

# HPCC Systems

Processamento e análise de big data



# Bem-vindo! – Agenda do curso

## ✓ **HPCC Systems: Visão geral**

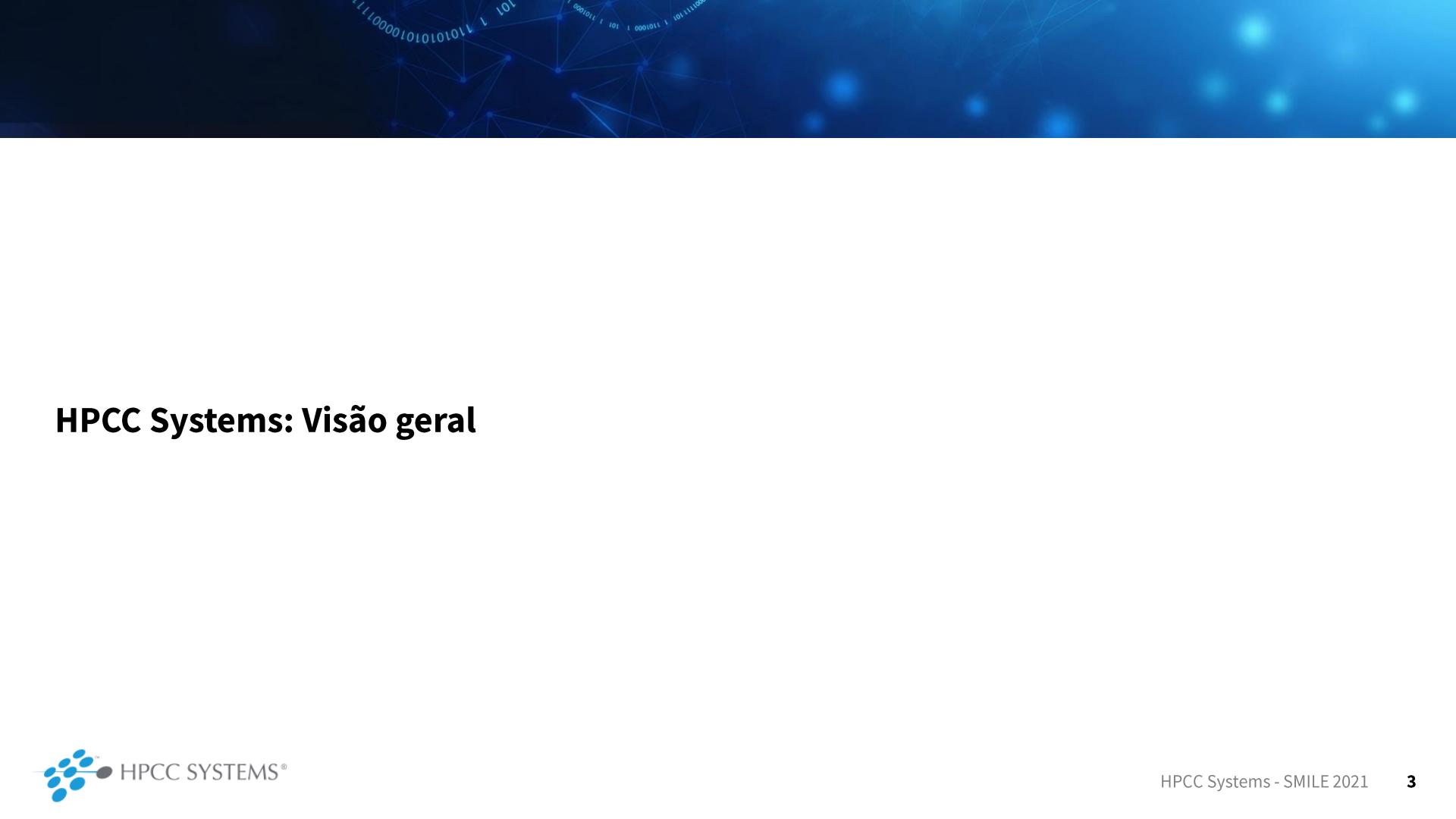
- ✓ O que é?
- ✓ Pra que serve?

## ✓ **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

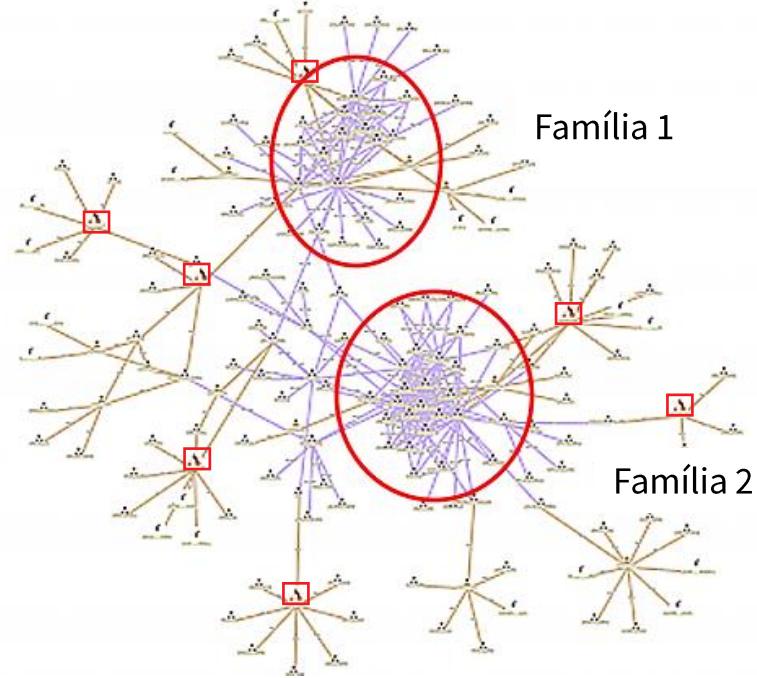
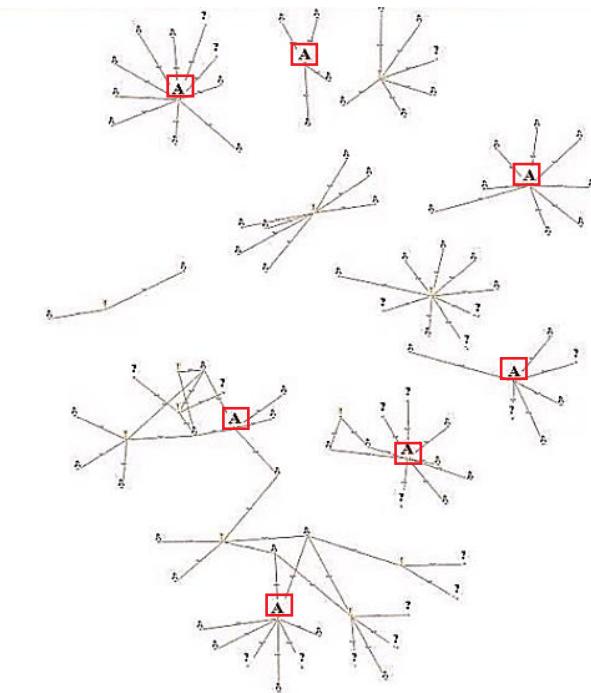
- Os cinco V's:
  - Volume
  - Variedade
  - Velocidade
  - Veracidade
  - Valor



