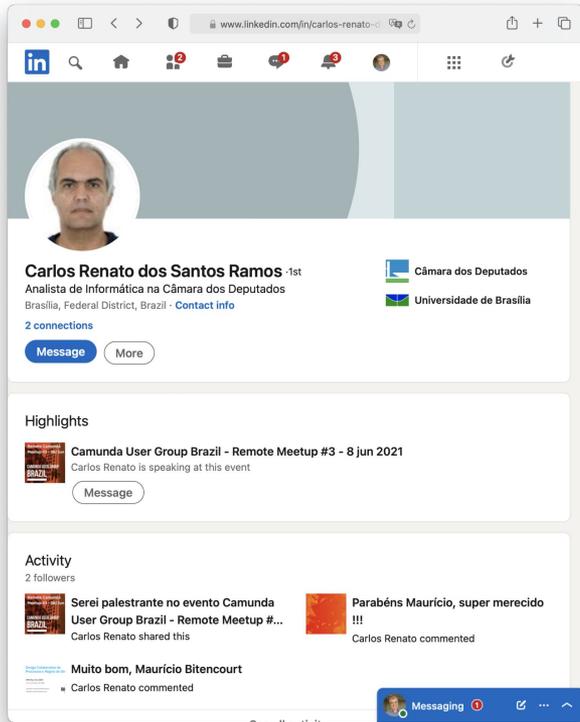


Remote Camunda Meetup #3 - 8 Jun 2021

CAMUNDA USER GROUP
BRAZIL

CAMUNDA

Automatizando testes de processos e regras de negócio no Camunda Run Community Edition utilizando Gherkin, Cucumber e Karate Framework, por **Carlos Renato dos Santos Ramos**, Analista de Informática na Câmara dos Deputados



Sou entusiasta de qualidade e testes. Acredito que o tipo de documentação que mais agrega valor é a especificação executável.

Processos e decisões executáveis (modelados com BPMN e DMN e implantados em um motor de processos e decisões) são especificações executáveis que mostram um caminho para o trabalho conjunto das áreas de Negócio e de TI visando desenvolver e manter aplicações com agilidade.

<https://www.linkedin.com/in/carlos-renato-dos-santos-ramos-a8a5301b9/>

Os Brigadeiros da Nice

Separação entre Processos
e Regras de negócio



Escrito e produzido por:

Fabrício Laguna

flaguna@giganteconsultoria.com.br

<https://www.linkedin.com/in/fabricio-laguna/>

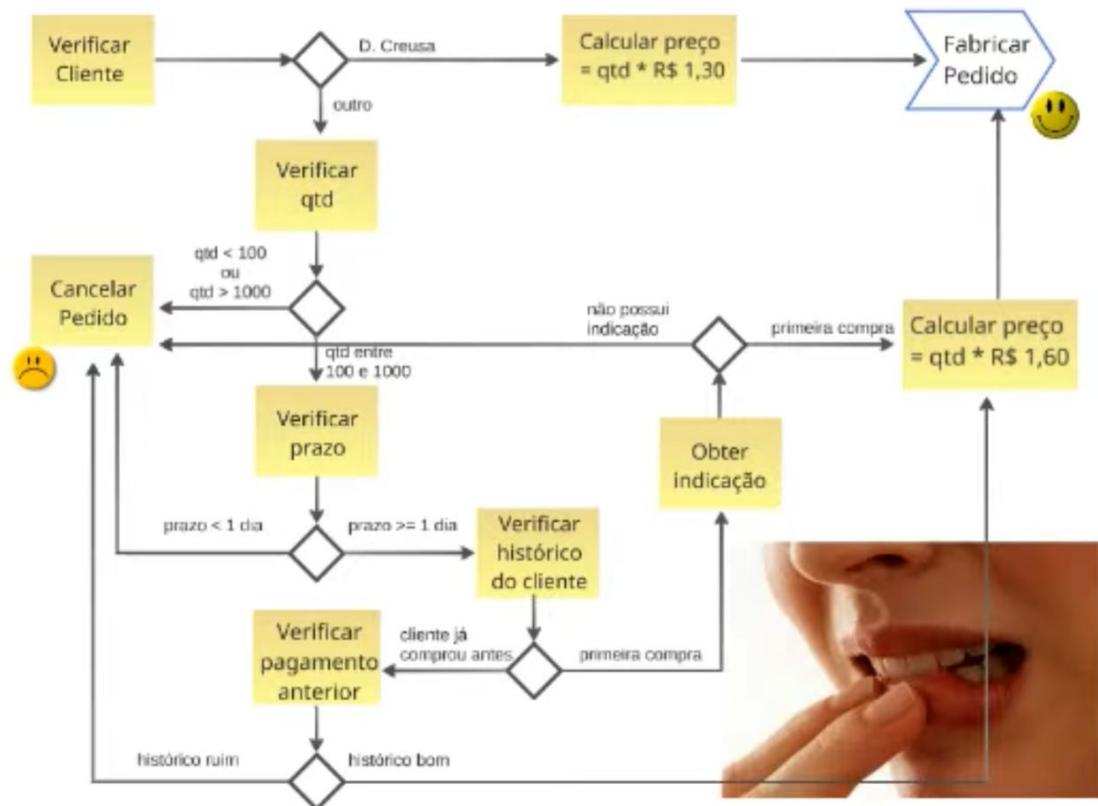
<https://www.youtube.com/watch?v=Y6sb1Eos1sl>

Processo ponta a ponta:
Do pedido à entrega de brigadeiros

Processo de Negócio



Fluxo do Processo: Negociar Pedido



Fluxo do Processo: Negociar Pedido



Regras de Decisão

O pedido é válido?



Pedido Válido
Tipo de Cliente
Vale a pena

Pedido Válido			
Nome	Perfil	Quantidade	Preço
Cliente 1	Perfil 1	10	100
Cliente 2	Perfil 2	20	200
Cliente 3	Perfil 3	30	300

Vale a pena
Quantidade pedida
Prazo pedido

Vale a pena			
Nome	Perfil	Quantidade	Prazo
Cliente 1	Perfil 1	10	10
Cliente 2	Perfil 2	20	20
Cliente 3	Perfil 3	30	30

Tipo de Cliente
Perfil do pagador
Tem indicação
Nome

Tipo de Cliente			
Nome	Perfil	Quantidade	Preço
Cliente 1	Perfil 1	10	100
Cliente 2	Perfil 2	20	200
Cliente 3	Perfil 3	30	300

Perfil do pagador
Já comprou antes
Pagou no prazo

Perfil do pagador			
Nome	Perfil	Quantidade	Preço
Cliente 1	Perfil 1	10	100
Cliente 2	Perfil 2	20	200
Cliente 3	Perfil 3	30	300

Qual o valor do pedido?



Valor do Pedido
Tipo de Cliente
Quantidade solicitada

Valor do Pedido			
Nome	Perfil	Quantidade	Preço
Cliente 1	Perfil 1	10	100
Cliente 2	Perfil 2	20	200
Cliente 3	Perfil 3	30	300

Tabelas de Decisão

Pedido Válido		
Tipo de Cliente	Vale a pena	Resultado
VIP	---	sim
Comum	sim	sim
Comum	não	não
Ruim	---	não

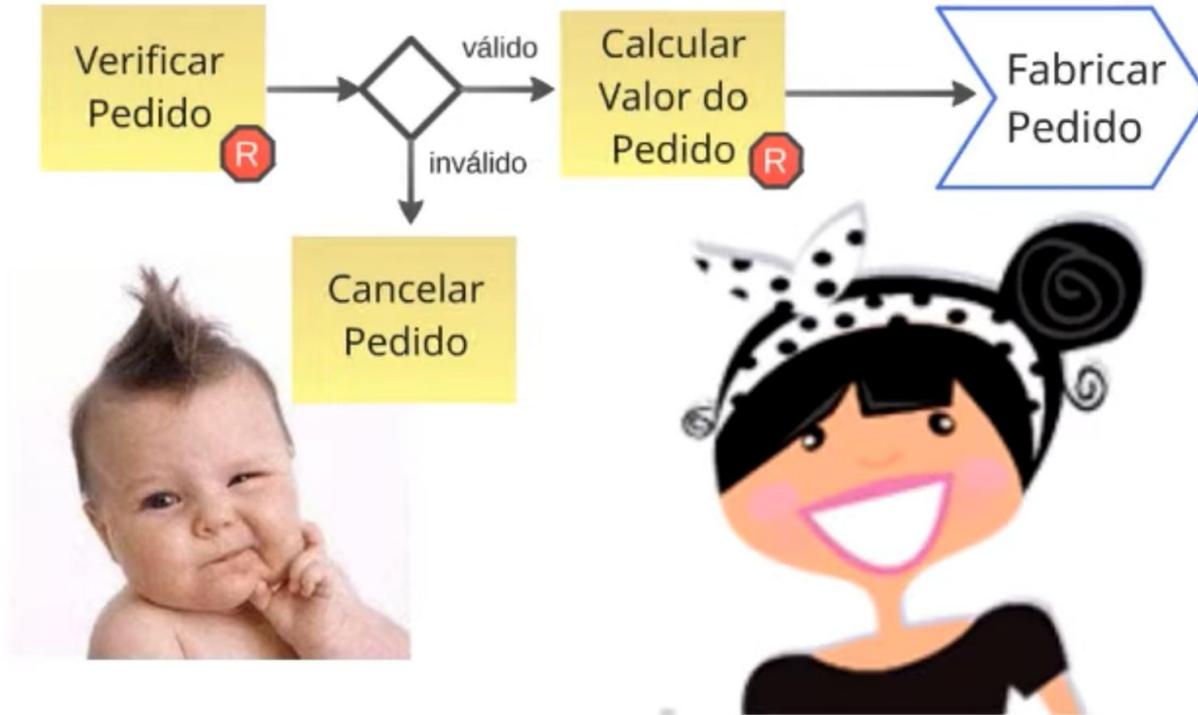
Valor do Pedido	
Tipo de Cliente	Resultado
VIP	qtd * R\$ 1,30
qualquer outro	qtd * R\$ 1,60

Vale a pena		
Quantidade pedida	Prazo pedido	Resultado
< 100	---	não
> 1000	---	não
entre 100 e 1000	< 1 dia	não
	>= 1 dia	sim

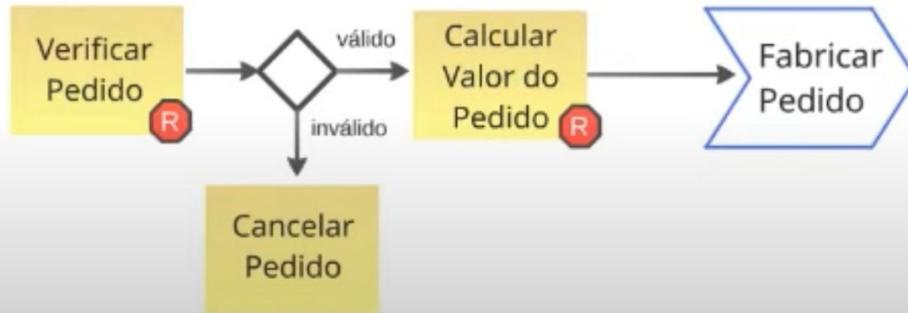
Tipo de Cliente			
Nome	Perfil pagador	Indicação	Resultado
D. Creusa	---	---	VIP
qualquer outro	bom pagador	---	Comum
	desconhecido	sim	Comum
	desconhecido	não	Ruim
	caloteiro	---	Ruim

Perfil do pagador		
Já comprou	Pagou	Resultado
sim	sim	bom pagador
sim	não	caloteiro
não	--	desconhecido

Fluxo do Processo: Negociar Pedido

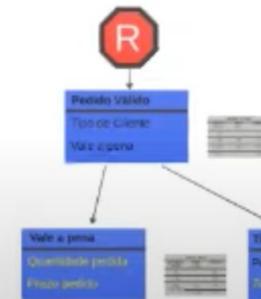


Processos

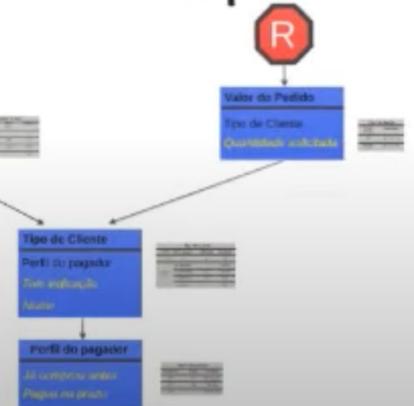


Regras

O pedido é válido?



Qual o valor do pedido?



Épico e Histórias de Usuário

Sendo um interessado em brigadeiros
Posso solicitar um pedido
Pois assim oferecê-los na minha festa

Sendo um interessado em brigadeiros
Posso ter o pedido analisado
Pois assim obter a aceitação do meu pedido

Sendo um interessado em brigadeiros com pedido aceito
Posso passar pela fabricação do pedido
Pois assim obter o pedido fabricado

Sendo um interessado em brigadeiros com pedido fabricado
Posso passar pela entrega de pedido
Pois assim obter o pedido entregue

História de Usuário e Critérios de Aceitação

Sendo um interessado em brigadeiros

Posso ter o pedido analisado

Pois assim obter a aceitação do meu pedido

Devem ser válidos apenas os pedidos que valem a pena e o cliente não é ruim

Devem ser considerados pedidos que valem a pena aqueles que a quantidade pedida e o prazo são aceitáveis

Devem ser considerados clientes ruins aqueles que já compraram e não pagaram

Devem ser considerados clientes normais aqueles que são indicados ou já compraram e pagaram

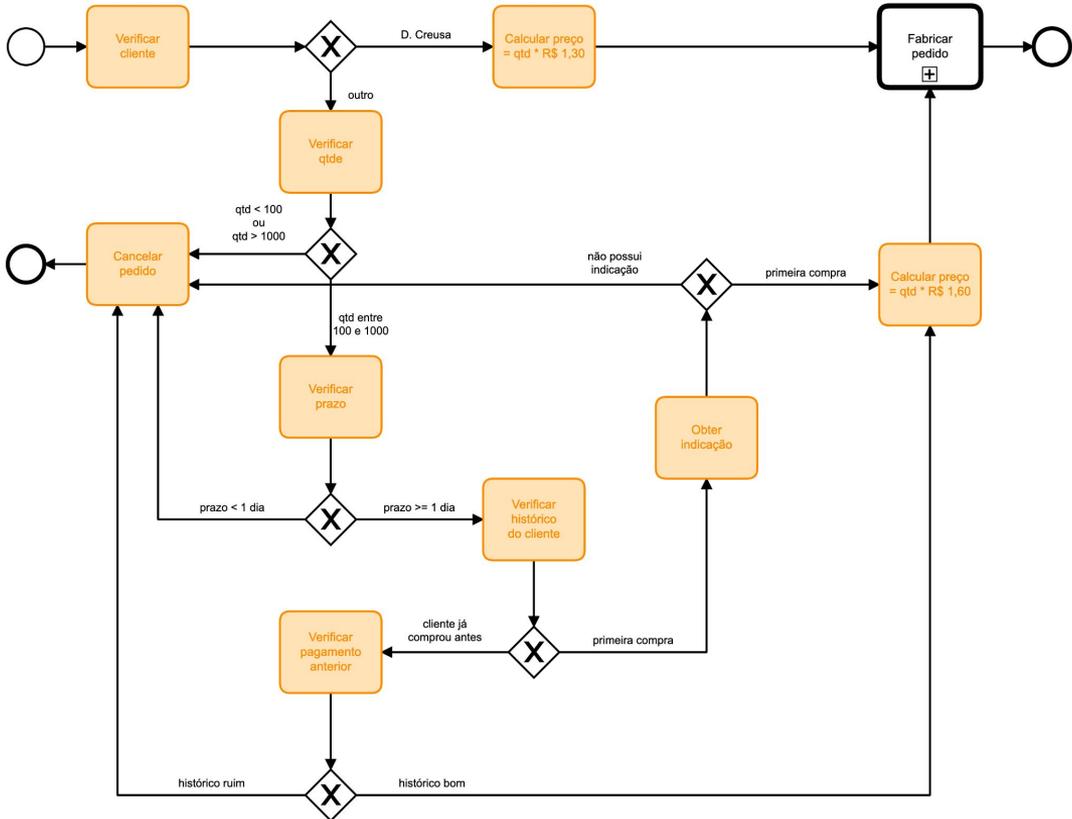
Devem ser considerados clientes vip aqueles que constam de lista específica

Deve ser cobrado pelo pedido o produto da quantidade pedida pelo preço unitário

Deve ser cobrado um valor unitário menor para cliente vip do que o cobrado para cliente normal

