarativa e não-procedural
- Códigos menores e reutilizáveis
- Biblioteca para manipulação de dados

Compilador

- Gera código otimizado (C++)
- Lógica para processamento paralelo e distribuído

Como fazer



vs.



O que fazer

Conceitos básicos de ECL

- Estrutura básica: **Nome := Expressão ;**
- ECL não é sensível a caixa alta/baixa
- Espaço em branco é ignorado para melhor leitura
- Comentários em linha (//) e em bloco (/* e */)
- ECL utiliza sintaxe objeto.propriedade

Dataset.Campo

// referencia um campo em um dataset

NomedoDiretorio.Definicao

// referencia uma definição em outro diretório

Tipos de dados primitivos

BOOLEAN

```
BOOLEAN IsFloridian := TRUE;
```

STRING[n]

```
STRING1 Gender := 'M';
```

INTEGER[n], UNSIGNED[n],

```
INTEGER1 ictr := -100;           // -128 to 127
```

```
UNSIGNED1 ctr := 0;             // 0 - 255
```

REAL[n], DECIMALn[_y]

```
REAL4 PI := 3.14159;
```

```
DECIMAL7_2 Salary := 75000.00;
```

Tipos de definição ECL

Booleana (boolean)

```
IsSeniorCitizen := People.birthdate>19600101;
```

Valor único (value)

```
MaleValue := 'M';
```

Conjunto de valores (set)

```
GenderValues := [ 'M', 'F' ];
```

People

##	firstname	lastname	middlename	namesuffix	filedate	bureaucode	maritalstatus	gender	dependentcount	birthdate	streetaddress
1	Cherianne	Khatchatourian	N		19990922	24		M	0		69 BOULDER RIDGE RD # 25
2	Muyesser	Raplee	X		20001111	353		F	0		55 SWAMP RD
3	Roselin	Viceconte			19990325	344		F	0	19800113	107 HILL TER
4	Inda	Provines			20000909	13		U	0		290 W MOUNT PLEASANT AVE
5	Inderdeep	Laurence	D		20001228	344		M	0		44 PROSPECT PL
6	Chrystine	Mangiapanne			19990827	315		F	0	19780306	1806 1ST AVE APT 8F
7	Adelene	Stock	R		20000827	252		M	0		1117 FARM RD
8	Mendy	Rufenblanchette			20000903	24		M	0		3 W 83RD ST APT 4C
9	Lannie	Amerantes	I		20001219	313		U	0		200 W 20TH ST APT 909
10	Tare	Gonyeau	T		19930807	48		F	0	19750801	6 CANDLE CT

Conjunto de registros (recordset)

```
SeniorPeople := People(IsSeniorCitizen);  
MalePeople := People(Gender=MaleValue);  
FemaleMalePeople := People(Gender IN GenderValues);
```

Ações vs. Definições

- ✓ O código ECL é constituído de:

- ✓ Definições: estabelecem o que as coisas são

MyString := 'Hello World'; // não inicia uma WU

- ✓ Ações: resultam em compilação e execução (arquivos BWR)

OUTPUT(MyString); // inicia uma WU