# Trade off do big data

- Problema N<sup>2</sup>
- 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

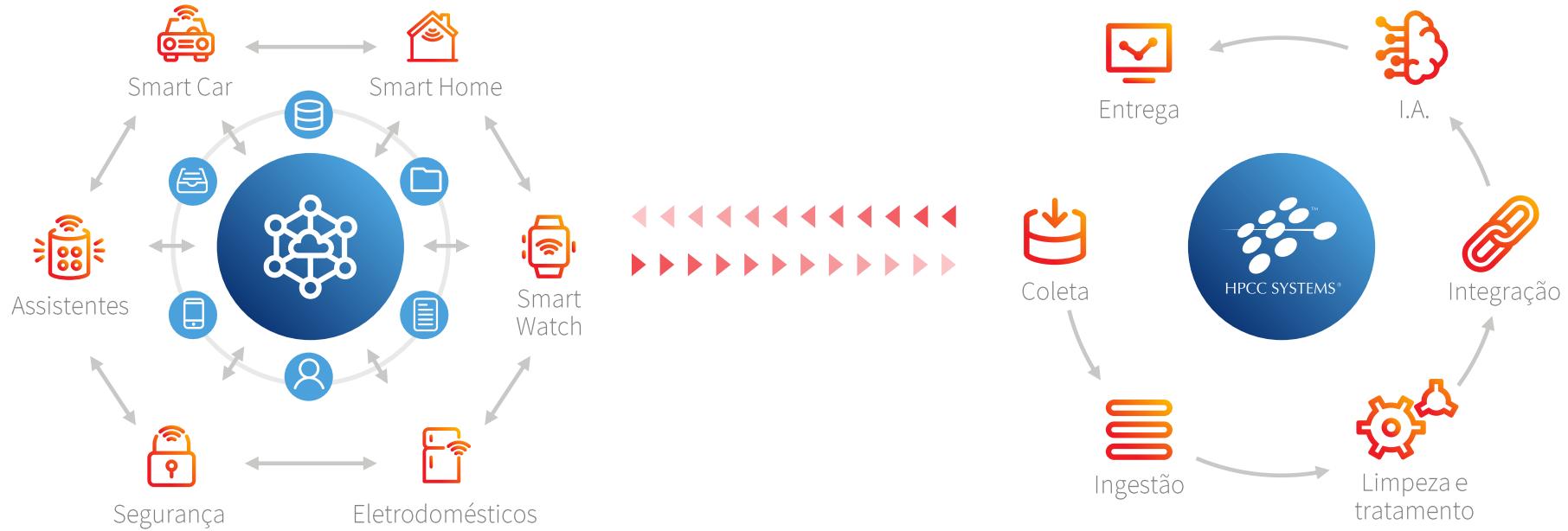
● Pessoas associadas aos acidentes



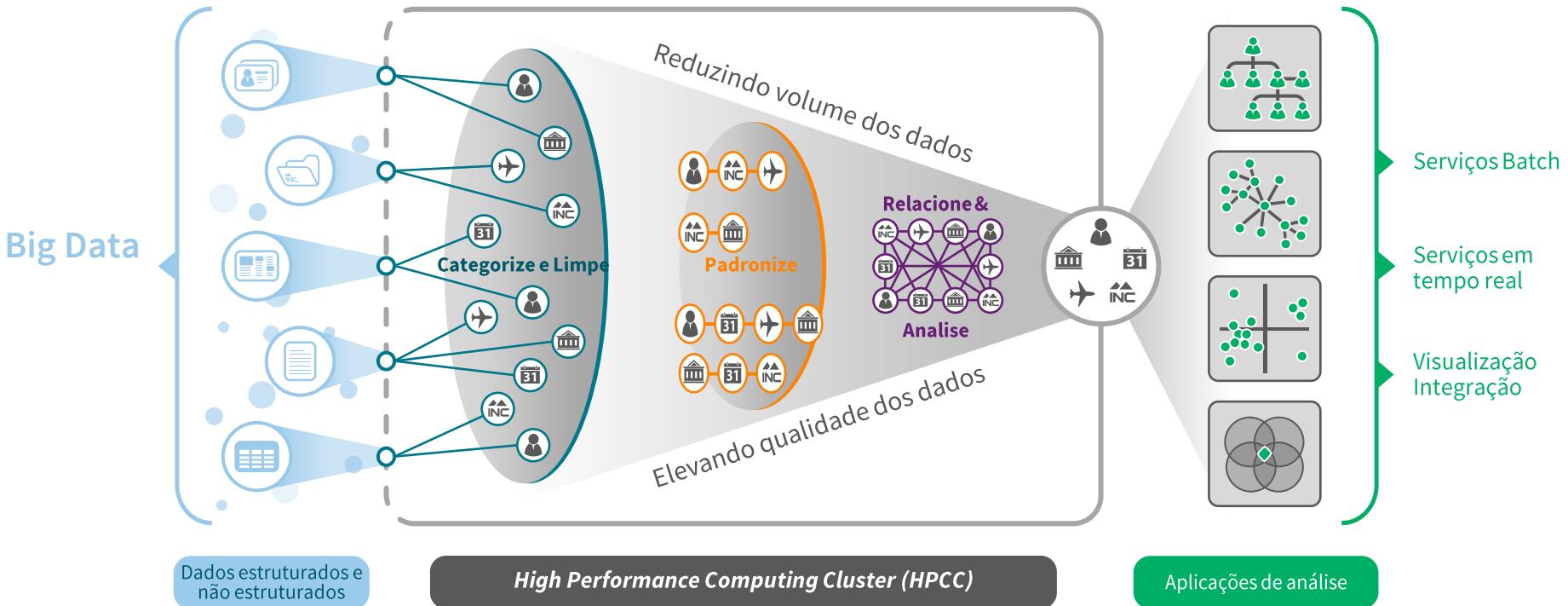
# Um exemplo mais atual ...