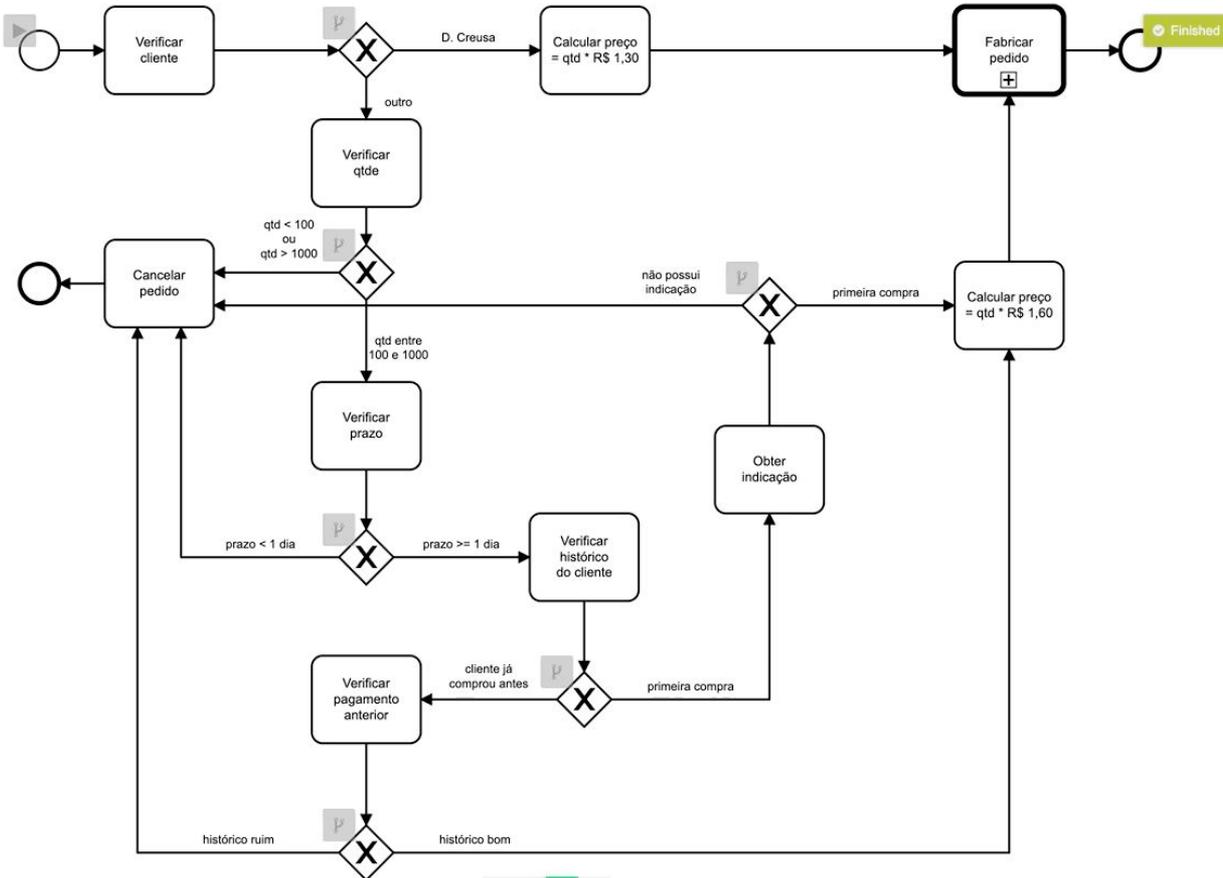
Fluxo do Processo: Negociar Pedido

Antes da separação de processos e decisões



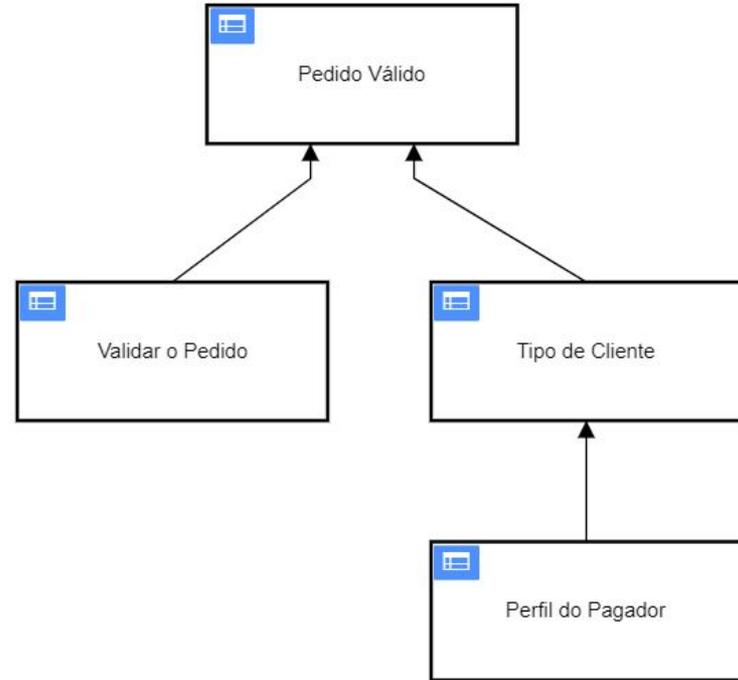
BrigadeirosDaNiceMacarronico.bpmn x

Token Simulation



Regras de Decisão do Processo: Negociar Pedido

DRD - Diagrama de
Requisito de Decisão



Créditos:

Marcelo Eduardo Schubert, Coordenador do Escritório de Gestão de Processos na Secretaria de Estado da Administração Santa Catarina e participante do curso Roadmap » Design de Processos e Regras de Decisão com Modelos Executáveis

Tabelas de Decisão em DMN

Pedido Válido		Hit Policy: Unique	
When	And	Then	Annotations
Tipo Cliente string	Vale a pena "Sim","Não"	Resultado string	
1 "VIP"	-	"Fabricar pedido"	Vale a pena
2 "Comum"	"Sim"	"Fabricar pedido"	
3 "Comum"	"Não"	"Cancelar Pedido"	
4 "Ruim"	-	"Cancelar pedido"	
+ -	-	-	



Validar o Pedido		Hit Policy: Unique	
When	And	Then	Annotations
Quantidade pedida integer	Prazo Pedido integer	Vale a pena string	
1 <100	-	"Não"	
2 >1000	-	"Não"	
3 [100..1000]	<1	"Não"	
4 [100..1000]	>=1	"Sim"	
+ -	-	-	

Tipo de Cliente		Hit Policy: First		
When	And	And	Then	Annotations
Nome string	Indicação string	Perfil Pagador string	Tipo do Cliente string	
1 "Creusa"	-	-	"VIP"	
2 -	-	"Bom pagador"	"Comum"	
3 -	"Sim"	"Desconhecido"	"Comum"	
4 -	"Não"	"Desconhecido"	"Ruim"	
5 -	-	"Caloteiro"	"Ruim"	
+ -	-	-	-	



Perfil do Pagador		Hit Policy: Unique	
When	And	Then	Annotations
Já comprou "Sim","Não"	Pagou "Sim","Não"	Perfil do Pagador string	
1 "Sim"	"Sim"	"Bom pagador"	
2 "Sim"	"Não"	"Caloteiro"	
3 "Não"	-	"Desconhecido"	
+ -	-	-	

Créditos:

Marcelo Eduardo Schubert, Coordenador do Escritório de Gestão de Processos na Secretaria de Estado da Administração Santa Catarina e participante do curso Roadmap » Design de Processos e Regras de Decisão com Modelos Executáveis

Simulação do Pedido Válido

The screenshot displays the DMN Simulator interface. At the top, the browser address bar shows `consulting.camunda.com/dmn-simulator/`. The application header includes the Camunda logo and the title "DMN Simulator".

Inputs:

- Decision table: `All tables` (dropdown)
- Beverages:**
 - Guests with children: `True` (dropdown), type: `boolean`
- Dish:**
 - Season: `Enter String` (text input), type: `string`
 - How many guests: `0` (text input), type: `integer`

Outputs:

- Beverages: (empty text input)
- Dish: (empty text input)

A green "Simulate now" button is located below the input fields.

The main area contains a decision diagram titled "Dinner Decisions" with the ID `dinnerDecisions`. The diagram features:

- A central **Dish** decision node.
- Inputs: `Season` (oval) and `Number of Guests` (oval) pointing to the **Dish** node.
- Output: `Men's Cookbook` (document icon) connected to the **Dish** node via a dashed arrow.
- A **Beverages** decision node pointing to the **Dish** node.
- Input: `Guests with children?` (oval) pointing to the **Beverages** node.

The BPMN.iO logo is visible in the bottom right corner of the diagram area.

Simulação do Pedido Inválido

The screenshot displays the DMN Simulator interface. At the top, the browser address bar shows `consulting.camunda.com/dmn-simulator/`. The page title is "DMN Simulator".

Inputs:

- Decision table: `All tables` (dropdown)
- Beverages:**
 - Guests with children: `True` (dropdown), type: `boolean`
- Dish:**
 - Season: `Enter String` (text input), type: `string`
 - How many guests: `0` (text input), type: `integer`

Outputs:

- Beverages: (empty text box)
- Dish: (empty text box)

A green "Simulate now" button is located below the input fields.

The main area contains a decision diagram titled "Dinner Decisions" with the ID `dinnerDecisions`. The diagram shows a central "Dish" decision node. It has three incoming arrows: one from a "Season" input node, one from a "Number of Guests" input node, and one from a "Guests with children?" decision node. A "Men's Cookbook" document icon is connected to the "Dish" node via a dashed arrow. The "Dish" node has two outgoing arrows: one to a "Beverages" output node and one to the "Guests with children?" decision node. A "BPMN.io" logo is visible in the bottom right corner of the diagram area.

At the bottom of the interface, there are two buttons: "Upload DMN Table" and "Download DMN Table".

Separação de processos e regras de decisão

Não afeta o
comportamento externo
da aplicação

- mesmas funcionalidades
- mesmos critérios de aceitação de cada funcionalidade
- mesmos testes de aceitação de cada funcionalidade

Altera a
arquitetura interna
da aplicação

- processo mais simples (facilidade de entendimento e manutenção)
- processo mais estável (regras de decisão mudam sem impacto no processo)

Por que a arquitetura importa ?

Texto citado:

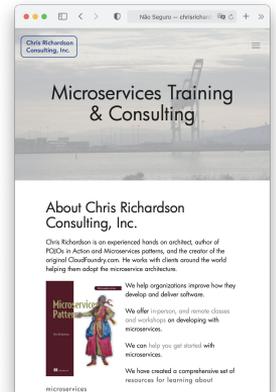
*An application has two categories of requirements. The first category includes the **functional** requirements, which define what the application must do. They're usually in the form of use cases or user stories.*

*Architecture has very little to do with the functional requirements. You can implement functional requirements with almost any architecture, even a **big ball of mud**.*

Tradução livre:

Um aplicativo possui duas categorias de requisitos. A primeira categoria inclui os requisitos **funcionais**, que definem o que o aplicativo deve fazer. Eles geralmente estão na forma de casos de uso ou histórias de usuários.

A arquitetura tem muito pouco a ver com os requisitos funcionais. Você pode implementar requisitos funcionais com quase qualquer arquitetura, até mesmo uma **grande bola de lama**.



Por que a arquitetura importa ?

Texto citado:

*Architecture is important because it enables an application to satisfy the second category of requirements: its **quality of service** requirements. These are also known as **quality attributes**.*

*The quality of service requirements define the **runtime qualities** such as scalability and reliability. They also define **development time qualities** including maintainability, testability, and deployability, which determine development velocity and agility.*

The architecture you choose for your application determines how well it meets these quality requirements.

Tradução livre:

A arquitetura é importante porque permite que um aplicativo satisfaça a segunda categoria de requisitos: seus requisitos de **qualidade de serviço**. Eles também são conhecidos como **atributos de qualidade**.

Os requisitos de qualidade de serviço definem as **qualidades de tempo de execução**, como escalabilidade e confiabilidade. Eles também definem as **qualidades do tempo de desenvolvimento**, incluindo capacidade de manutenção, testabilidade e implementação, que determinam a velocidade e agilidade de desenvolvimento.

A arquitetura que você escolhe para seu aplicativo determina o quão bem ele atende a esses requisitos de qualidade.

<http://chrisrichardson.net/post/architecture/2019/07/23/who-does-architecture.html>



Processo executável Negociar Pedido

- Modelagem do processo com **Camunda Modeler**
 - modelagem ponta a ponta com BPMN 2.0
 - diagrama de requisitos de decisão completo com DMN 1.3
- Execução de processo e decisões com **Camunda Plataforma Run**
 - Cockpit (`http://localhost:8080/camunda/app/cockpit`)
 - Tasklist (`http://localhost:8080/camunda/app/tasklist`)
 - API REST (`http://localhost:8080/engine-rest`)

Camunda Platform Run

```
camunda-bpm-run-ee-7.15.2-ee — java • start.sh — 173x29
MacBook-Pro-15:camunda-bpm-run-ee-7.15.2-ee mauriciobitencourt$ ./start.sh
JAVA_HOME is not set. Unexpected results may occur.
Set JAVA_HOME to the directory of your local JDK to avoid this message.
REST API enabled
WebApps enabled
Swagger UI enabled
classpath: ./internal/webapps/./internal/rest/./internal/swaggerui/./configuration/userlib/./configuration/keystore/

Camunda Platform Run

Spring-Boot: (v2.4.3)
Camunda Platform: (v7.15.2-ee)

2021-06-04 17:33:45.383 INFO 19401 --- [main] org.camunda.bpm.run.CamundaBpmRun : Starting CamundaBpmRun v7.15.2-ee using Java 11.0.11 on MacBook-Pro-15 w
ith PID 19401 (/Applications/camunda-bpm-run-ee-7.15.2-ee/internal/camunda-bpm-run-core.jar started by mauriciobitencourt in /Applications/camunda-bpm-run-ee-7.15.2-ee)
2021-06-04 17:33:45.388 INFO 19401 --- [main] org.camunda.bpm.run.CamundaBpmRun : No active profile set, falling back to default profiles: default
2021-06-04 17:33:47.102 INFO 19401 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2021-06-04 17:33:47.123 INFO 19401 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-06-04 17:33:47.124 INFO 19401 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.43]
2021-06-04 17:33:47.236 INFO 19401 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2021-06-04 17:33:47.236 INFO 19401 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1766 ms
2021-06-04 17:33:47.291 INFO 19401 --- [main] .c.b.s.b.s.r.CamundaJerseyResourceConfig : Configuring camunda rest api.
2021-06-04 17:33:47.334 INFO 19401 --- [main] .c.b.s.b.s.r.CamundaJerseyResourceConfig : Finished configuring camunda rest api.
2021-06-04 17:33:47.807 INFO 19401 --- [main] org.camunda.bpm.spring.boot : STARTER-SB040 Setting up jobExecutor with corePoolSize=3, maxPoolSize:10
2021-06-04 17:33:47.810 INFO 19401 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'camundaTaskExecutor'
```

<https://docs.camunda.org/manual/latest/user-guide/camunda-bpm-run/>

<https://camunda.com/download/enterprise/>

Diagrama em BPMN 2.0

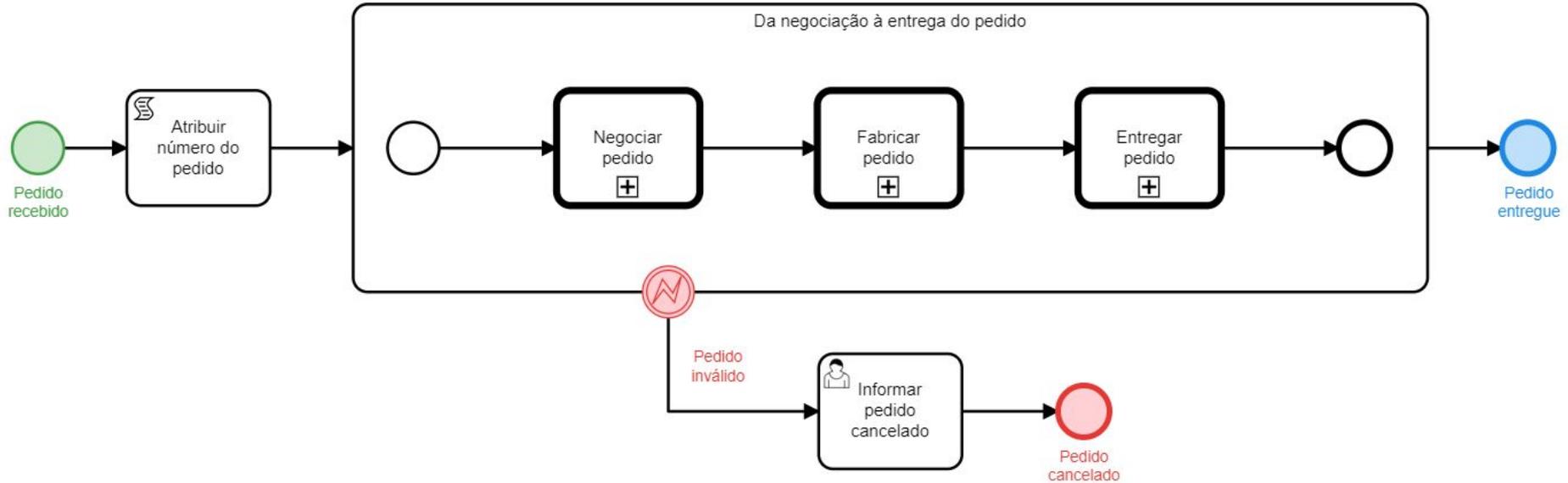


Diagrama em BPMN 2.0

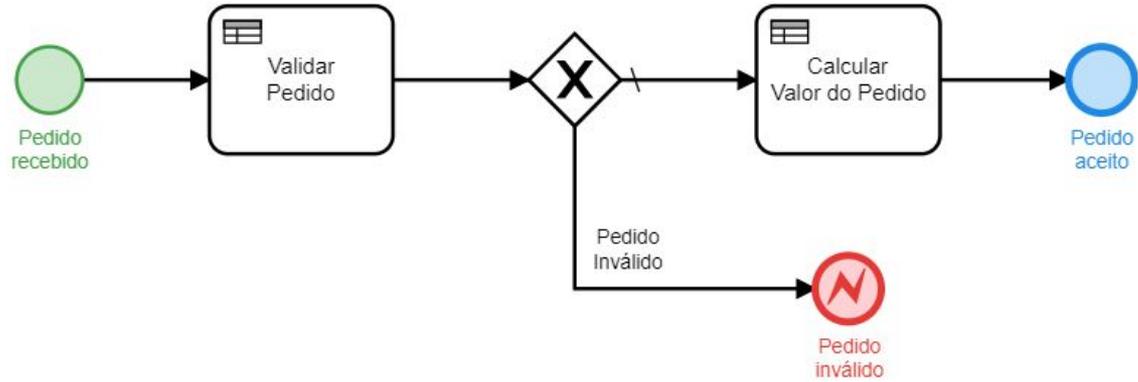


Diagrama em BPMN 2.0

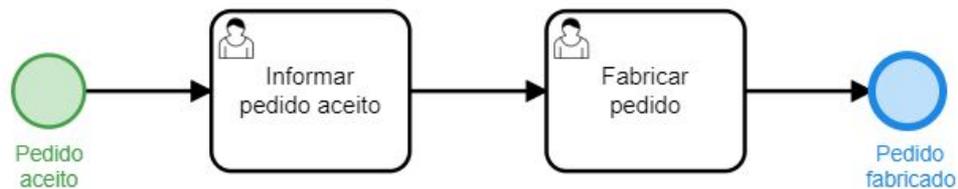
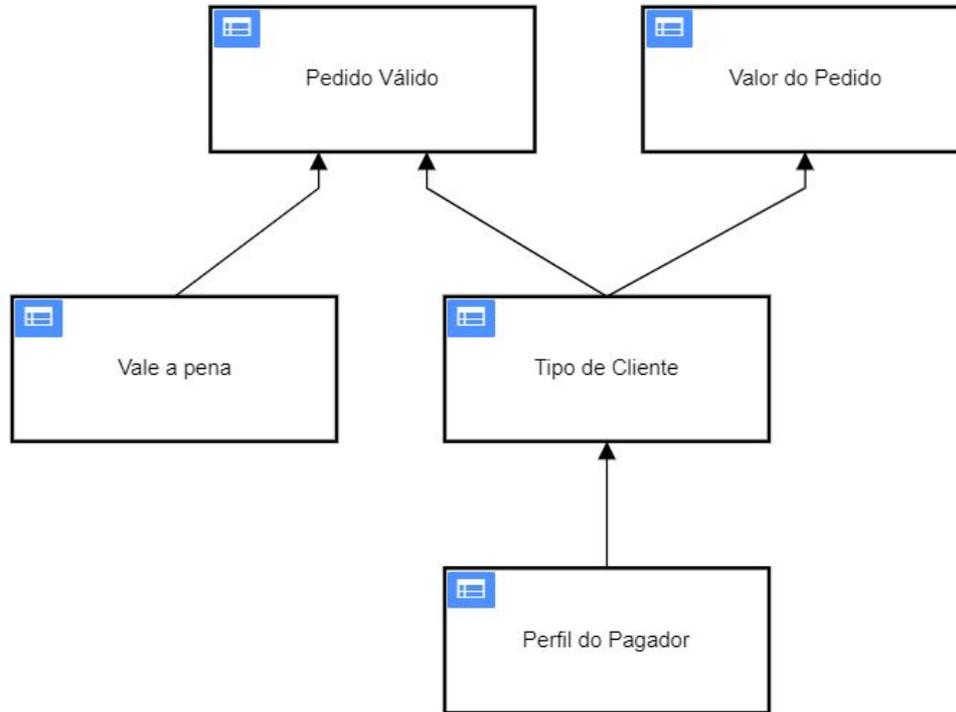


Diagrama em BPMN 2.0



Diagrama em DMN 1.3



Tabelas de Decisão em DMN

Pedido Válido		Hit Policy: Unique	
When	And	Then	And
Tipo Cliente	Vale a pena	Pedido Válido	Mensagem
string	boolean	boolean	string
1 "VIP"	-	true	"Pedido de cliente vip"
2 "Comum"	true	true	"Pedido de cliente comum"
3 "Comum"	false	false	"Pedido não vale a pena"
4 "Ruim"	-	false	"Pedido de cliente ruim"
+ -	-	-	-

Valor do Pedido		Hit Policy: Unique	
When	Then		
Tipo do Cliente	Valor do Pedido	Annotations	
string	double		
1 "VIP"	1.30 * quantidadePedida		
2 "Comum"	1.60 * quantidadePedida		
+ -	-		

Vale a pena		Hit Policy: Unique	
When	And	Then	And
Quantidade pedida	Prazo Pedido	Vale a pena	Annotations
integer	integer	boolean	
1 <100	-	false	
2 >1000	-	false	
3 [100..1000]	<1	false	
4 [100..1000]	>=1	true	
+ -	-	-	-

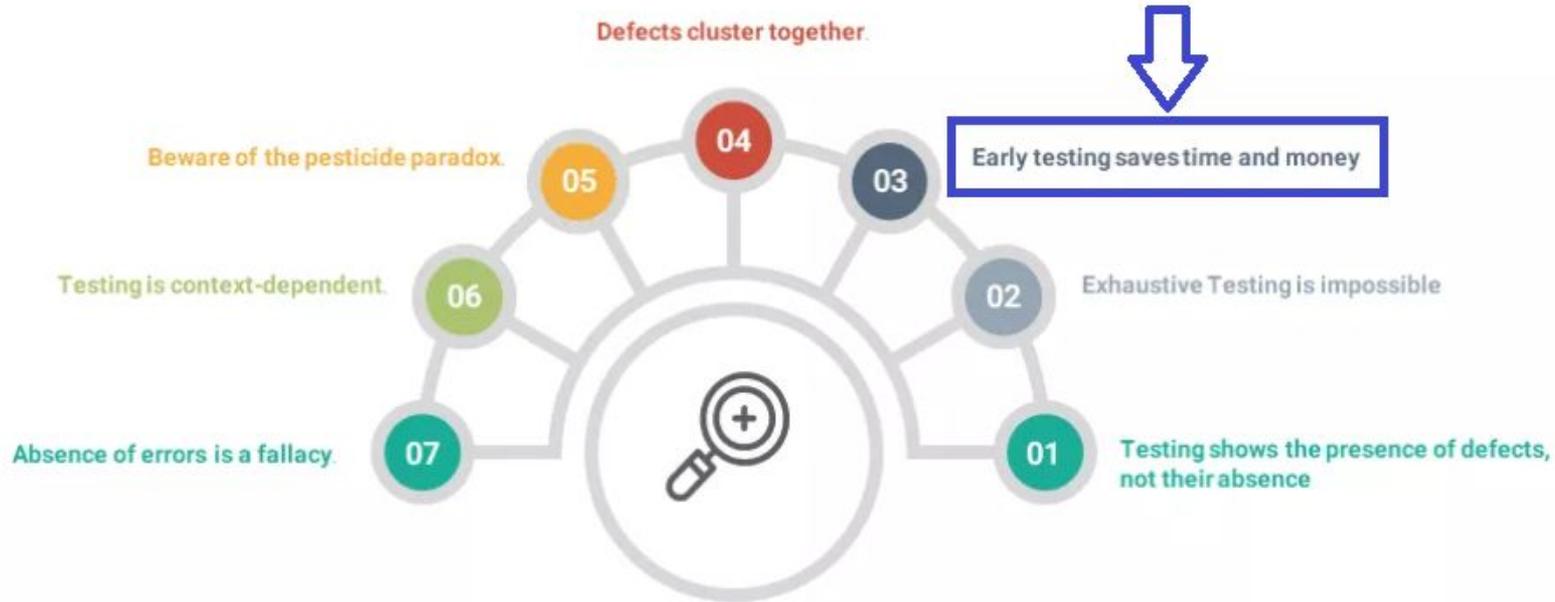
Tipo de Cliente		Hit Policy: First	
When	And	Then	And
Nome Cliente	Perfil Pagador	Indicação	Tipo do Cliente
string	string	boolean	string
1 "Creusa"	-	-	"VIP"
2 -	"Bom pagador"	-	"Comum"
3 -	"Desconhecido"	true	"Comum"
4 -	"Desconhecido"	false	"Ruim"
5 -	"Caloteiro"	-	"Ruim"
+ -	-	-	-

Perfil do Pagador		Hit Policy: Unique	
When	And	Then	And
Já comprou	Pagou	Perfil do Pagador	Annotations
boolean	boolean	string	
1 true	true	"Bom pagador"	
2 true	false	"Caloteiro"	
3 false	-	"Desconhecido"	
+ -	-	-	-

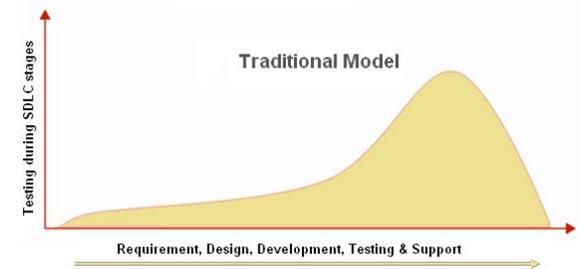
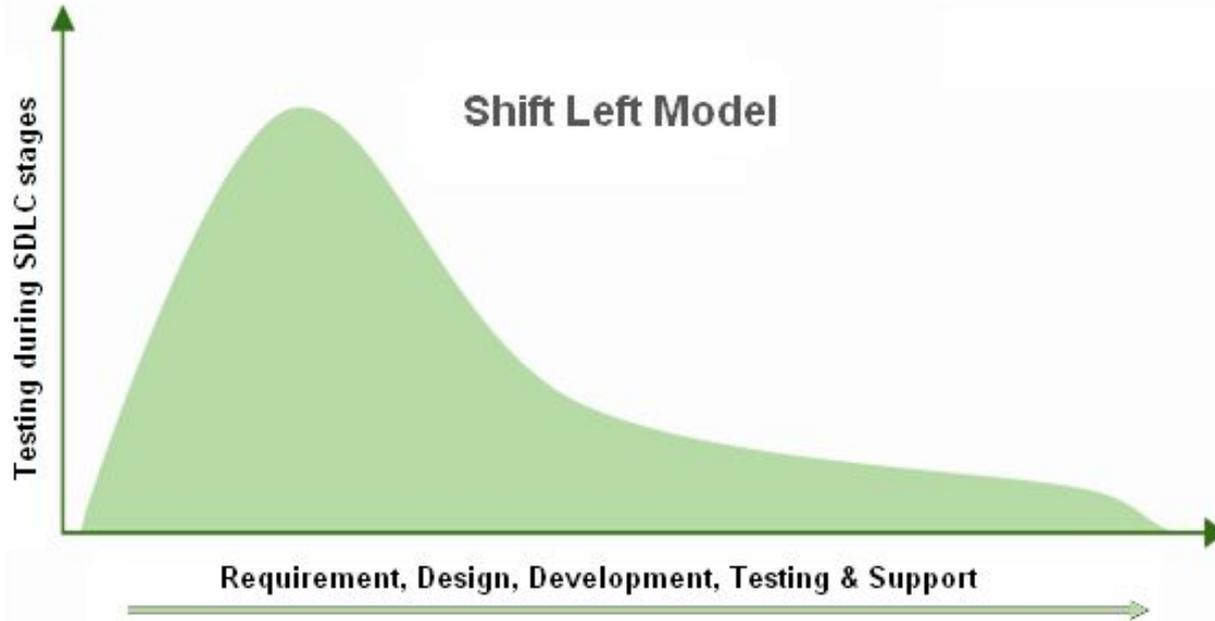
Testes no processo Negociar Pedido

- Em que momento testamos ?
- Como partimos dos requisitos e chegamos aos testes ?
- De quem é a responsabilidade pelos testes ?

Os sete princípios de teste



Antecipar os testes



DevOps Blog
Shift Left Testing: What, Why & How To Shift Left



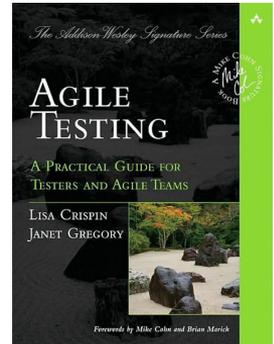
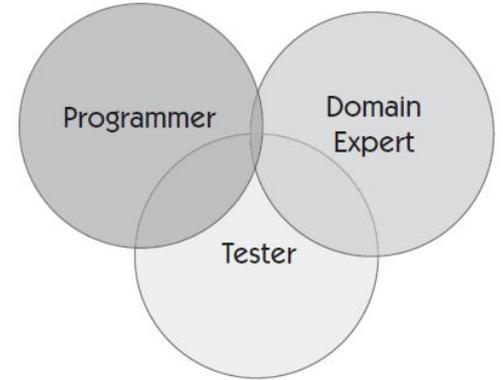
November 30, 2020
5 min read
Kirstie Magowan



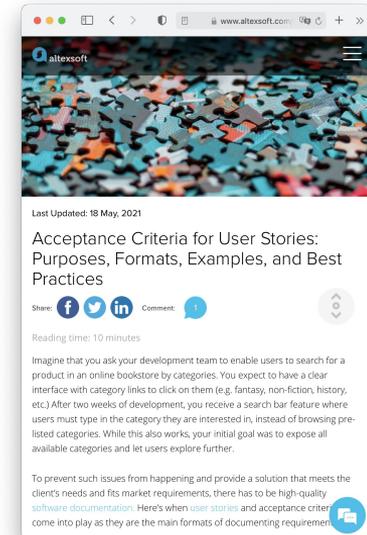
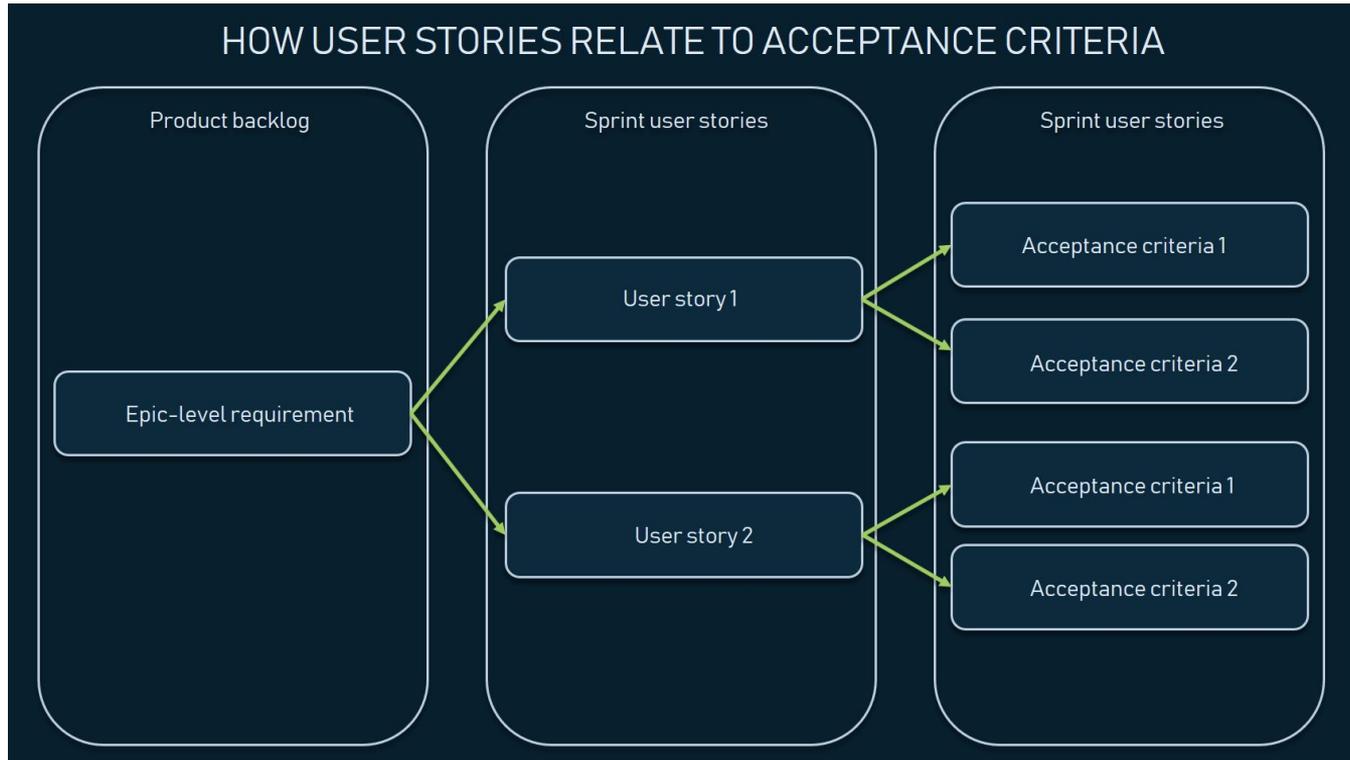
Os “três amigos”