<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

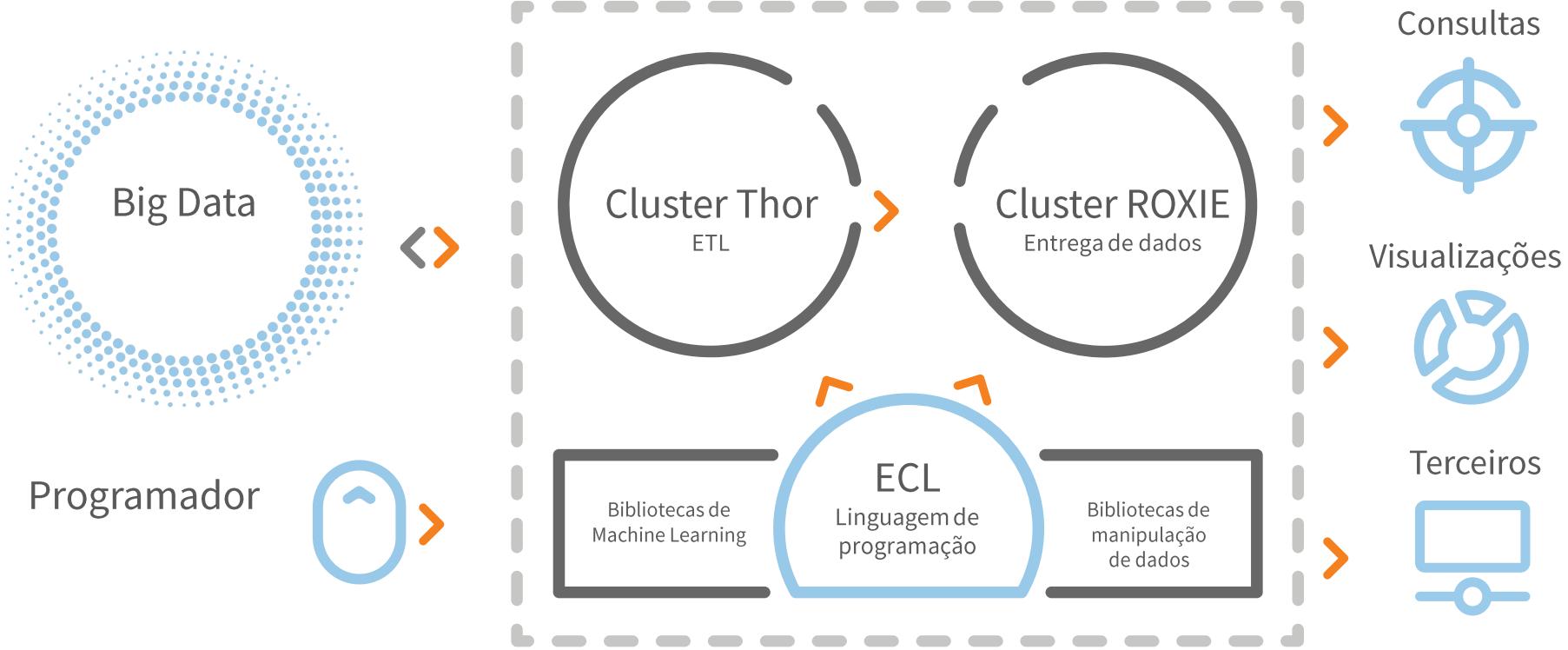
Supervisionado, não-supervisionado, aprendizagem profunda