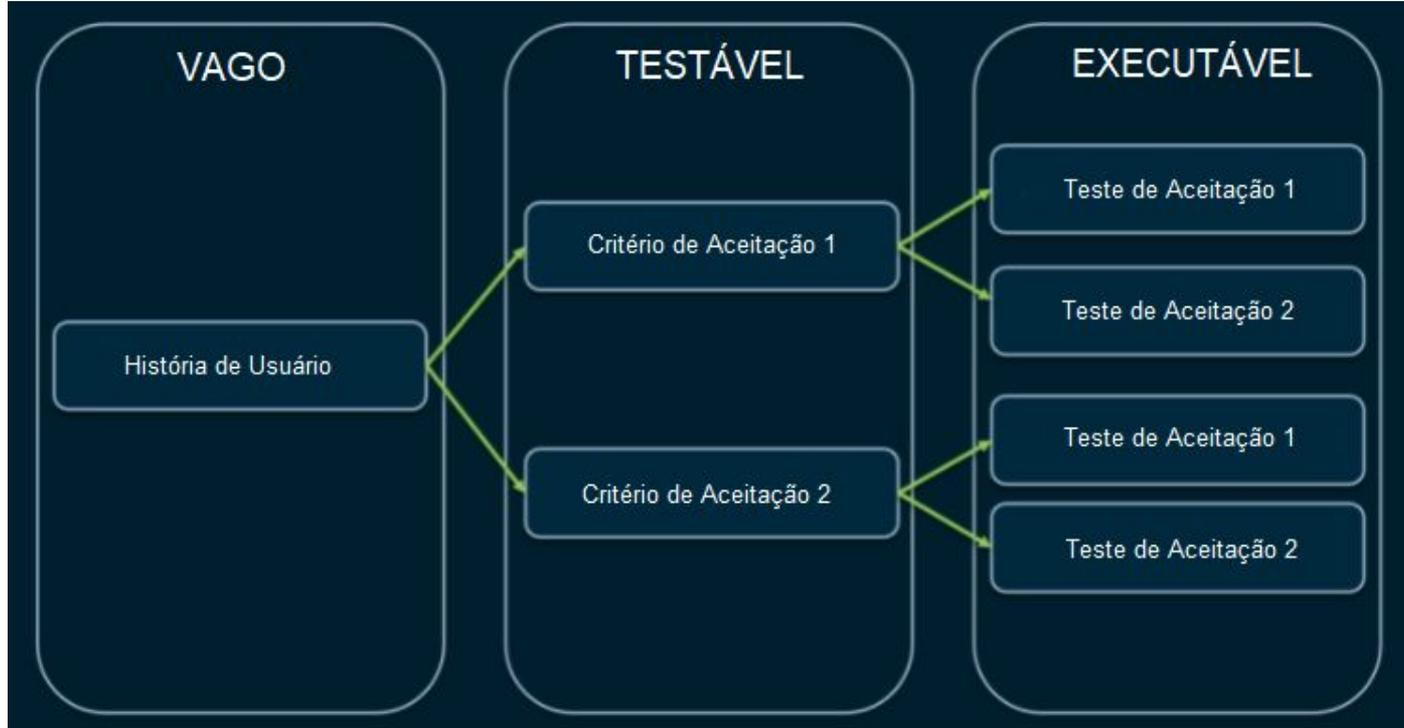
Interaction of Roles



Critérios de aceitação



Testes de aceitação



Especificações executáveis

Texto citado:

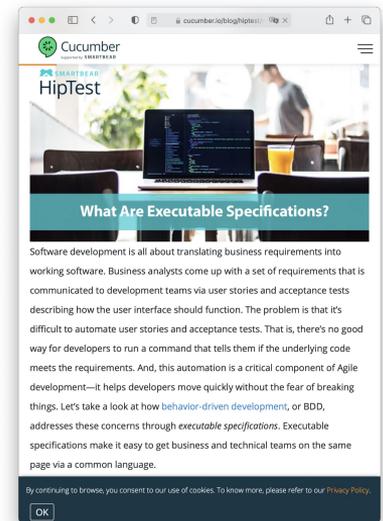
Executable specifications make it easy to get business and technical teams on the same page via a common language.

Gherkin is the most commonly used plain-text language for writing executable specifications in *Given-When-Then-And-But* steps.

Tradução livre:

Especificações executáveis facilitam colocar as equipes técnicas e de negócios na mesma página por meio de uma linguagem comum.

Gherkin é a linguagem de texto simples mais comumente usada para escrever especificações executáveis em etapas *Dado-Quando-Então-E-Mas*.



<https://cucumber.io/blog/hiptest/what-are-executable-specifications/>

Especificações executáveis

Texto citado:

The **Given-When-Then** formula is a template intended to guide the writing of [acceptance tests](#) for a [User Story](#):

- **(Given)** some context
- **(When)** some action is carried out
- **(Then)** a particular set of **observable consequences** should obtain

Tradução livre:

A fórmula **Dado-Quando-Então** é um modelo destinado a orientar a escrita de [testes de aceitação](#) para uma [História de Usuário](#):

(Dado) algum contexto

(Quando) alguma ação é realizada

(Então) um determinado conjunto de **consequências observáveis** deve obter



Definition

The Given-When-Then formula is a template intended to guide the writing of acceptance tests for a User Story:

- (Given) some context
- (When) some action is carried out
- (Then) a particular set of observable consequences should obtain

An example:

- Given my bank account is in credit, and I made no withdrawals recently,
- When I attempt to withdraw an amount less than my card's limit,
- Then the withdrawal should complete without errors or warnings

Tools such as JBehave, RSpec or Cucumber encourage use of this template, though it can also be used purely as a heuristic irrespective of any tool.

Gherkin

Texto citado:

The primary keywords are:

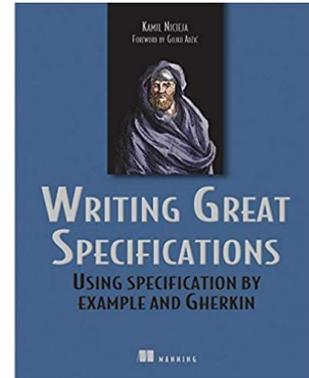
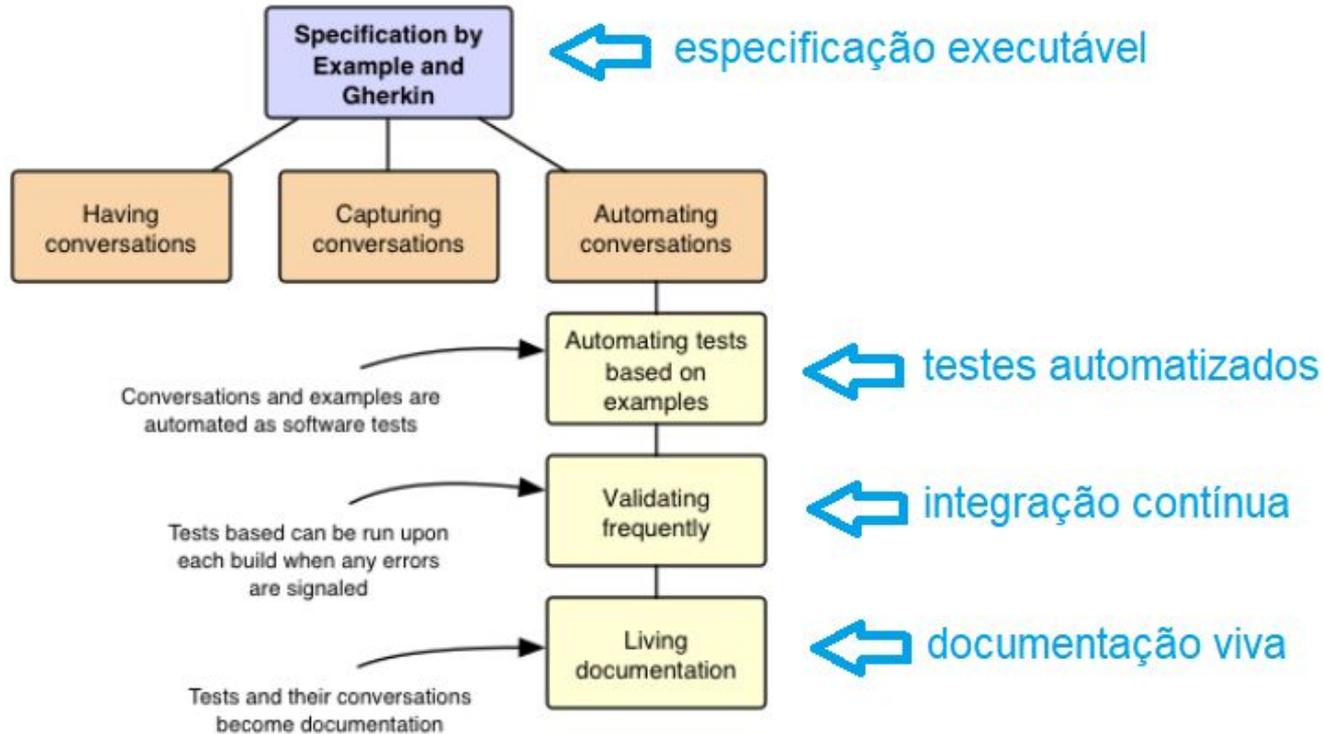
- **Feature**
- **Rule** (as of Gherkin 6)
- **Example** (or **Scenario**)
- **Given**, **When**, **Then**, **And**, **But** for steps (or *****)
- **Background**
- **Scenario Outline** (or **Scenario Template**)
- **Examples** (or **Scenarios**)

Tradução livre:

As palavras-chave primárias são:

- **Funcionalidade**
- **Regra** (a partir de Gherkin 6)
- **Exemplo** (or **Cenário**)
- **Dado**, **Quando**, **Então**, **E**, **Mas** para passos (ou *****)
- **Contexto**
- **Esquema de Cenário**
- **Exemplos** (or **Cenários**)

Documentação viva



THE TESTING Manifesto

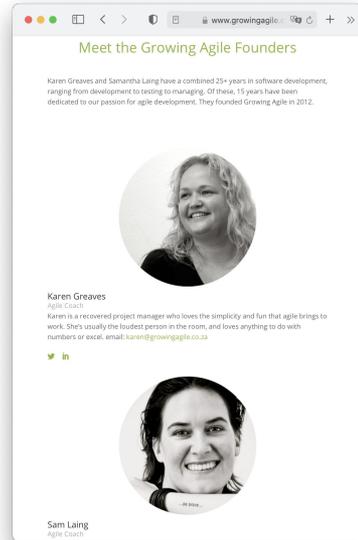


we value:

- Testing throughout **OVER** testing at the end
- Preventing bugs **OVER** finding bugs
- Testing understanding **OVER** checking functionality
- Building the best system **OVER** breaking the system
- Team responsibility for quality **OVER** tester responsibility

www.growingagile.co.za

@growingagile



Testes no processo Negociar Pedido

- Como será o planejamento dos testes ?
 - quantos casos de teste são necessários ?
- Como será a execução dos testes ?
 - com um simulador (Camunda Simulator, etc.) ?
 - com um cliente REST (Postman, Insomnia, etc.) ?
 - com uma ferramenta de teste ?

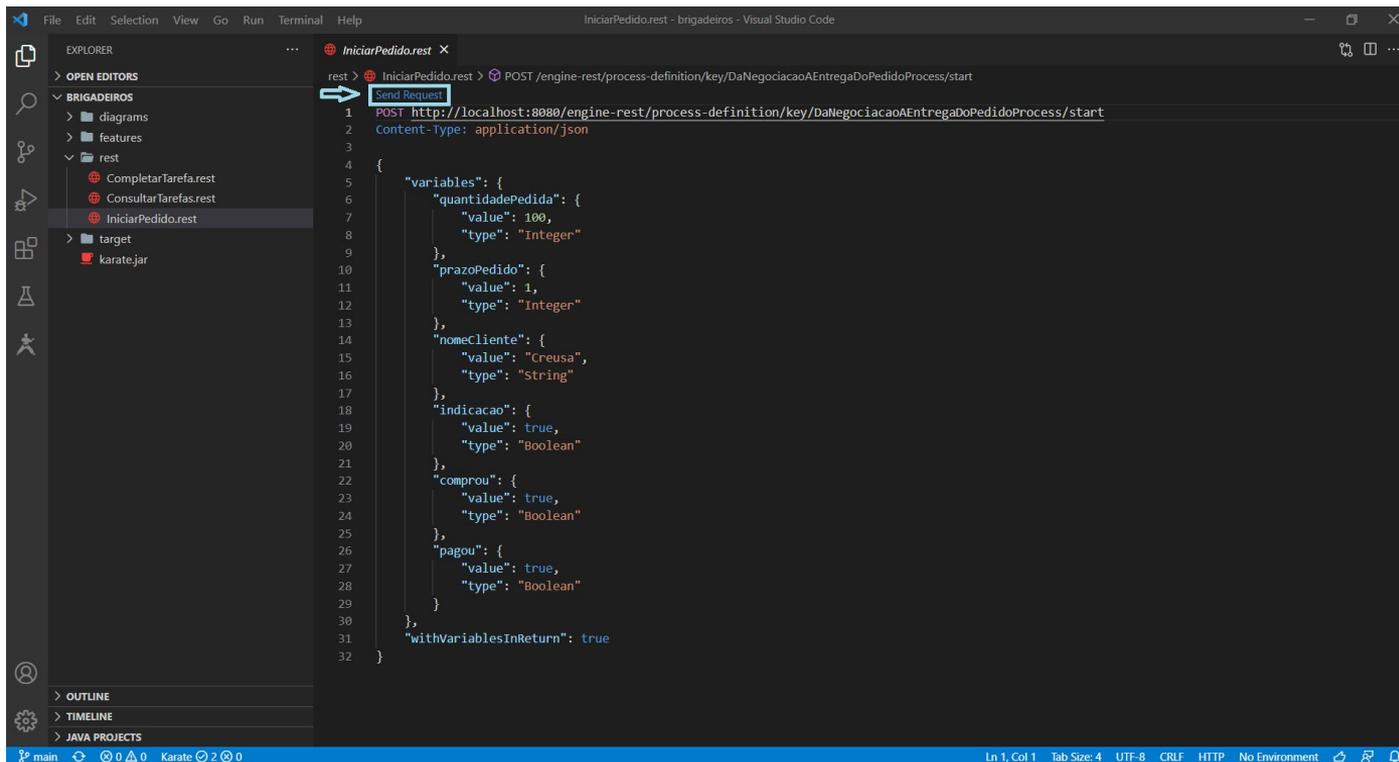
Simulação das Tabelas de Decisão em DMN

The screenshot shows the DMN Simulator interface. On the left, under 'Inputs', there are several fields: 'Decision table' (Pedido Válido), 'Vale a pena' (Quantity: 500, Deadline: 2), 'Tipo de Cliente' (Name: Creusa, Indication: True), and 'Perfil do Pagador' (Already bought: True, Paid: True). A 'Simulate now' button is located below the inputs. On the right, under 'Outputs', the results are displayed: 'Pedido Válido: true', 'Mensagem: Pedido aceito', 'Vale a pena: true', 'Tipo do Cliente: VIP', and 'Perfil do Pagador: Bom pagador'. Below the simulator is a 'View DRD' section showing a decision table for 'Pedido Válido' with a 'Hit Policy' of 'Unique'. The table has columns for 'When', 'And', 'Then', and 'And', with a final 'Annotations' column. Two rows are visible, with the first row highlighted in blue.

	When	And	Then	And	Annotations
1	"VIP"	-	true	"Pedido aceito"	
2	"Comum"	true	true	"Pedido aceito"	

← conferência visual

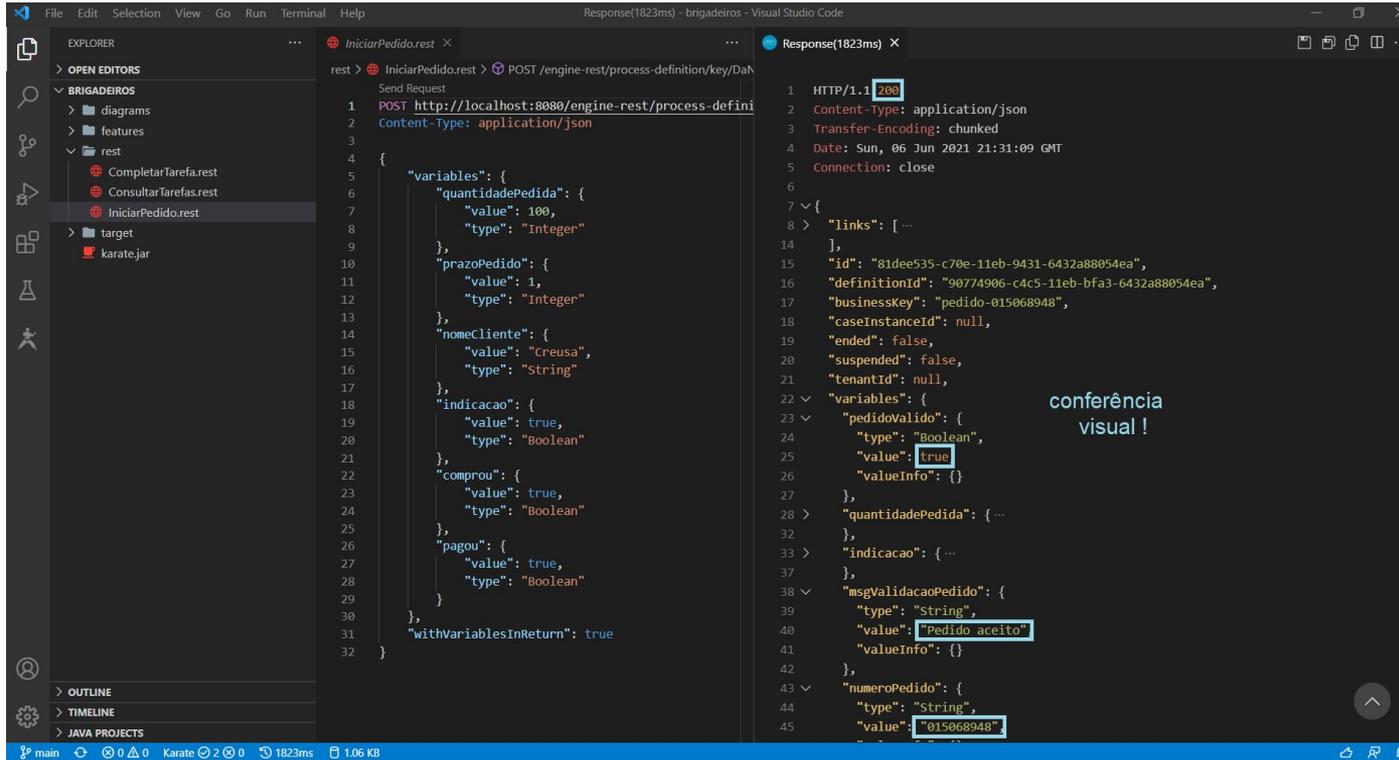
Cliente REST



The screenshot shows the Visual Studio Code interface with a REST client configuration. The Explorer on the left shows a project named 'BRIGADEIROS' with a file named 'IniciarPedido.rest'. The main editor displays the configuration for a POST request to 'http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start'. The request body is a JSON object with the following structure:

```
1 POST http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start
2 Content-Type: application/json
3
4 {
5   "variables": {
6     "quantidadePedida": {
7       "value": 100,
8       "type": "Integer"
9     },
10    "prazoPedido": {
11      "value": 1,
12      "type": "Integer"
13    },
14    "nomeCliente": {
15      "value": "Creusa",
16      "type": "String"
17    },
18    "indicacao": {
19      "value": true,
20      "type": "Boolean"
21    },
22    "comprou": {
23      "value": true,
24      "type": "Boolean"
25    },
26    "pagou": {
27      "value": true,
28      "type": "Boolean"
29    }
30  },
31  "withVariablesInReturn": true
32 }
```

Cliente REST



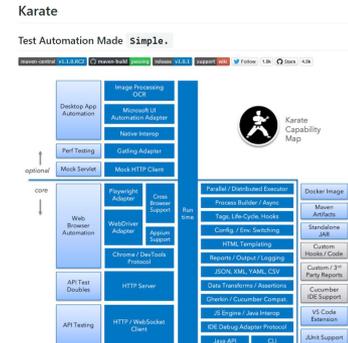
```
rest > IniciarPedido.rest > POST /engine-rest/process-definition/key/DaH
Send Request
1 POST http://localhost:8080/engine-rest/process-defini
Content-Type: application/json
2
3
4
5 {
6   "variables": {
7     "quantidadePedida": {
8       "value": 100,
9       "type": "Integer"
10    },
11    "prazoPedido": {
12      "value": 1,
13      "type": "Integer"
14    },
15    "nomeCliente": {
16      "value": "Creusa",
17      "type": "String"
18    },
19    "indicacao": {
20      "value": true,
21      "type": "Boolean"
22    },
23    "comprou": {
24      "value": true,
25      "type": "Boolean"
26    },
27    "pagou": {
28      "value": true,
29      "type": "Boolean"
30    }
31  },
32  "withVariablesInReturn": true
}

Response(1823ms) X
1 HTTP/1.1 200
2 Content-Type: application/json
3 Transfer-Encoding: chunked
4 Date: Sun, 06 Jun 2021 21:31:09 GMT
5 Connection: close
6
7 {
8   "links": [ ...
14  ],
15   "id": "81dee535-c70e-11eb-9431-6432a88054ea",
16   "definitionId": "90774906-c4c5-11eb-bfa3-6432a88054ea",
17   "businessKey": "pedido-015068948",
18   "caseInstanceId": null,
19   "ended": false,
20   "suspended": false,
21   "tenantId": null,
22   "variables": {
23     "pedidoValidado": {
24       "type": "Boolean",
25       "value": true
26     },
27     "quantidadePedida": { ...
32   },
33   "indicacao": { ...
37   },
38   "msgValidacaoPedido": {
39     "type": "String",
40     "value": "Pedido aceito"
41   },
42   },
43   "numeroPedido": {
44     "type": "String",
45     "value": "015068948"
46   }
47 }
```

conferência visual !

Karate Framework

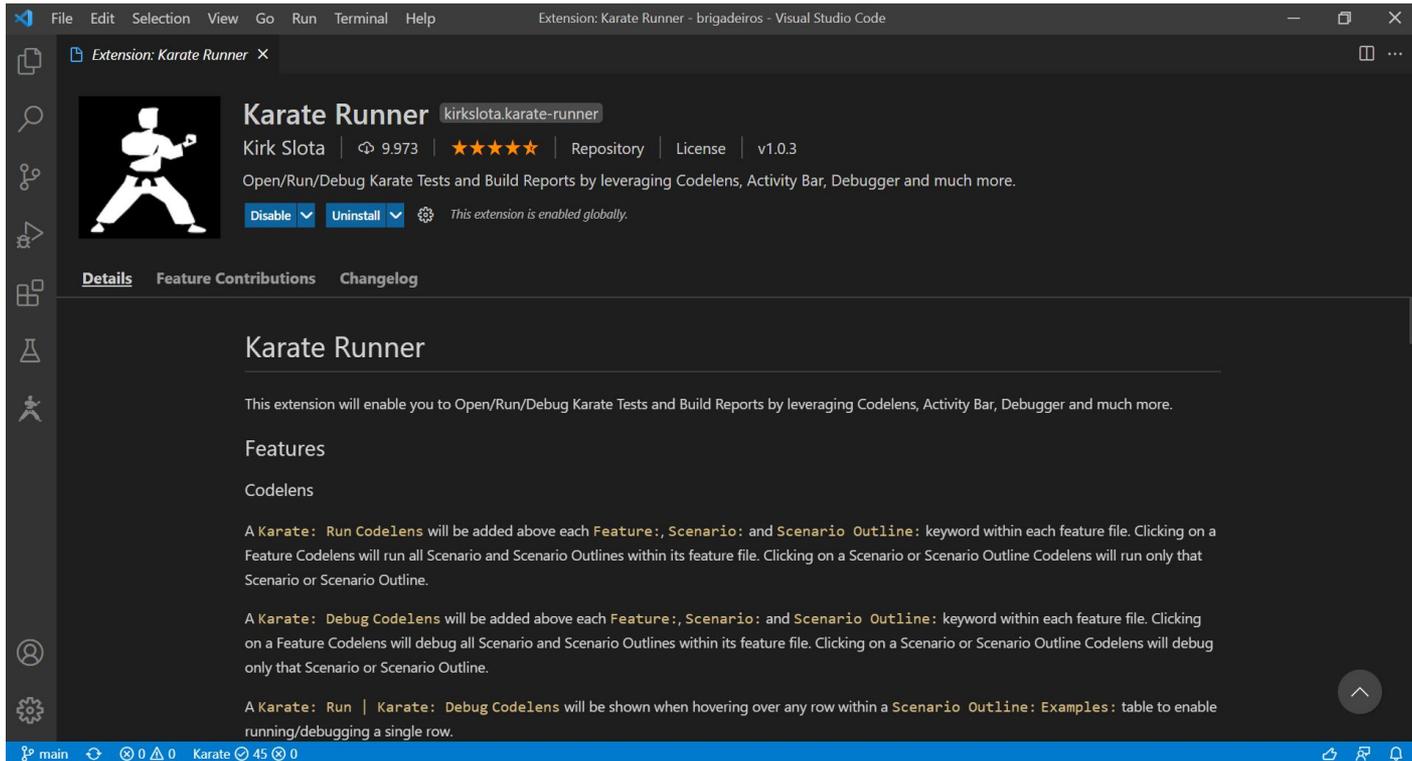
- **Based** on the popular [Cucumber](#) / [Gherkin](#) standard
- Elegant [DSL](#) syntax 'natively' supports JSON and XML
- Embedded [JavaScript](#) engine
- [Java](#) knowledge is not required and even non-programmers can write tests
- Option to use as a light-weight stand-alone executable - convenient for teams not comfortable with [Java](#)



Testes no processo Negociar Pedido

- Karate DSL (standalone)
- VS Code (extensão Karate Runner)

Visual Studio Code (extensão Karate Runner)



The screenshot displays the Visual Studio Code interface with the 'Karate Runner' extension page open. The extension is by 'kirkslota.karate-runner' and has a rating of 9.973 stars. It is currently installed and enabled globally. The page includes a description of the extension's functionality, a list of features, and a section on Codelens. The status bar at the bottom shows the current file is 'main' and the extension is running on 'Karate' with 45 tests.

File Edit Selection View Go Run Terminal Help Extension: Karate Runner - brigadeiros - Visual Studio Code

Extension: Karate Runner X

Karate Runner kirkslota.karate-runner

Kirk Slota | 9.973 | ★★★★★ | Repository | License | v1.0.3

Open/Run/Debug Karate Tests and Build Reports by leveraging Codelens, Activity Bar, Debugger and much more.

Disable Uninstall This extension is enabled globally.

Details Feature Contributions Changelog

Karate Runner

This extension will enable you to Open/Run/Debug Karate Tests and Build Reports by leveraging Codelens, Activity Bar, Debugger and much more.

Features

Codelens

A **Karate: Run Codelens** will be added above each **Feature:**, **Scenario:** and **Scenario Outline:** keyword within each feature file. Clicking on a Feature Codelens will run all Scenario and Scenario Outlines within its feature file. Clicking on a Scenario or Scenario Outline Codelens will run only that Scenario or Scenario Outline.

A **Karate: Debug Codelens** will be added above each **Feature:**, **Scenario:** and **Scenario Outline:** keyword within each feature file. Clicking on a Feature Codelens will debug all Scenario and Scenario Outlines within its feature file. Clicking on a Scenario or Scenario Outline Codelens will debug only that Scenario or Scenario Outline.

A **Karate: Run** | **Karate: Debug Codelens** will be shown when hovering over any row within a **Scenario Outline: Examples:** table to enable running/debugging a single row.

main Karate 45

Registrar Pedido (Scenario)

The screenshot shows the Visual Studio Code interface with a Karate feature file open. The feature file content is as follows:

```
features > RegistrarPedido.feature
Karate: Run | Karate: Debug
1 Feature: Registrar Pedido
2
3 > Background: ...
Karate: Run | Karate: Debug
6 > Scenario: Registrar Pedido de cliente vip ...
Karate: Run | Karate: Debug
28 > Scenario: Registrar Pedido de cliente comum ...
Karate: Run | Karate: Debug
50 > Scenario: Registrar Pedido que não vale a pena ...
Karate: Run | Karate: Debug
72 > Scenario: Registrar Pedido de cliente ruim ...
```

Below the code editor, a table titled "Pedido Válido" is displayed. The table has a "Hit Policy" of "Unique" and contains the following data:

	When	And	Then	And
	Tipo Cliente	Vale a pena	Pedido Válido	Mensagem
	string	boolean	boolean	string
1	"VIP"	-	true	"Pedido de cliente vip"
2	"Comum"	true	true	"Pedido de cliente comum"
3	"Comum"	false	false	"Pedido não vale a pena"
4	"Ruim"	-	false	"Pedido de cliente ruim"
+	-	-	-	-

The status bar at the bottom indicates the current position is Ln 72, Col 47, with 4 spaces, UTF-8 encoding, and CRLF line endings.

Registrar Pedido (Scenario)

The image shows a screenshot of the Visual Studio Code editor with a Karate feature file open. The file is named `RegistrarPedido.feature` and is located in the `features` directory. The editor displays the following code:

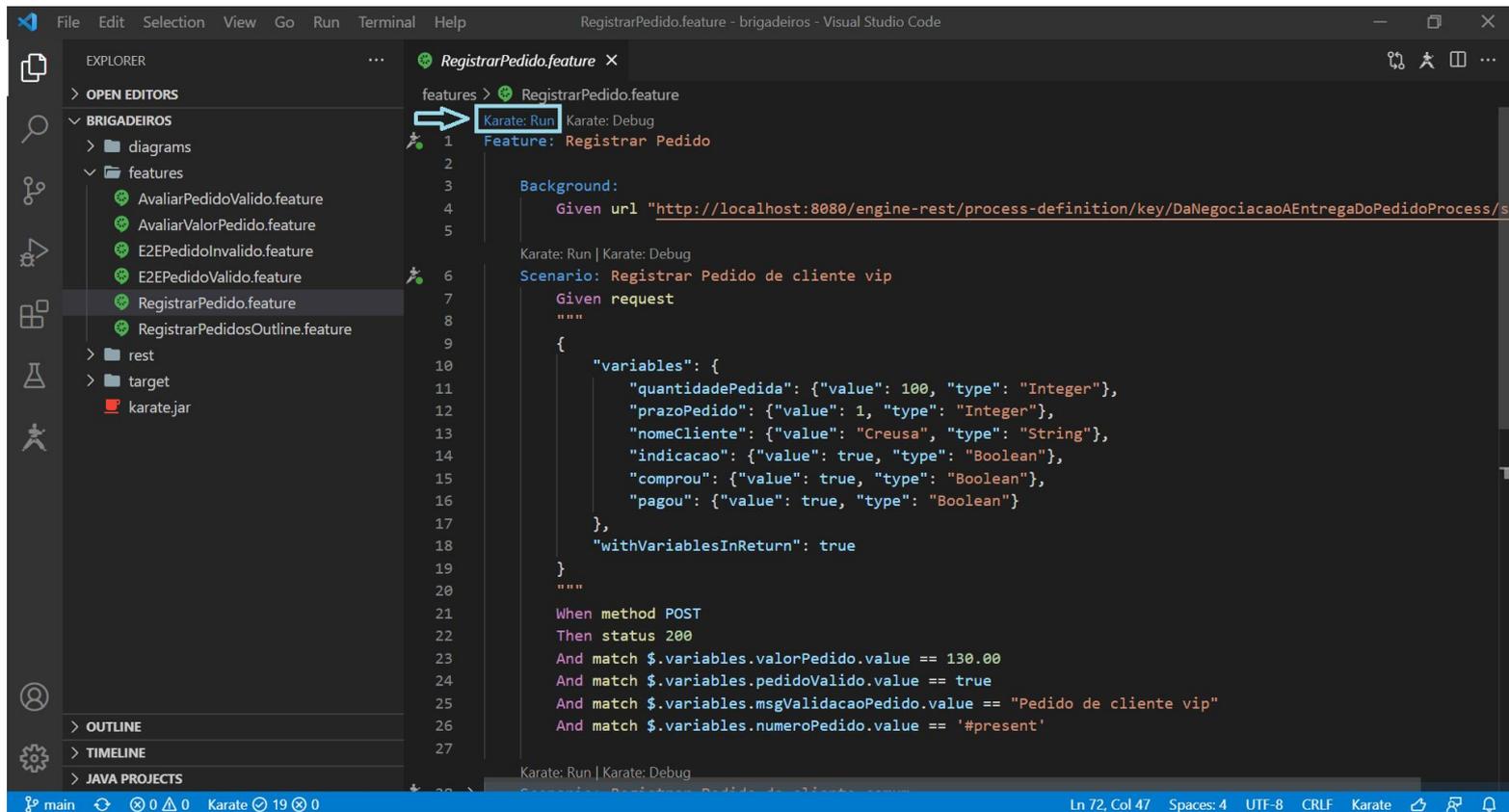
```
1 Feature: Registrar Pedido
2
3
4 Background:
5   Given url "http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess"
6
7 Scenario: Registrar Pedido de cliente vip
8   Given request
9     """
10    {
11      "variables": {
12        "quantidadePedida": {"value": 100, "type": "Integer"},
13        "prazoPedido": {"value": 1, "type": "Integer"},
14        "nomeCliente": {"value": "Creusa", "type": "String"},
15        "indicacao": {"value": true, "type": "Boolean"},
16        "comprou": {"value": true, "type": "Boolean"},
17        "pagou": {"value": true, "type": "Boolean"}
18      },
19      "withVariablesInReturn": true
20    }
21    """
22   When method POST
23   Then status 200
24   And match $.variables.valorPedido.value == 130.00
25   And match $.variables.pedidoValido.value == true
26   And match $.variables.msgValidacaoPedido.value == "Pedido de cliente vip"
27   And match $.variables.numeroPedido.value == '#present'
```

Annotations on the right side of the editor indicate the structure of the scenario:

- An arrow points to the `Given url` line, labeled "situação inicial".
- An arrow points to the `When method POST` line, labeled "ação a ser testada".
- An arrow points to the `Then status 200` and `And match` lines, labeled "resultado esperado".

The status bar at the bottom shows the current cursor position: `Ln 72, Col 47`, `Spaces: 4`, `UTF-8`, `CRLF`, and `Karate`.