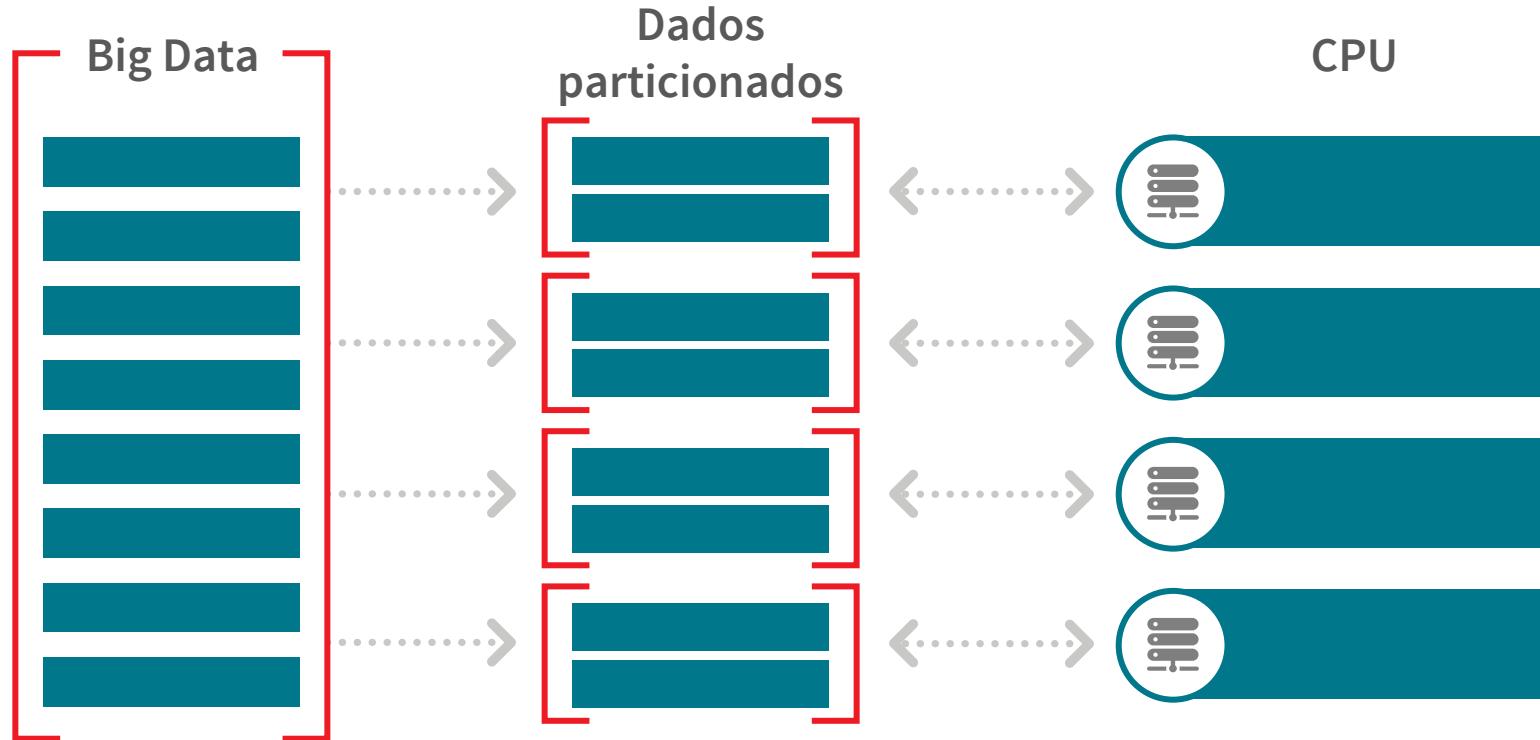
## Conectividade

Plugins de integração com outros sistemas

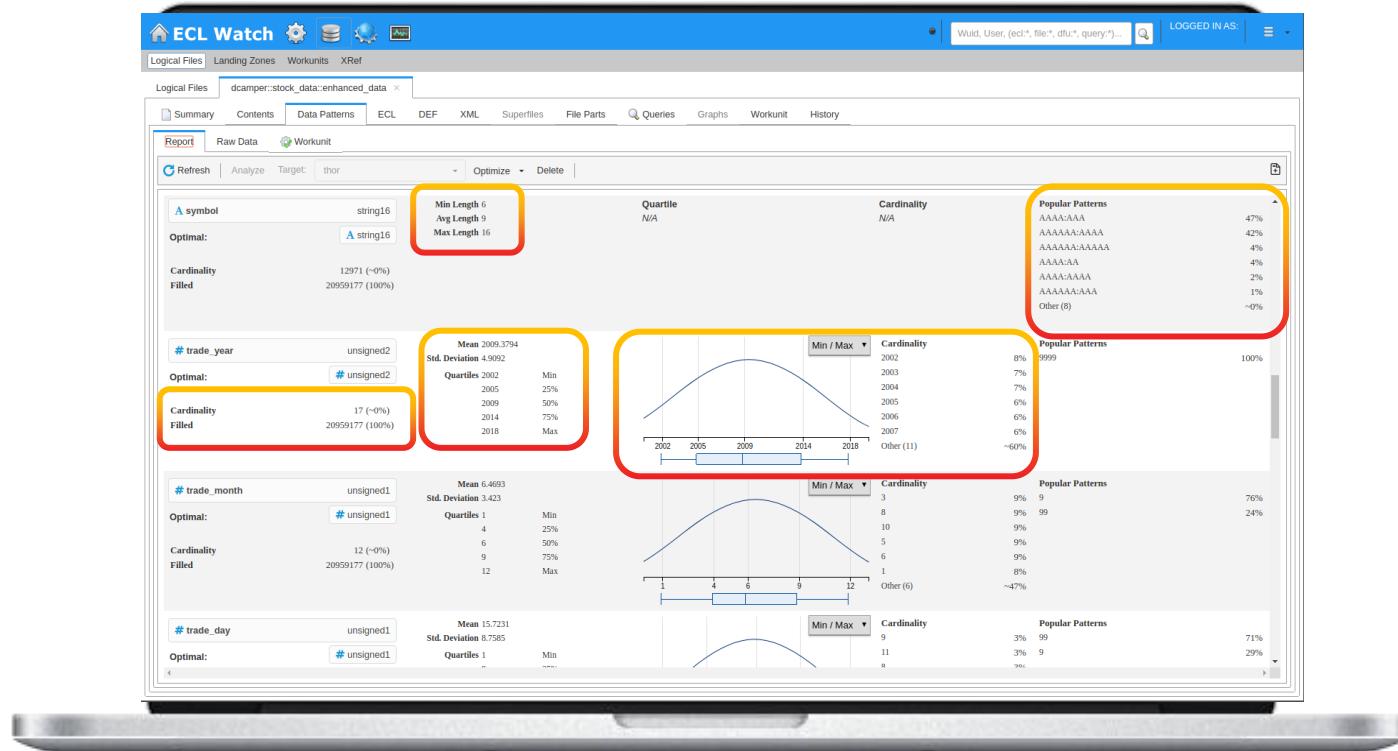
# Arquitetura do HPCC Systems



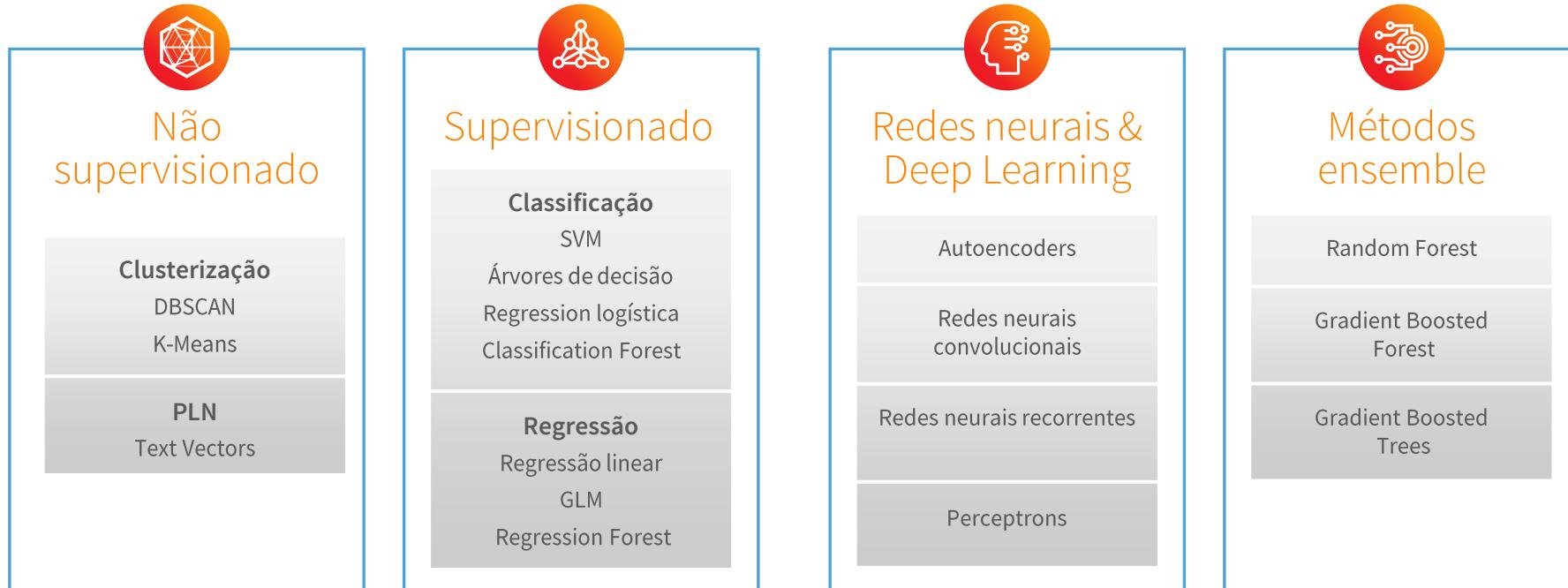
# Dados distribuídos e processamento paralelo



# 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

hthor  
fn\_getprice-xxx.1 Dynamic Form ▾

FN\_GETPRICE\_HMW\_1REQUEST

assess_val:	118720
bedrooms:	3
full_baths:	2
half_baths:	1
land_sq_ft:	14774
living_sq_ft:	1437
year_acq:	2011
year_built:	1968
zip:	95451

Output Tables FORM POST Submit Clear All

(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

### fn\_getprice-hmw.l Response

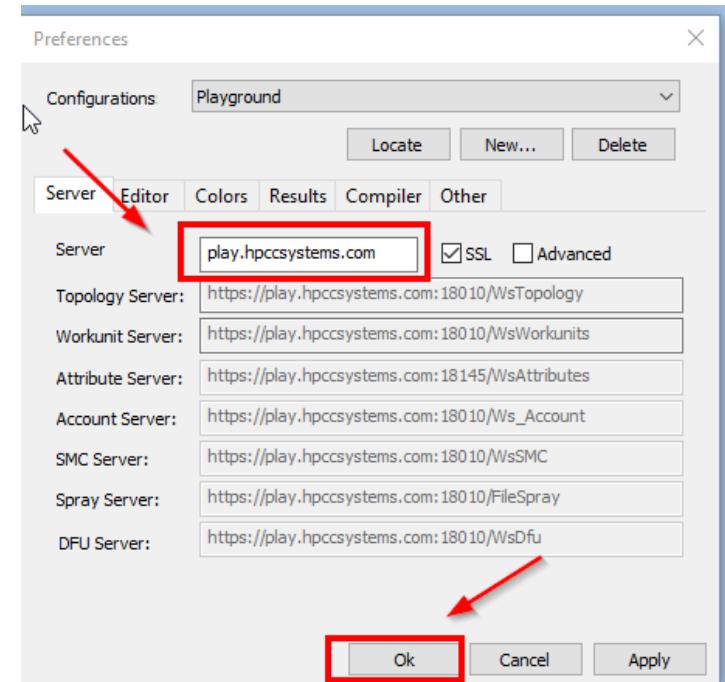
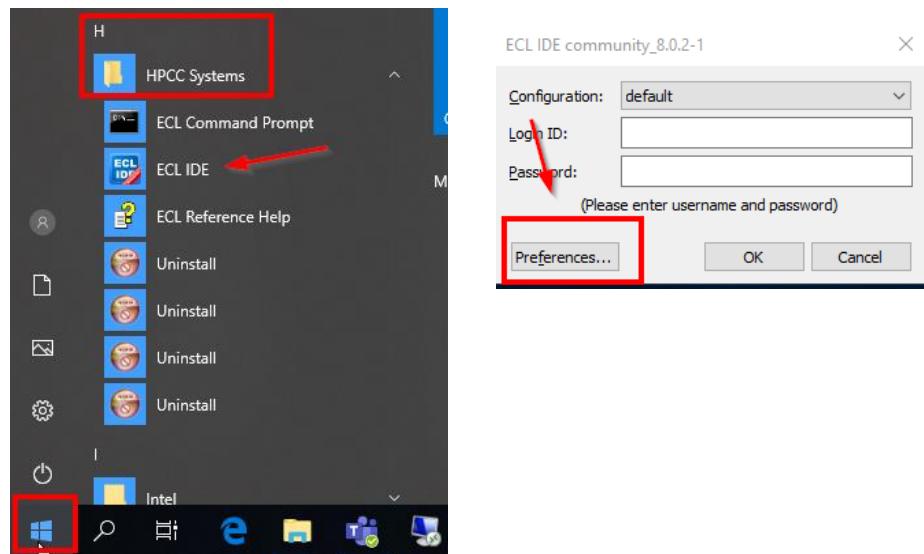
Dataset: Result 1

	preco
1	116732

# Preparação do ambiente

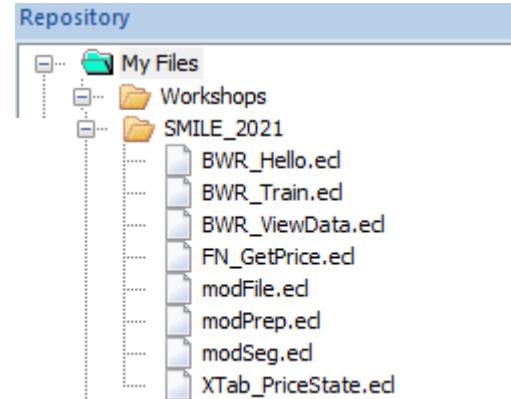
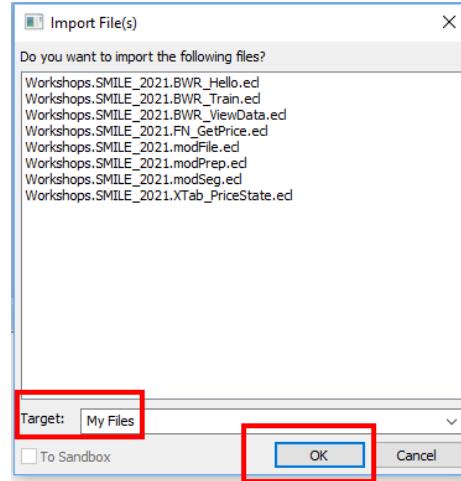
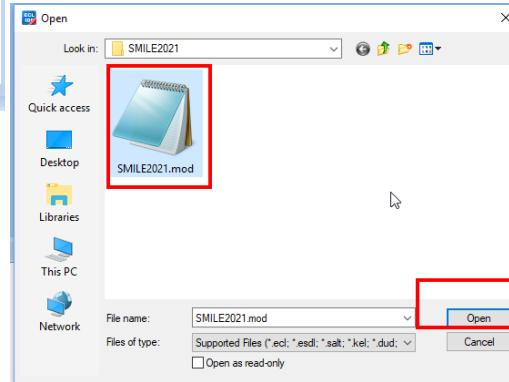
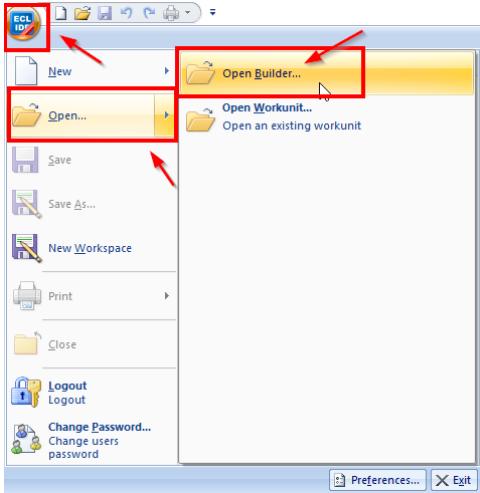
Cluster de treinamento: <https://play.hpccsystems.com:18010/>

ECL IDE:

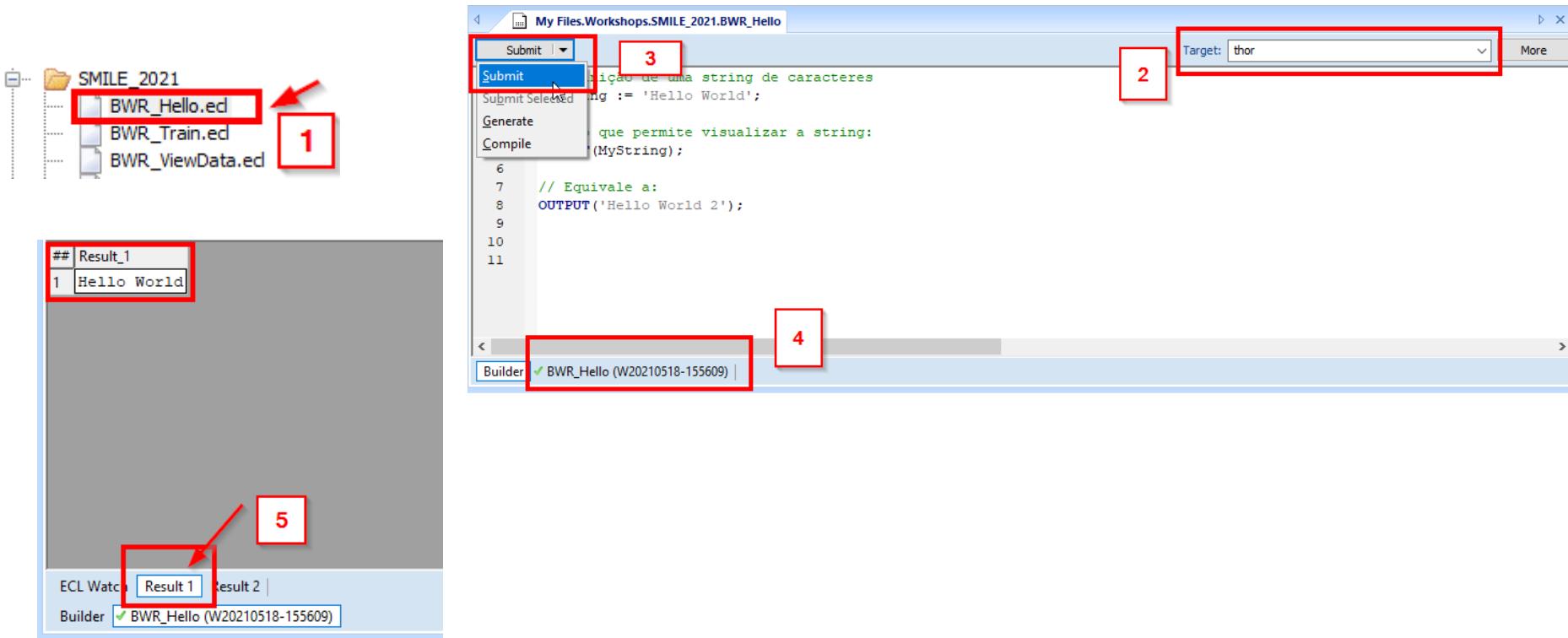


# Preparação do ambiente (cont.)

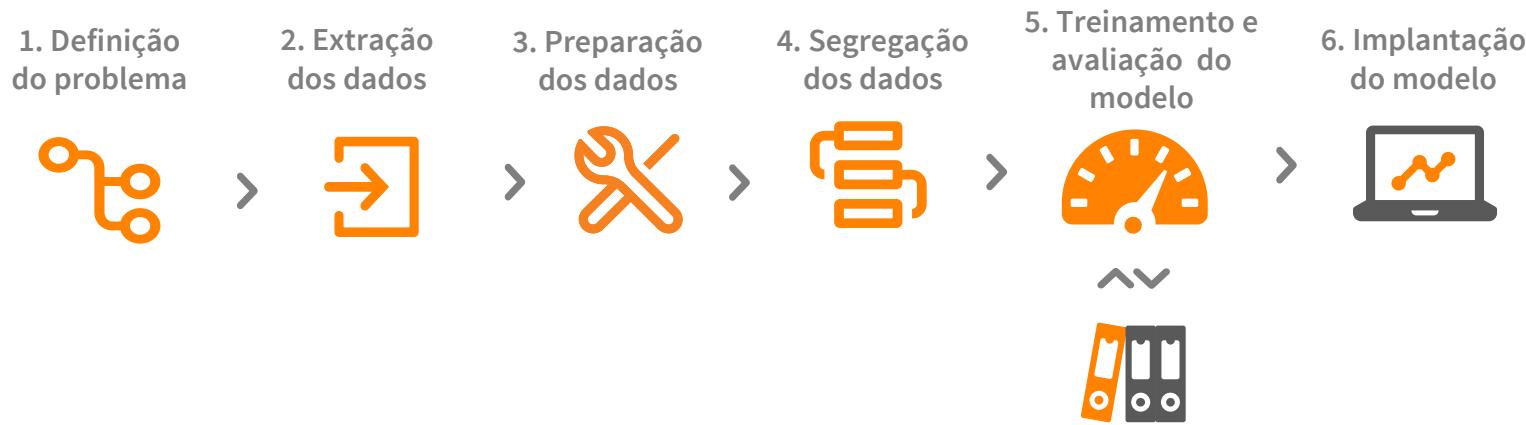
## SMILE2021.mod



# Teste do ambiente



# Fluxo de aprendizado supervisionado

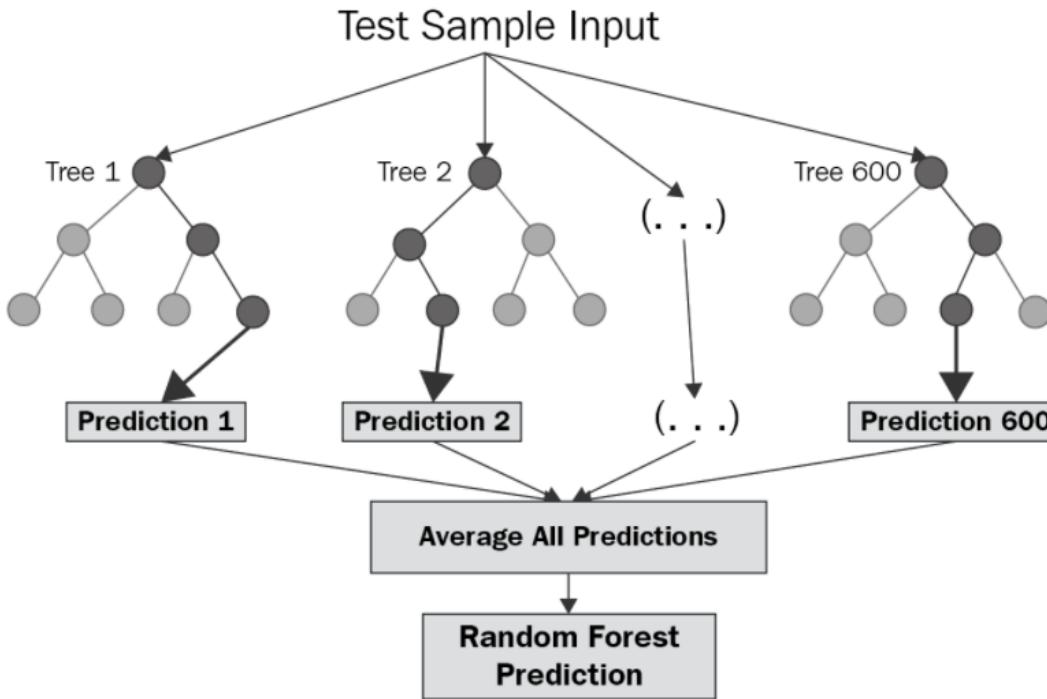


# 1. Definição do problema

“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.)