Registrar Pedido (Scenario)



The image shows a screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project structure with a folder named 'BRIGADEIROS' containing a 'features' subfolder. The 'RegistrarPedido.feature' file is selected and highlighted. A white arrow points from the Explorer to the file name in the editor's title bar. The main editor area displays the content of 'RegistrarPedido.feature', which is a Karate feature file. The file starts with 'Feature: Registrar Pedido' and includes a 'Background' section with a 'Given' clause for a URL. The main 'Scenario: Registrar Pedido de cliente vip' section begins with a 'Given request' block containing a JSON object with variables for 'quantidadePedida', 'prazoPedido', 'nomeCliente', 'indicacao', 'comprou', and 'pagou'. The scenario continues with a 'When method POST' clause, followed by several 'Then' and 'And match' clauses that verify the response status and specific fields.

```
features > RegistrarPedido.feature
Karate: Run | Karate: Debug
1 Feature: Registrar Pedido
2
3
4 Background:
5   Given url "http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/s
6
7 Karate: Run | Karate: Debug
8 Scenario: Registrar Pedido de cliente vip
9   Given request
10    ""
11    {
12      "variables": {
13        "quantidadePedida": {"value": 100, "type": "Integer"},
14        "prazoPedido": {"value": 1, "type": "Integer"},
15        "nomeCliente": {"value": "Creusa", "type": "String"},
16        "indicacao": {"value": true, "type": "Boolean"},
17        "comprou": {"value": true, "type": "Boolean"},
18        "pagou": {"value": true, "type": "Boolean"}
19      },
20      "withVariablesInReturn": true
21    }
22    ""
23   When method POST
24   Then status 200
25   And match $.variables.valorPedido.value == 130.00
26   And match $.variables.pedidoValido.value == true
27   And match $.variables.msgValidacaoPedido.value == "Pedido de cliente vip"
28   And match $.variables.numeroPedido.value == '#present'
```

Registrar Pedido (Scenario)

The screenshot shows the Visual Studio Code interface with the following components:

- Explorer:** Shows the project structure with folders for 'diagrams' and 'features'. The 'RegistrarPedido.feature' file is selected.
- Terminal:** Displays the output of a Karate test run. It shows connection details for a REST client, followed by a summary of test results: 'feature: features/RegistrarPedido.feature', 'scenarios: 4 | passed: 4 | failed: 0 | time: 1,7748'. A detailed summary table is highlighted with a blue box:

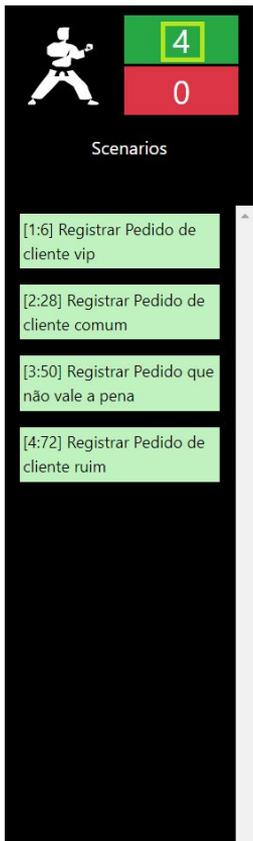
Karate version: 1.1.0.RC1			
=====			
elapsed:	3,63	threads:	1
features:	1	skipped:	0
scenarios:	4	passed:	4
		failed:	0
=====			

Below the table, a link to the HTML report is provided: [HTML report: \(paste into browser to view\) | Karate version: 1.1.0.RC1](#). The full path to the report is: `file:///C:/Users/brigadeiros/.vscode/extensions/com.intuit.karatelabs/automated-testing-of-bpmn-and-dmn-diagrams/brigadeiros/target/karate-reports/karate-summary.html`.

The terminal also shows the message: 'Terminal will be reused by tasks, press any key to close it.'

Page-Footer: Ln 72, Col 47 Spaces: 4 UTF-8 CRLF Karate

Registrar Pedido (Scenario)



Scenários

- [1:6] Registrar Pedido de cliente vip
- [2:28] Registrar Pedido de cliente comum
- [3:50] Registrar Pedido que não vale a pena
- [4:72] Registrar Pedido de cliente ruim

Summary Tags Feature: features/RegistrarPedido.feature Registrar Pedido	
Scenario: [1:6] Registrar Pedido de cliente vip ms: 1215	
>> Background:	
7 Given request	31
<pre>{ "variables": { "quantidadePedido": {"value": 100, "type": "Integer"}, "prazoPedido": {"value": 1, "type": "Integer"}, "nomeCliente": {"value": "Creusa", "type": "String"}, "indicacao": {"value": true, "type": "Boolean"}, "comprou": {"value": true, "type": "Boolean"}, "pagou": {"value": true, "type": "Boolean"} }, "withVariablesInReturn": true }</pre>	
21 When method POST	1075
22 Then status 200	0
23 And match \$.variables.valorPedido.value == 130.00	16
24 And match \$.variables.pedidoValido.value == true	1
25 And match \$.variables.msgValidacaoPedido.value == "Pedido de cliente vip"	2
26 And match \$.variables.numeroPedido.value == '#present'	2
Scenario: [2:28] Registrar Pedido de cliente comum ms: 367	
>> Background:	
29 Given request	1
43 When method POST	363
44 Then status 200	0
45 And match \$.variables.valorPedido.value == 160.00	1
46 And match \$.variables.pedidoValido.value == true	0
47 And match \$.variables.msgValidacaoPedido.value == "Pedido de cliente comum"	1
48 And match \$.variables.numeroPedido.value == '#present'	0

Registrar Pedido (Scenario)



4
0

Scenários

- [1:6] Registrar Pedido de cliente vip
- [2:28] Registrar Pedido de cliente comum
- [3:50] Registrar Pedido que não vale a pena
- [4:72] Registrar Pedido de cliente ruim

26	And match \$.variables.numeroPedido.value == '#present'	2
Scenario: [2:28] Registrar Pedido de cliente comum		ms: 367
>>	Background:	
29	Given request	1
43	When method POST	363
44	Then status 200	0
45	And match \$.variables.valorPedido.value == 160.00	1
46	And match \$.variables.pedidoValido.value == true	0
47	And match \$.variables.msgValidacaoPedido.value == "Pedido de cliente comum"	1
48	And match \$.variables.numeroPedido.value == '#present'	0
Scenario: [3:50] Registrar Pedido que não vale a pena		ms: 137
>>	Background:	
51	Given request	1
65	When method POST	131
66	Then status 200	0
67	And match \$.variables.valorPedido.value == '#notpresent'	2
68	And match \$.variables.pedidoValido.value == false	1
69	And match \$.variables.msgValidacaoPedido.value == "Pedido não vale a pena"	1
70	And match \$.variables.numeroPedido.value == '#present'	0
Scenario: [4:72] Registrar Pedido de cliente ruim		ms: 121
>>	Background:	
73	Given request	0
87	When method POST	118
88	Then status 200	0
89	And match \$.variables.valorPedido.value == '#notpresent'	0
90	And match \$.variables.pedidoValido.value == false	0
91	And match \$.variables.msgValidacaoPedido.value == "Pedido de cliente ruim"	1
92	And match \$.variables.numeroPedido.value == '#present'	1

Registrar Pedidos (Scenario Outline)

The image shows a screenshot of the Visual Studio Code editor with a Cucumber feature file open. The file is named `RegistrarPedidosOutline.feature` and is located in the `features` directory. The editor displays the following content:

```
1 Feature: Registrar Pedidos
2
3 > Background: ...
4
5 Karate: Run | Karate: Debug
6 Scenario Outline: Registrar Pedidos com sucesso
7   Given path "/process-definicao/Key/DaNegociacaoAEntregaDoPedidoProcess/start"
8   Given request
9     """
10    { ...
11    }
12    """
13
14 When method POST
15 Then status 200
16
17 And match $.variables.valorPedido.value == <valorPedido>
18 And match $.variables.pedidoValido.value == <pedidoValido>
19 And match $.variables.msgValidacaoPedido.value == <msgValidacaoPedido>
20 And match $.variables.numeroPedido.value == '#present'
```

Annotations in the image:

- A red box highlights the `Feature: Registrar Pedidos` line.
- A red box highlights the `Scenario Outline: Registrar Pedidos com sucesso` line, with an arrow pointing to the text "Scenario Outline".
- A red box highlights the variable placeholders `<valorPedido>`, `<pedidoValido>`, and `<msgValidacaoPedido>` in the `And match` lines, with an arrow pointing to the text "variáveis".
- A red box highlights the `Examples: Pedidos de cliente vip...` line, with an arrow pointing to the text "Examples".

The status bar at the bottom shows the current position: `Ln 1, Col 19 Spaces: 4 UTF-8 CRLF Karate`.

Registrar Pedidos (Scenario Outline)

```
File Edit Selection View Go Run Terminal Help RegistrarPedidosOutline.feature - brigadeiros - Visual Studio Code
RegistrarPedidosOutline.feature
features > RegistrarPedidosOutline.feature
29 Examples: Pedidos de cliente vip
30 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido | valorPedido
31 | 99 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" | 128.70
32 | 100 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" | 130.00
33 | 999 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" | 1298.70
34 | 1001 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" | 1301.30
35 | 99 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" | 128.70
36 | 100 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" | 130.00
37 | 999 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" | 1298.70
38 | 1001 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" | 1301.30
39
40 Examples: Pedidos de cliente comum
41 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido | valorPedido
42 | 100 | 1 | Maria | true | false | false | true | "Pedido de cliente comum" | 160.00
43 | 999 | 1 | Maria | true | false | false | true | "Pedido de cliente comum" | 1598.40
44 | 100 | 1 | Maria | false | true | true | true | "Pedido de cliente comum" | 160.00
45 | 999 | 1 | Maria | false | true | true | true | "Pedido de cliente comum" | 1598.40
46
47 Examples: Pedidos que não valem a pena
48 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido | valorPedido
49 | 100 | 0 | Penna | true | true | true | false | "Pedido não vale a pena" | '#notpresent'
50 | 999 | 0 | Penna | true | true | true | false | "Pedido não vale a pena" | '#notpresent'
51 | 99 | 1 | Penna | true | true | true | false | "Pedido não vale a pena" | '#notpresent'
52 | 1001 | 1 | Penna | true | true | true | false | "Pedido não vale a pena" | '#notpresent'
53
54 Examples: Pedidos de cliente ruim
55 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido | valorPedido
56 | 100 | 1 | Frias | true | true | false | false | "Pedido de cliente ruim" | '#notpresent'
57 | 100 | 1 | Frias | false | true | false | false | "Pedido de cliente ruim" | '#notpresent'
58 | 100 | 1 | Frias | false | false | false | false | "Pedido de cliente ruim" | '#notpresent'
59
main 0 0 Karate 4 0 Ln 1, Col 19 Spaces: 4 UTF-8 CRLF Karate
```

Registrar Pedidos (Scenario Outline)

The screenshot displays the Visual Studio Code interface with the following components:

- EXPLORER:** Shows a project structure under 'BRIGADEIROS' with folders 'diagrams', 'features', 'rest', and 'target'. The 'features' folder is expanded, showing several feature files, with 'RegistrarPedidosOutline.feature' selected.
- EDITOR:** Displays the content of 'RegistrarPedidosOutline.feature'. The file starts with a 'Feature: Registrar Pedidos' and a 'Scenario Outline: Registrar Pedidos com sucesso'. The scenario outline includes a 'Given' block with request variables and a 'When method POST' block with status and match assertions. A red arrow points to the 'Karate: Run' button in the editor's gutter.
- STATUS BAR:** Shows 'Ln 3, Col 16', 'Spaces: 4', 'UTF-8', 'CRLF', 'Karate', and other editor settings.

```
features > RegistrarPedidosOutline.feature
1 Karate: Run Karate: Debug
2 Feature: Registrar Pedidos
3
4 Background:
5   Given url "http://localhost:8080/engine-rest"
6
7 Karate: Run | Karate: Debug
8 Scenario Outline: Registrar Pedidos com sucesso
9   Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"
10  Given request
11  """
12  {
13    "variables": {
14      "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
15      "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
16      "nomeCliente": {"value": <nomeCliente>, "type": "String"},
17      "indicacao": {"value": <indicacao>, "type": "Boolean"},
18      "comprou": {"value": <comprou>, "type": "Boolean"},
19      "pagou": {"value": <pagou>, "type": "Boolean"}
20    },
21    "withVariablesInReturn": true
22  }
23  """
24  When method POST
25  Then status 200
26  And match $.variables.valorPedido.value == <valorPedido>
27  And match $.variables.pedidoValido.value == <pedidoValido>
28  And match $.variables.msgValidacaoPedido.value == <msgValidacaoPedido>
29  And match $.variables.numeroPedido.value == '#present'
```

Registrar Pedidos (Scenario Outline)



19

0

Scenários

- [1.1:31] Registrar Pedidos com sucesso
- [1.2:32] Registrar Pedidos com sucesso
- [1.3:33] Registrar Pedidos com sucesso
- [1.4:34] Registrar Pedidos com sucesso
- [1.5:35] Registrar Pedidos com sucesso
- [1.6:36] Registrar Pedidos com sucesso
- [1.7:37] Registrar Pedidos com sucesso
- [1.8:38] Registrar Pedidos com sucesso

Summary Tags Feature: <i>features/RegistrarPedidosOutline.feature</i> Registrar Pedidos	
Scenario: [1.1:31] Registrar Pedidos com sucesso ms: 1558	
>> Background:	
7 Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"	14
8 Given request	37
<pre>{ "variables": { "quantidadePedida": {"value": 99 "type": "Integer"}, "prazoPedido": {"value": 0 "type": "Integer"}, "nomeCliente": {"value": "Creusa" "type": "String"}, "indicacao": {"value": false "type": "Boolean"}, "comprou": {"value": false "type": "Boolean"}, "pagou": {"value": false "type": "Boolean"} }, "withVariablesInReturn": true }</pre>	
22 When method POST	1385
23 Then status 200	0
24 And match \$.variables.valorPedido.value == 128.70	13
25 And match \$.variables.pedidoValido.value == true	1
26 And match \$.variables.msgValidacaoPedido.value == "Pedido aceito"	1
27 And match \$.variables.numeroPedido.value == '#present'	1
Scenario: [1.2:32] Registrar Pedidos com sucesso ms: 147	
>> Background:	
7 Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"	1
8 Given request	1
22 When method POST	144

Avaliar Pedido (Scenario Outline)

The screenshot displays the Visual Studio Code editor with a Cucumber feature file named `AvaliarPedidoValido.feature` open. The Explorer sidebar on the left shows the project structure, including a `features` folder containing several feature files. The main editor area shows the following content:

```
features > AvaliarPedidoValido.feature
Karate: Run | Karate: Debug
1 Feature: Avaliar Pedido
2
3 > Background: ...
Karate: Run | Karate: Debug
6 Scenario Outline: Avaliar Pedidos com sucesso
7   Given path "/decision-definition/key/PedidoValidoDecision/evaluate"
8   Given request
9     """
10    {
11      "variables": {
12        "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
13        "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
14        "nomeCliente": {"value": <nomeCliente>, "type": "String"},
15        "indicacao": {"value": <indicacao>, "type": "Boolean"},
16        "comprou": {"value": <comprou>, "type": "Boolean"},
17        "pagou": {"value": <pagou>, "type": "Boolean"}
18      }
19    }
20    """
21   When method POST
22   Then status 200
23   And match $[0].pedidoValido.value == <pedidoValido>
24   And match $[0].msgValidacaoPedido.value == <msgValidacaoPedido>
25
26 > Examples: Pedidos válidos ...
41 > Examples: Pedidos inválidos ...
```

The status bar at the bottom indicates the current position: `Ln 6, Col 51`, `Spaces: 4`, `UTF-8`, `CRLF`, `Karate`, and shows 4 test runs.

Avaliar Pedido (Scenario Outline)

```
File Edit Selection View Go Run Terminal Help AvaliarPedidoValido.feature - brigadeiros - Visual Studio Code
```

```
AvaliarPedidoValido.feature X
```

```
features > AvaliarPedidoValido.feature
```

```
21 When method POST
22 Then status 200
23 And match ${0}.pedidoValido.value == <pedidoValido>
24 And match ${0}.msgValidacaoPedido.value == <msgValidacaoPedido>
25
26 Examples: Pedidos válidos
27 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido |
28 | 99 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
29 | 100 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
30 | 999 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
31 | 1001 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
32 | 99 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
33 | 100 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
34 | 999 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
35 | 1001 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
36 | 100 | 1 | Maria | true | false | false | true | "Pedido de cliente comum" |
37 | 999 | 1 | Maria | true | false | false | true | "Pedido de cliente comum" |
38 | 100 | 1 | Maria | false | true | true | true | "Pedido de cliente comum" |
39 | 999 | 1 | Maria | false | true | true | true | "Pedido de cliente comum" |
40
41 Examples: Pedidos inválidos
42 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido |
43 | 100 | 0 | Penna | true | true | true | false | "Pedido não vale a pena" |
44 | 999 | 0 | Penna | true | true | true | false | "Pedido não vale a pena" |
45 | 99 | 1 | Penna | true | true | true | false | "Pedido não vale a pena" |
46 | 1001 | 1 | Penna | true | true | true | false | "Pedido não vale a pena" |
47 | 100 | 1 | Frias | true | true | false | false | "Pedido de cliente ruim" |
48 | 100 | 1 | Frias | false | true | false | false | "Pedido de cliente ruim" |
49 | 100 | 1 | Frias | false | false | false | false | "Pedido de cliente ruim" |
50
```

```
main 0 0 Karate 7 12 Ln 50, Col 1 Spaces: 4 UTF-8 CRLF Karate
```

Avaliar Pedido (Scenario Outline)

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure under 'BRIGADEIROS' > 'features'. The file 'AvaliarPedidoValido.feature' is selected.
- EDITOR:** Displays the Karate feature file content. A red arrow points to the 'Karate: Run' button in the top left of the editor area.
- Code Content:**

```
features > AvaliarPedidoValido.feature
1 Karate: Run
2 Feature: Avaliar Pedido
3
4 Background:
5   Given url "http://localhost:8080/engine-rest"
6
7 Karate: Run | Karate: Debug
8 Scenario Outline: Avaliar Pedidos com sucesso
9   Given path "/decision-definition/key/PedidoValidoDecision/evaluate"
10  Given request
11  """
12  {
13    "variables": {
14      "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
15      "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
16      "nomeCliente": {"value": <nomeCliente>, "type": "String"},
17      "indicacao": {"value": <indicacao>, "type": "Boolean"},
18      "comprou": {"value": <comprou>, "type": "Boolean"},
19      "pagou": {"value": <pagou>, "type": "Boolean"}
20    }
21  }
22  """
23  When method POST
24  Then status 200
25  And match $[0].pedidoValido.value == <pedidoValido>
26  And match $[0].msgValidacaoPedido.value == <msgValidacaoPedido>
27
28 Examples: Pedidos válidos
29 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValid
30 | 99                | 0           | Creusa      | false     | false   | false  | true          | "Pedido
```
- STATUS BAR:** Shows 'Ln 50, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Karate', and a search icon.