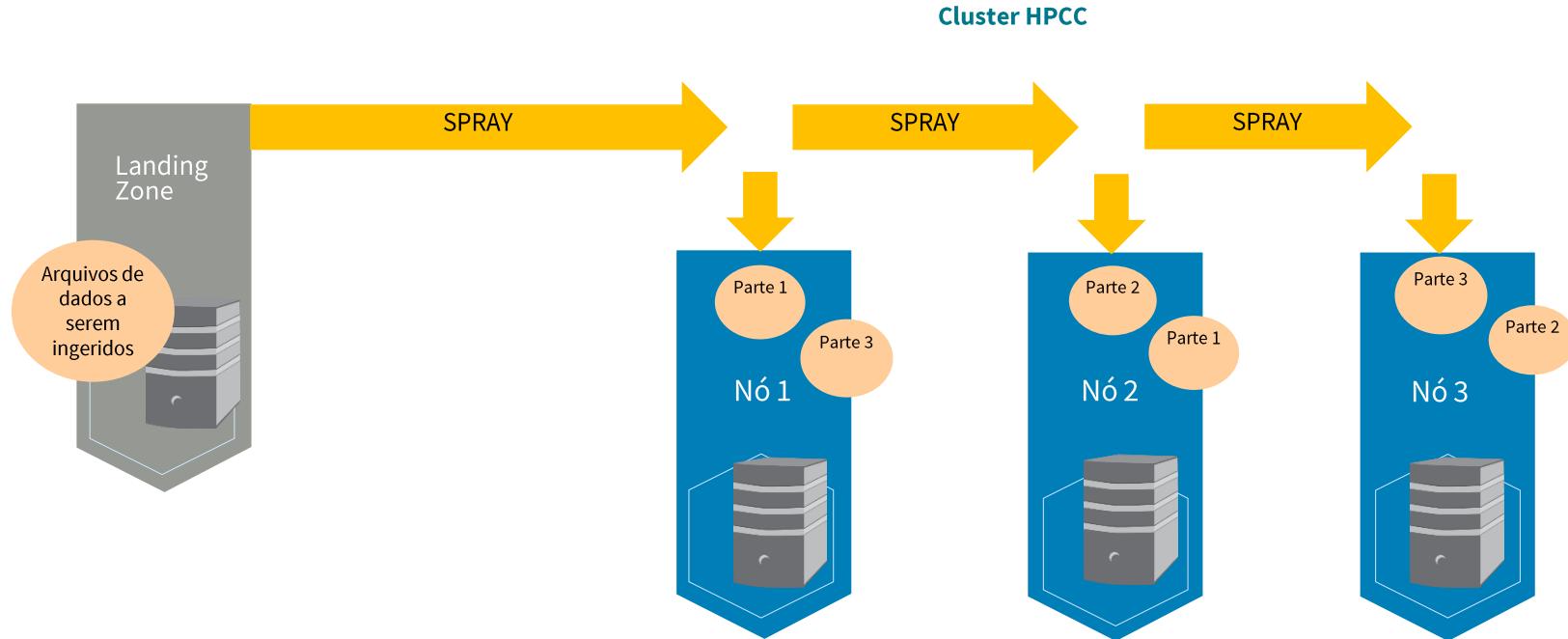
## 2. Extração dos dados

“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	214582956185891	54135	4			WAINWRIGHT	DR			NORTH FORT MYERS	FL	33917	159724
8	214582956185891	762012	125			SHIPYARD	DR		000150	MELBOURNE VILLAGE	FL	32904	96300
9	214582956185891	2331721	1190			LITTLEOAK	DR			HOUSTON	TX	77011	238854
10	214582956185891	3276109	2506			MEADOW	DR			LA QUINTA	CA	92253	30977
11	214582956185891	4414145	5508		S	STEVENS	AVE			MT LEMMON	AZ	85619	145000
12	345438575926606	1565512	521			SENON	DR			BAKERSFIELD	CA	93300	245000
13	562092156665191	1496019	485			VEAL	RD			SUGARLOAF	CA	92325	140000

##	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, the 'Logical Files' tab displays a tree view of files and folders. A red box labeled '1' highlights the 'propriedades' folder under the 'accounts' category. In the main workspace, the 'Delimited' tab is selected, indicated by a red box labeled '2'. The 'Target Name' field contains 'propriedadesXXX', highlighted by a red box labeled '3'. The 'Options' panel on the right shows various settings: 'Format: ASCII', 'Max Record Length: 8192', 'Separators: \,', 'Line Terminators: \n,\r\n', and 'Replicate: No Common: checked'. A red box labeled '4' highlights the 'Spray' button at the bottom of the options panel.

<https://play.hpccsystems.com:18010/>  
(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.

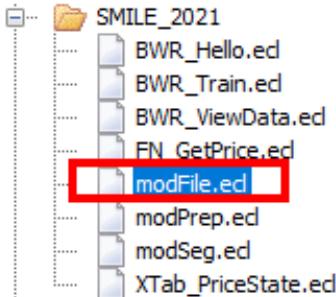
The bottom section shows a detailed view of the "...propriedadesxxx" file. The "Data Patterns" tab is selected, and the "Analyze" button is highlighted with a red box.

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.

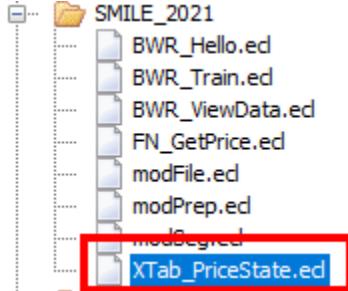
## 2. Extração dos dados (cont.)



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			PARMINTER	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

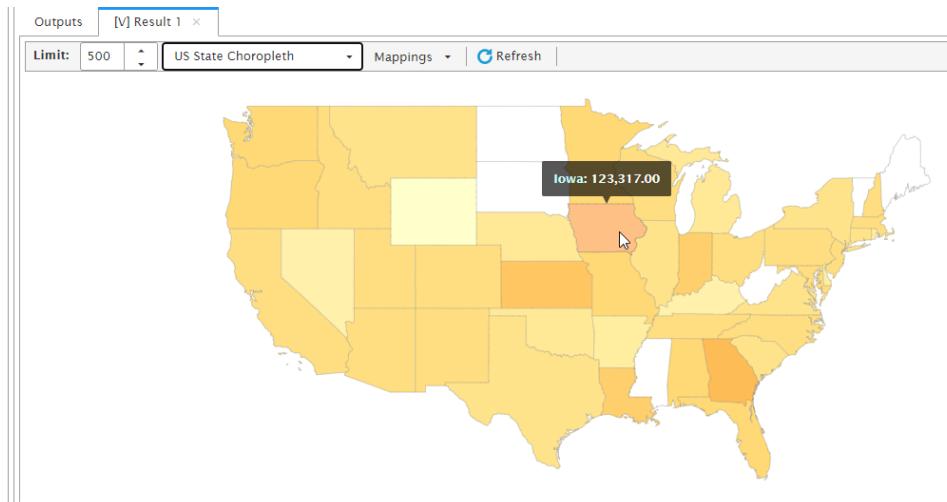


```
IMPORT $;

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
160350	20706	18020	2007	4610	2594	2	2	0	1916	185000
82569	60527	78477	2007	6098	1032	3	2	0	1992	78477
2192898	77386	3170	2010	14810	1760	3	2	1	1969	79290
2223942	92506	18660	2010	2700	924	3	1	0	1967	45511
4648854	77354	39900	2010	3000	2016	3	1	1	1908	39900
2367580	18104	108610	2007	16772	1223	3	2	0	1956	108610
607178	77018	31072	2006	2550	1152	3	2	1	2001	31072
1584497	90815	82132	2007	7350	2000	4	2	1	1987	88284
3615520	12764	341400	2008	5000	500	2	1	1	1972	341400
2103806	80013	58520	2007	5096	1650	3	2	1	1993	58520
2209348	75065	8540	2007	169927	1040	2	1	1	1962	87610
1298734	33474	66175	2007	4639	1354	4	1	0	1961	66175
1310023	86314	29400	2007	7515	1440	3	2	0	1988	301644
2840506	87121	94200	2006	15225	1708	3	2	0	1961	94200

### 3. Preparação dos dados



```
SMILE_2021
  BWR_Hello.ed
  BWR_Train.ed
  BWR_ViewData.ed
  FN_GetPrice.ed
  modFile.ed
  modPrep.ed
  modSeg.ed
  XTab_PriceState.ed