Avaliar Pedido (Scenario Outline)



19
0

Scenarios

- [1.1:28] Avaliar Pedidos com sucesso
- [1.2:29] Avaliar Pedidos com sucesso
- [1.3:30] Avaliar Pedidos com sucesso
- [1.4:31] Avaliar Pedidos com sucesso
- [1.5:32] Avaliar Pedidos com sucesso
- [1.6:33] Avaliar Pedidos com sucesso
- [1.7:34] Avaliar Pedidos com sucesso
- [1.8:35] Avaliar Pedidos com sucesso

Summary | Tags | Feature: [features/AvaliarPedidoValido.feature](#) | Avaliar Pedido

Scenario: [1.1:28] Avaliar Pedidos com sucesso		ms: 1144
>>	Background:	
7	Given path "/decision-definition/key/PedidoValidoDecision/evaluate"	17
8	Given request	34
<pre>{ "variables": { "quantidadePedida": {"value": 99, "type": "Integer"}, "prazoPedido": {"value": 0, "type": "Integer"}, "nomeCliente": {"value": "Creusa", "type": "String"}, "indicacao": {"value": false, "type": "Boolean"}, "comprou": {"value": false, "type": "Boolean"}, "pagou": {"value": false, "type": "Boolean"} } }</pre>		
21	When method POST	970
22	Then status 200	0
23	And match \${0}.pedidoValido.value == true	15
24	And match \${0}.msgValidacaoPedido.value == "Pedido de cliente vip"	3

Scenario: [1.2:29] Avaliar Pedidos com sucesso		ms: 18
>>	Background:	
7	Given path "/decision-definition/key/PedidoValidoDecision/evaluate"	0
8	Given request	0
21	When method POST	17
22	Then status 200	0
23	And match \${0}.pedidoValido.value == true	0
24	And match \${0}.msgValidacaoPedido.value == "Pedido de cliente vip"	0

Avaliar Valor Pedido (Scenario Outline)

The screenshot shows the Visual Studio Code editor with the following content:

```
features > AvaliarValorPedido.feature
1 Feature: Avaliar Valor do Pedido
2
3 Background:
4   Given url "http://localhost:8080/engine-rest"
5
6 Karate: Run | Karate: Debug
7 Scenario Outline: Avaliar Valor do Pedido com sucesso
8   Given path "/decision-definition/key/ValorPedidoDecision/evaluate"
9   Given request
10  """"
11  {
12    "variables": {
13      "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
14      "prazoPedido": {"value": 1, "type": "Integer"},
15      "nomeCliente": {"value": <nomeCliente>, "type": "String"},
16      "indicacao": {"value": true, "type": "Boolean"},
17      "comprou": {"value": true, "type": "Boolean"},
18      "pagou": {"value": true, "type": "Boolean"}
19    }
20  }
21  """"
22 When method POST
23 Then status 200
24 And match response[0].valorPedido.value == <valorPedido>
25
26 Examples: Pedidos válidos
27 | nomeCliente | quantidadePedida | valorPedido |
28 | Creusa      | 100              | 130.00     |
29 | Maria       | 100              | 160.00     |
```

Red boxes in the image highlight the following elements:

- The **Karate: Run** button in the Explorer view.
- The **Scenario Outline: Avaliar Valor do Pedido com sucesso** text in the editor.
- The **Given** step in the editor.
- The **Examples: Pedidos válidos** table in the editor.

nomeCliente	quantidadePedida	valorPedido
Creusa	100	130.00
Maria	100	160.00

Avaliar Valor Pedido (Scenario Outline)



2
0

Scenários

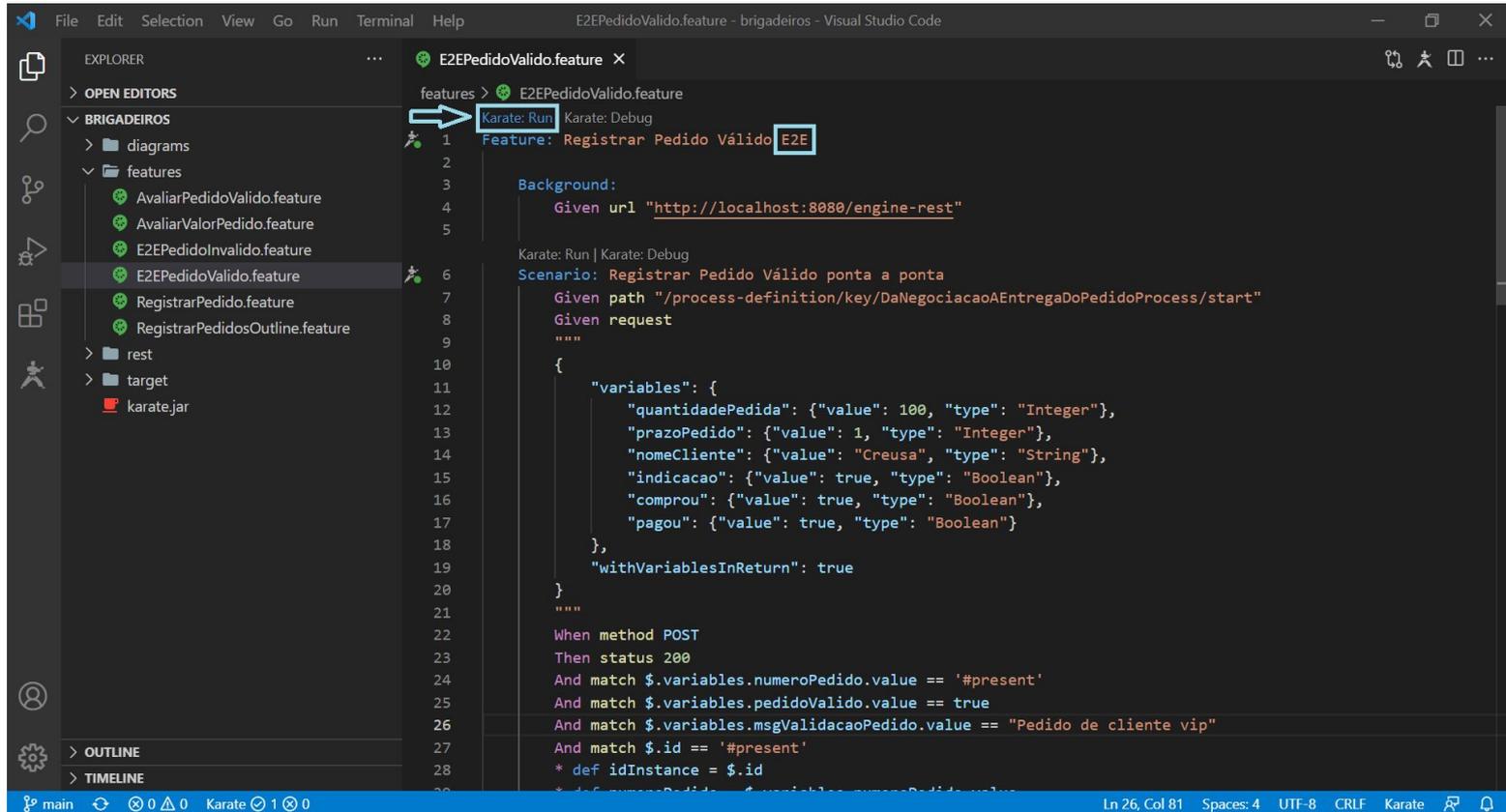
[1.1:27] Avaliar Valor do Pedido com sucesso

[1.2:28] Avaliar Valor do Pedido com sucesso

Summary | Tags | Feature: [features/AvaliarValorPedido.feature](#) | Avaliar Valor do Pedido

Scenario: [1.1:27] Avaliar Valor do Pedido com sucesso		ms: 1294
>>	Background:	
7	Given path <code>"/decision-definition/key/ValorPedidoDecision/evaluate"</code>	15
8	Given request	33
<pre>{ "variables": { "quantidadePedida": {"value": 100, "type": "Integer"}, "prazoPedido": {"value": 1, "type": "Integer"}, "nomeCliente": {"value": "Creusa", "type": "String"}, "indicacao": {"value": true, "type": "Boolean"}, "comprou": {"value": true, "type": "Boolean"}, "pagou": {"value": true, "type": "Boolean"} } }</pre>		
21	When method POST	1136
22	Then status 200	0
23	And match <code>response[0].valorPedido.value == 130.00</code>	16
Scenario: [1.2:28] Avaliar Valor do Pedido com sucesso		ms: 99
>>	Background:	
7	Given path <code>"/decision-definition/key/ValorPedidoDecision/evaluate"</code>	0
8	Given request	0
21	When method POST	97
22	Then status 200	0
23	And match <code>response[0].valorPedido.value == 160.00</code>	2

Registrar Pedido Válido (ponta-a-ponta)



The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the Editor view on the right. The Explorer view shows a project structure with folders like 'diagrams', 'features', 'rest', and 'target'. The 'features' folder is expanded, showing several feature files, with 'E2EPedidoValido.feature' selected. The Editor view shows the content of 'E2EPedidoValido.feature', which is a Karate feature file. The file contains a background section with a 'Given' clause for the URL 'http://localhost:8080/engine-rest', a scenario section for 'Registrar Pedido Válido ponta a ponta', and a 'Given' clause for a request body. The request body is a JSON object with the following structure:

```
Feature: Registrar Pedido Válido E2E

1
2
3
4 Background:
5   Given url "http://localhost:8080/engine-rest"
6
7 Karate: Run | Karate: Debug
8 Scenario: Registrar Pedido Válido ponta a ponta
9   Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"
10  Given request
11  """
12  {
13    "variables": {
14      "quantidadePedida": {"value": 100, "type": "Integer"},
15      "prazoPedido": {"value": 1, "type": "Integer"},
16      "nomeCliente": {"value": "Creusa", "type": "String"},
17      "indicacao": {"value": true, "type": "Boolean"},
18      "comprou": {"value": true, "type": "Boolean"},
19      "pagou": {"value": true, "type": "Boolean"}
20    },
21    "withVariablesInReturn": true
22  }
23  """
24  When method POST
25  Then status 200
26  And match $.variables.numeroPedido.value == '#present'
27  And match $.variables.pedidoValido.value == true
28  And match $.variables.msgValidacaoPedido.value == "Pedido de cliente vip"
29  * def idInstance = $.id
30  * def numeroPedido = $.variables.numeroPedido.value
```

The status bar at the bottom indicates the current position is Ln 26, Col 81, with 4 spaces, UTF-8 encoding, and CRLF line endings. The Karate version is 1.0.0.

Registrar Pedido Válido (ponta-a-ponta)



Scenários

[1:6] Registrar Pedido Válido ponta a ponta

Summary | Tags | Feature: *features/E2EPedidoValido.feature* | Registrar Pedido Válido E2E

Scenario: [1:6] Registrar Pedido Válido ponta a ponta		ms: 2892
>>	Background:	
7	Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"	14
8	Given request	29
22	When method POST	2428
23	Then status 200	0
24	And match \$.variables.numeroPedido.value == '#present'	13
25	And match \$.variables.pedidoValido.value == true	3
26	And match \$.variables.msgValidacaoPedido.value == "Pedido de cliente vip"	2
27	And match \$.id == '#present'	0
28	* def idInstance = \$.id	0
29	* def numeroPedido = \$.variables.numeroPedido.value	0
30	* print 'idInstance: ' + idInstance	108
31	* print 'numeroPedido: ' + numeroPedido	2
33	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	7
34	When method GET	33
35	Then status 200	0
36	And match \$[0].id == '#present'	1
37	And match \$[0].taskDefinitionKey == 'InformarPedidoAceitoTask'	1
38	* def idTask = \$[0].id	0
40	Given path "/task/" + idTask + "/complete"	2
41	And header Content-Type = 'application/json'	2
42	When method POST	15
43	Then status 204	0
45	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	0
46	When method GET	19

← Iniciar instância

← Verificar posição da instância

← Completar tarefa

Registrar Pedido Válido (ponta-a-ponta)



Scenários

[1:6] Registrar Pedido Válido ponta a ponta

40	Given path "/task/" + idTask + "/complete"	2
41	And header Content-Type = 'application/json'	2
42	When method POST	15
43	Then status 204	0
45	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	0
46	When method GET	19
47	Then status 200	0
48	And match \${0}.id == '#present'	0
49	And match \${0}.taskDefinitionKey == 'FabricarPedidoTask'	1
50	* def idTask = \${0}.id	1
52	Given path "/task/" + idTask + "/complete"	0
53	And header Content-Type = 'application/json'	0
54	When method POST	35
55	Then status 204	0
57	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	0
58	When method GET	29
59	Then status 200	0
60	And match \${0}.id == '#present'	1
61	And match \${0}.taskDefinitionKey == 'EntregarPedidoTask'	1
62	* def idTask = \${0}.id	1
64	Given path "/task/" + idTask + "/complete"	0
65	And header Content-Type = 'application/json'	0
66	When method POST	28
67	Then status 204	0
69	Given path "/history/process-instance/" + idInstance	1
70	When method GET	17
71	Then status 200	0
72	And match \$.state == 'COMPLETED'	1

← Verificar posição da instância

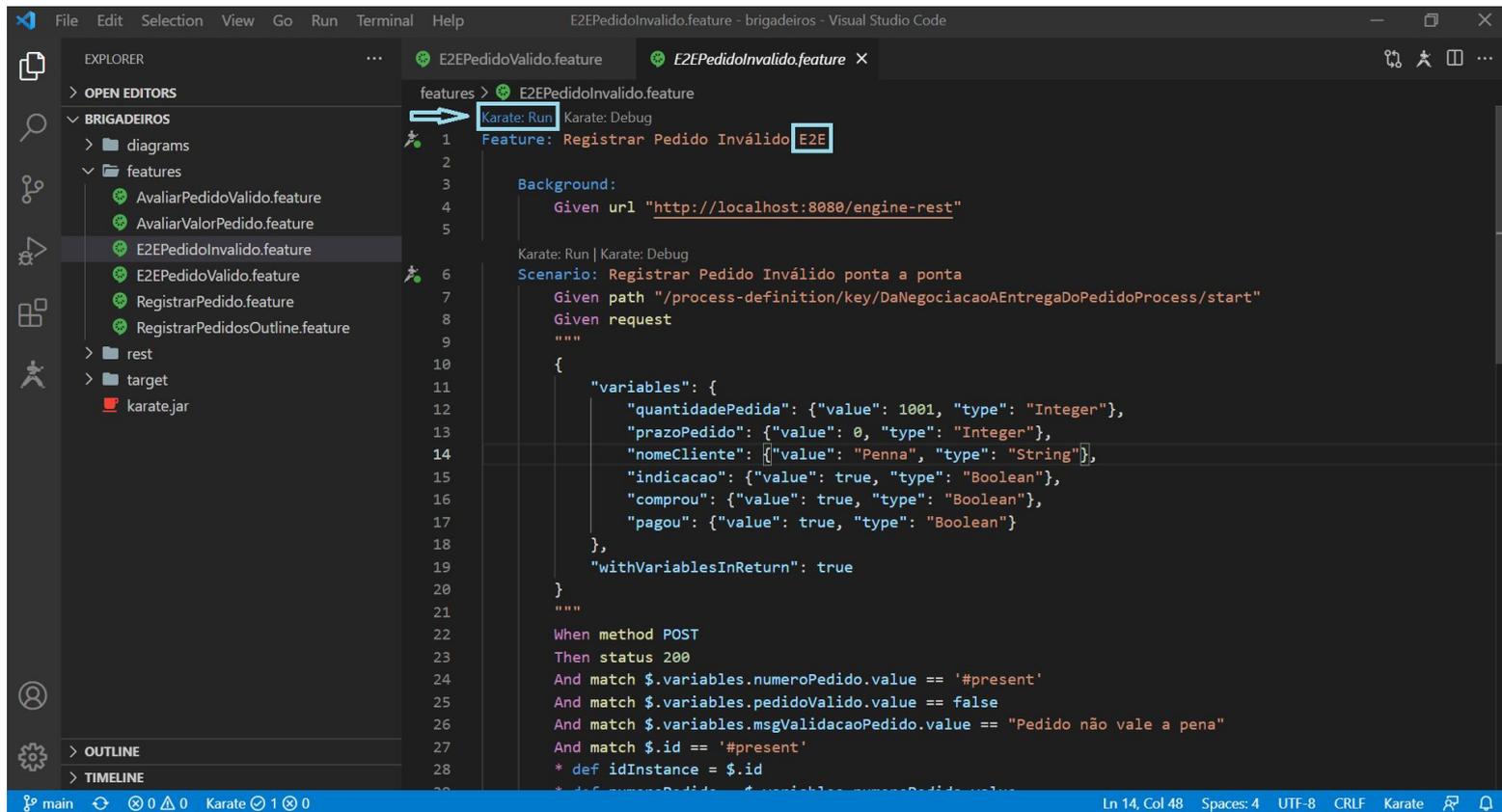
← Completar tarefa

← Verificar posição da instância

← Completar tarefa

← Verificar posição da instância

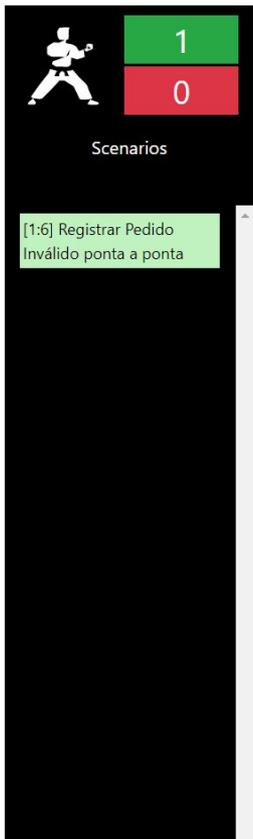
Registrar Pedido Inválido (ponta-a-ponta)