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('~/SMILE: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;
```

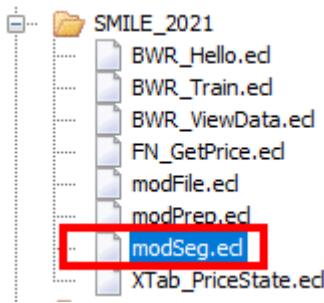
## 4. Segregação dos dados

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

wi	id	number	value
1	160350	1	185000
1	82569	1	78477
1	2192898	1	79290
1	2223942	1	45511
1	4648854	1	39900
1	2367580	1	108610
1	607178	1	31072
1	1584497	1	88284
1	3615520	1	341400
1	2103806	1	58520
1	2209348	1	87610
1	1298734	1	66175
1	1310023	1	301644
1	2840506	1	94200
1	3600022	1	262700
1	131449	1	16500
1	4649661	1	84000
1	1042629	1	38740
1	1732698	1	197700

wi	id	number	value
1	160350	1	20706
1	160350	2	18020
1	160350	3	2007
1	160350	4	4610
1	160350	5	2594
1	160350	6	2
1	160350	7	2
1	160350	8	0
1	160350	9	1916
1	82569	1	60527
1	82569	2	78477
1	82569	3	2007
1	82569	4	6098
1	82569	5	1032
1	82569	6	3
1	82569	7	2
1	82569	8	0
1	82569	9	1992

## 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

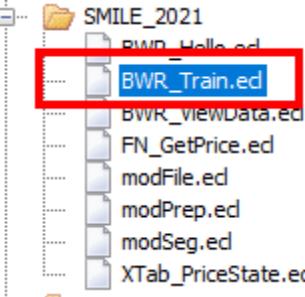
**“Obtenção de modelo a partir da amostra de treinamento e validação na amostra de teste”**

	wi	value	indexes
			Item
1	0	3372.0	3
			3
			1
2	0	2948.0	3
			3
			2
3	0	3479.0	3
			3
			3
4	0	2008.0	3
			3
			4
5	0	521.0	3
			3
			5
6	0	541.0	3
			3
			6

##	wi	id	number	value
1	1	1759	1	191080.8910737162
2	1	4482	1	189844.5799222115
3	1	21296	1	71741.64722389914
4	1	44342	1	81523.51183213605
5	1	49794	1	51806.91906294142
6	1	56887	1	574925.502836564
7	1	60740	1	201154.0583403441
8	1	64618	1	222536.1532409647
9	1	73244	1	94920.15844152826
10	1	74066	1	192821.5484365852

wi	regressor	r2	mse	rmse
1	1	0.8666527195133626	2109018505.726316	45924.05149511871

## 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,'~SMILE::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\_hmw Dynamic Form ▾

**FN\_GETPRICE\_HMWREQUEST**

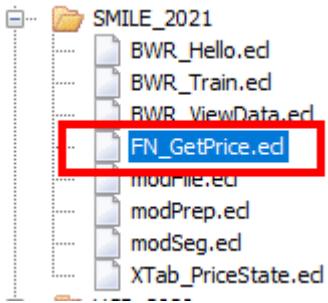
assess_val:	<input type="text"/>
bedrooms:	<input type="text"/>
full_baths:	<input type="text"/>
half_baths:	<input type="text"/>
land_sq_ft:	<input type="text"/>
living_sq_ft:	<input type="text"/>
year_acq:	<input type="text"/>
year_built:	<input type="text"/>
zip:	<input type="text"/>

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 two windows from the HPCC Systems interface:

- Top Window:** A code editor titled "My Files.Workshops.SMILE\_2021.FN\_GetPrice". The code is written in ECL. A context menu is open over the file "FN\_GetPrice.ed", with the "Compile" option highlighted. A red box labeled "1" highlights the "Target: roxie" dropdown at the top right of the window.
- Bottom Window:** A job submission and monitoring interface. It shows a job named "W20210518-174327" with status "compiled". The "Publish" tab is selected. A red box labeled "2" highlights the "Job Name: FN\_GetPrice-XXX" input field. The "Submit" button at the bottom right is also highlighted with a red box.

# Serviço disponível para uso!

ECL Watch

Queries Package Maps

Queries fn\_getprice-xxx.1

Refresh Open Delete Suspend Unsuspend Activate Deactivate Filter Set Options Recreate Query

ID fn\_getprice-xxx.1

Name

Test Pages

Legacy Form

Dataset: Result 1

fn\_getprice-hmw.1 Response

preco

1 116732

fn\_getprice-xxx.1 REQUEST

assess\_val: 118720  
bedrooms: 3  
full\_baths: 2  
half\_baths: 1  
land\_sq\_ft: 14774  
living\_sq\_ft: 1437  
year\_acq: 2011  
year\_built: 1968  
zip: 95451

Output Tables FORM POST Submit Clear All

HPCC SYSTEMS®

HPCC Systems - SMILE 2021

40



## Próximos passos

# Cursos online: +170 aulas ([learn.lexisnexis.com/hpcc](https://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

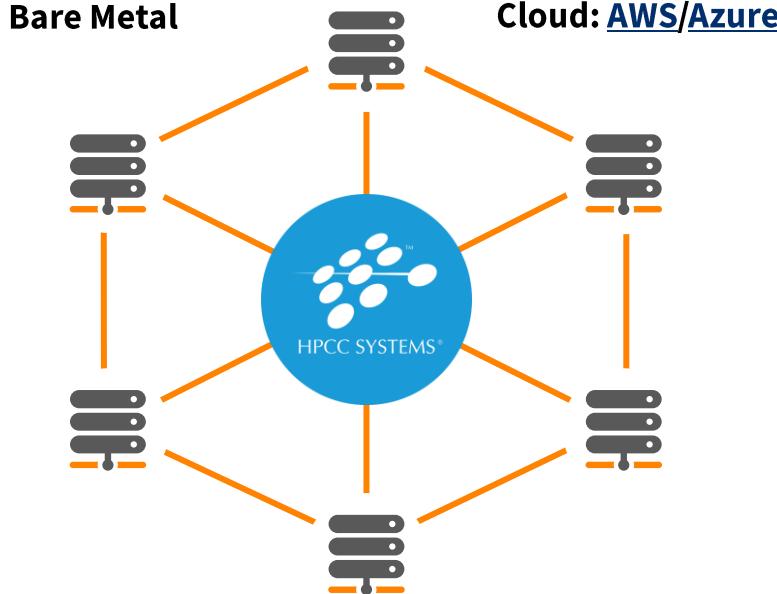
## Administração de Sistemas

- Conceitos e operação básica

## HPCC para gestores

- Visão geral e aplicações da plataforma

# Opções de uso: [play.hpccsystems.com](https://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](http://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.

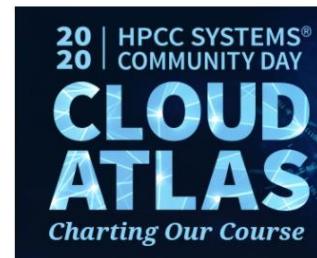
## Tecnologia de código aberto

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)



- 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/>



# Links úteis

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



Faça parte da  
Comunidade

Registre-se em [hpccsystems.com](http://hpccsystems.com)

# Considerações Finais



# 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