The image shows a screenshot of the Visual Studio Code editor with a Karate feature file open. The Explorer view on the left shows a project structure with folders for 'diagrams', 'features', 'rest', and 'target'. The 'features' folder is expanded, showing several feature files, with 'E2EPedidoInvalido.feature' selected. The main editor displays the content of 'E2EPedidoInvalido.feature', which includes a 'Feature: Registrar Pedido Inválido' section with an 'E2E' tag. The 'Background' section contains a 'Given url' statement. The 'Scenario: Registrar Pedido Inválido ponta a ponta' section contains 'Given' statements for path and request, followed by a JSON request body with variables for 'quantidadePedida', 'prazoPedido', 'nomeCliente', 'indicacao', and 'comprou'. The 'When' section contains a 'When method POST' statement, and the 'Then' section contains several 'And match' statements for validation. The status bar at the bottom shows 'Ln 14, Col 48' and 'Spaces: 4'.

```
features > E2EPedidoInvalido.feature
1 Karate: Run | Karate: Debug
2 Feature: Registrar Pedido Inválido E2E
3
4 Background:
5   Given url "http://localhost:8080/engine-rest"
6
7 Karate: Run | Karate: Debug
8 Scenario: Registrar Pedido Inválido ponta a ponta
9   Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"
10  Given request
11  """
12  {
13    "variables": {
14      "quantidadePedida": {"value": 1001, "type": "Integer"},
15      "prazoPedido": {"value": 0, "type": "Integer"},
16      "nomeCliente": {"value": "Penna", "type": "String"},
17      "indicacao": {"value": true, "type": "Boolean"},
18      "comprou": {"value": true, "type": "Boolean"},
19      "pagou": {"value": true, "type": "Boolean"}
20    },
21    "withVariablesInReturn": true
22  }
23  """
24  When method POST
25  Then status 200
26  And match $.variables.numeroPedido.value == '#present'
27  And match $.variables.pedidoValido.value == false
28  And match $.variables.msgValidacaoPedido.value == "Pedido não vale a pena"
29  And match $.id == '#present'
30  * def idInstance = $.id
31  * def numeroPedido = $.variables.numeroPedido.value
```

Registrar Pedido Inválido (ponta-a-ponta)



Scenários

[1:6] Registrar Pedido Inválido ponta a ponta

Summary Tags Feature: features/E2EPedidoInvalido.feature Registrar Pedido Inválido E2E		
Scenário: [1:6] Registrar Pedido Inválido ponta a ponta		
ms: 2309		
>> Background:		
7	Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"	14
8	Given request	31
22	When method POST	1765
23	Then status 200	0
24	And match \$.variables.numeroPedido.value == '#present'	13
25	And match \$.variables.pedidoValido.value == false	1
26	And match \$.variables.msgValidacaoPedido.value == "Pedido não vale a pena"	1
27	And match \$.id == '#present'	0
28	* def idInstance = \$.id	0
29	* def numeroPedido = \$.variables.numeroPedido.value	0
30	* print 'idInstance: ' + idInstance	240
31	* print 'numeroPedido: ' + numeroPedido	8
33	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	31
34	When method GET	38
35	Then status 200	0
36	And match \${0}.id == '#present'	2
37	And match \${0}.taskDefinitionKey == 'InformarPedidoCanceladoTask'	1
38	* def idTask = \${0}.id	1
40	Given path "/task/" + idTask + "/complete"	2
41	And header Content-Type = 'application/json'	1
42	When method POST	41
43	Then status 204	0
45	Given path "/history/process-instance/" + idInstance	1
46	When method GET	15
47	Then status 200	0
48	And match \$.state == 'COMPLETED'	1

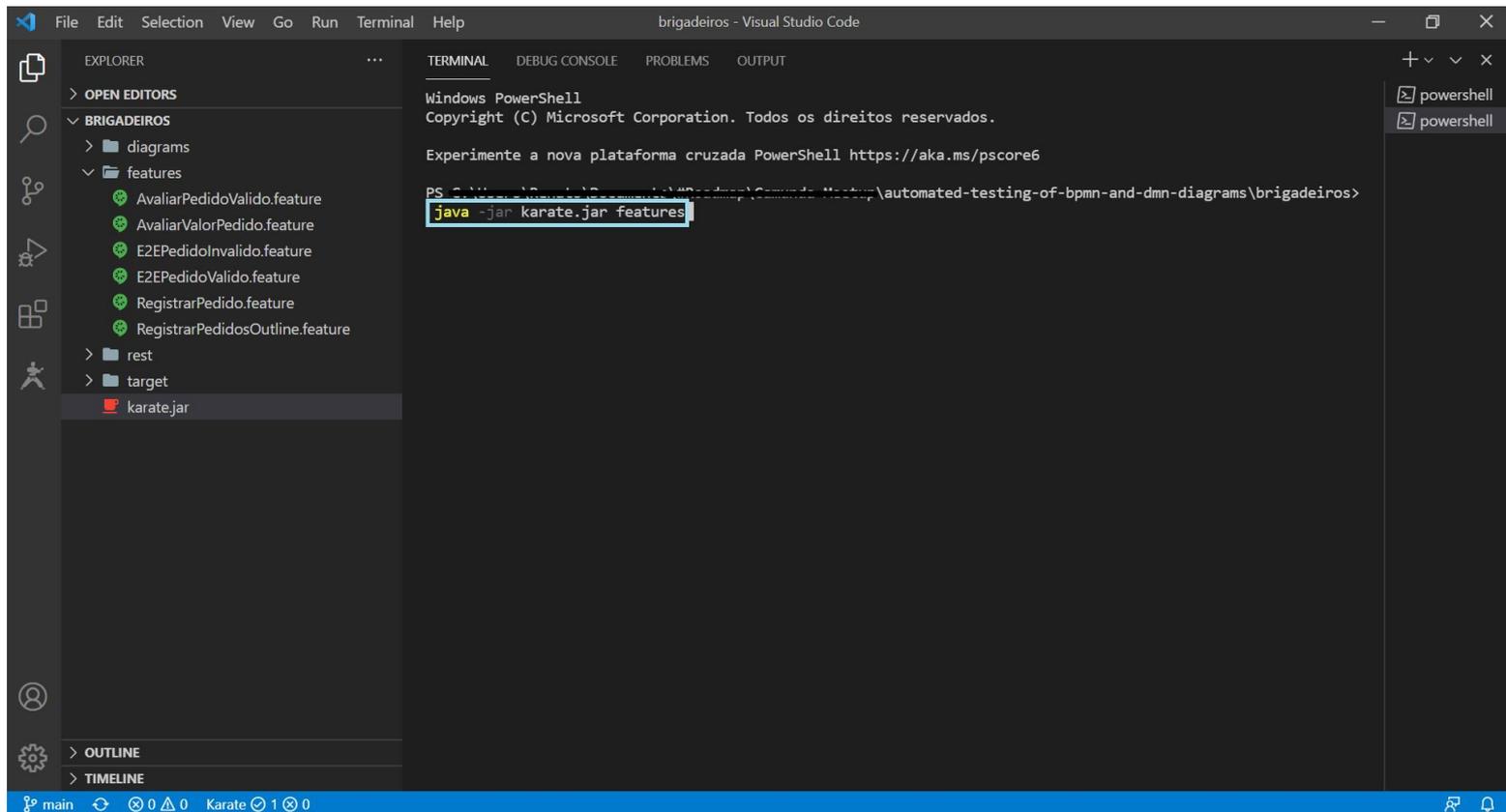
← Iniciar instância

← Verificar posição da instância

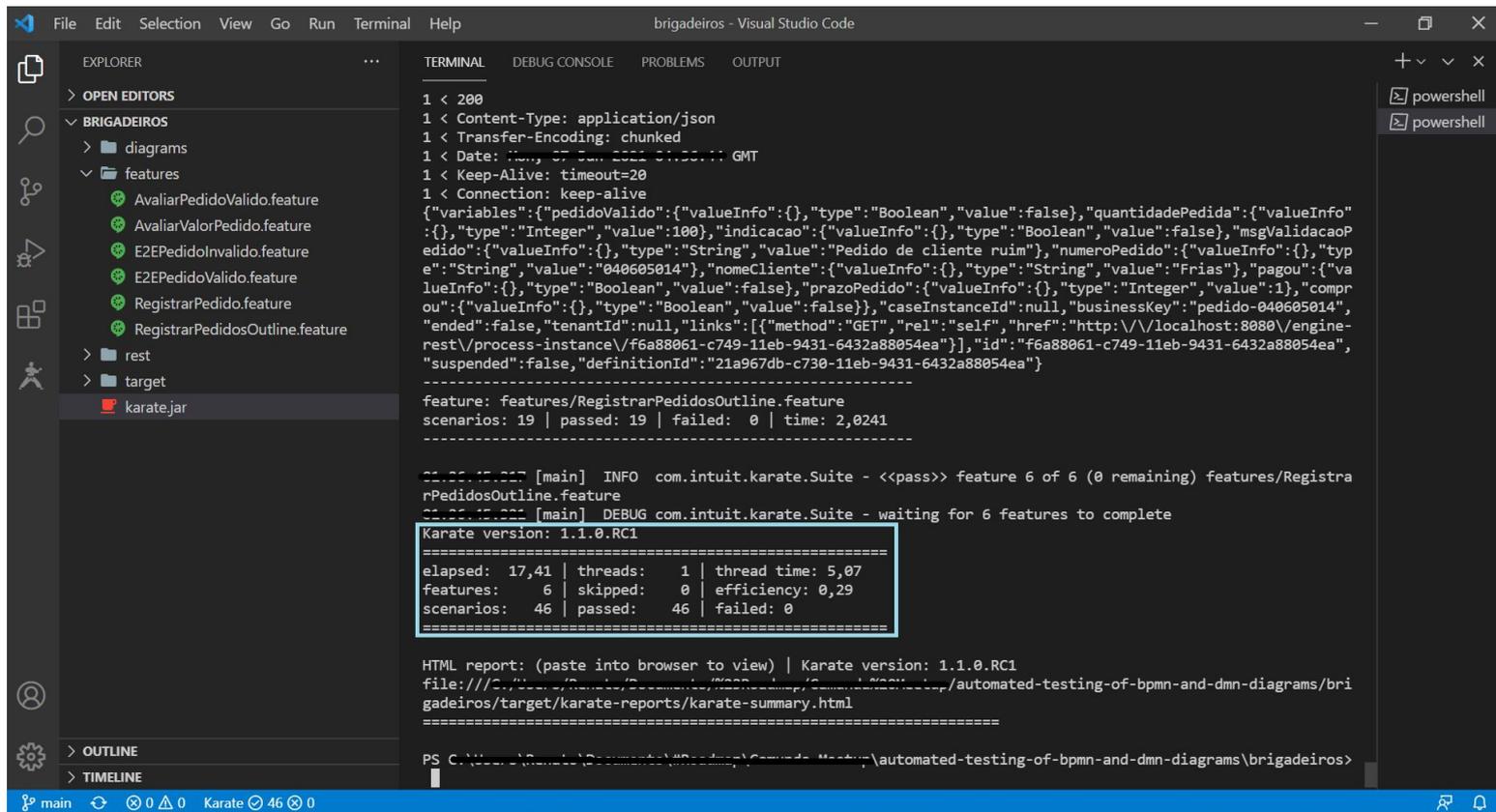
← Completar tarefa

← Verificar posição da instância

Suite de testes completa



Suite de testes completa



```
File Edit Selection View Go Run Terminal Help brigadeiros - Visual Studio Code
EXPLORER
OPEN EDITORS
BRIGADEIROS
  diagrams
  features
    AvaliarPedidoValido.feature
    AvaliarValorPedido.feature
    E2EPedidoInvalido.feature
    E2EPedidoValido.feature
    RegistrarPedido.feature
    RegistrarPedidosOutline.feature
  rest
  target
    karate.jar
TERMINAL
1 < 200
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Date: Mon, 07 Jun 2021 07:00:11 GMT
1 < Keep-Alive: timeout=20
1 < Connection: keep-alive
{"variables":{"pedidoValido":{"valueInfo":{},"type":"Boolean","value":false},"quantidadePedida":{"valueInfo":{},"type":"Integer","value":100},"indicacao":{"valueInfo":{},"type":"Boolean","value":false},"msgValidacaoPedido":{"valueInfo":{},"type":"String","value":"Pedido de cliente ruim"},"numeroPedido":{"valueInfo":{},"type":"String","value":"040605014"},"nomeCliente":{"valueInfo":{},"type":"String","value":"Frias"},"pagou":{"valueInfo":{},"type":"Boolean","value":false},"prazoPedido":{"valueInfo":{},"type":"Integer","value":1},"comprou":{"valueInfo":{},"type":"Boolean","value":false},"caseInstanceId":null,"businessKey":"pedido-040605014","ended":false,"tenantId":null,"links":[{"method":"GET","rel":"self","href":"http://localhost:8080/engine-rest/process-instance/f6a88061-c749-11eb-9431-6432a88054ea"}],"id":"f6a88061-c749-11eb-9431-6432a88054ea","suspended":false,"definitionId":"21a967db-c730-11eb-9431-6432a88054ea"}
-----
feature: features/RegistrarPedidosOutline.feature
scenarios: 19 | passed: 19 | failed: 0 | time: 2,0241
-----

07:00:45.017 [main] INFO com.intuit.karate.Suite - <<pass>> feature 6 of 6 (0 remaining) features/RegistrarPedidosOutline.feature
07:00:45.022 [main] DEBUG com.intuit.karate.Suite - waiting for 6 features to complete

Karate version: 1.1.0.RC1
=====
elapsed: 17,41 | threads: 1 | thread time: 5,07
features: 6 | skipped: 0 | efficiency: 0,29
scenarios: 46 | passed: 46 | failed: 0
=====

HTML report: (paste into browser to view) | Karate version: 1.1.0.RC1
file:///C:/Users/brunob/Documents/Workshop/Carreira/Workshop/automated-testing-of-bpmn-and-dmn-diagrams/brigadeiros/target/karate-reports/karate-summary.html

PS C:\Users\brunob\Documents\Workshop\Carreira\Workshop\automated-testing-of-bpmn-and-dmn-diagrams\brigadeiros>
```

Suite de testes completa



6

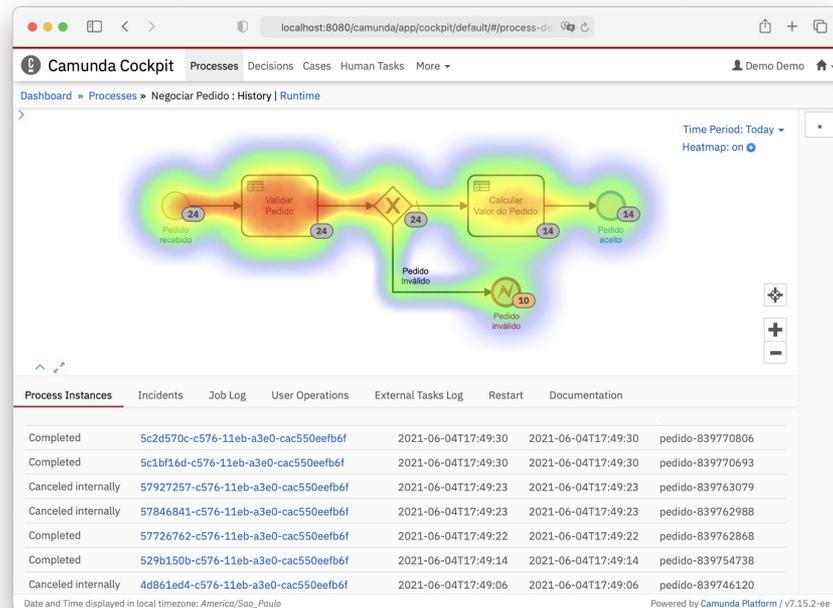
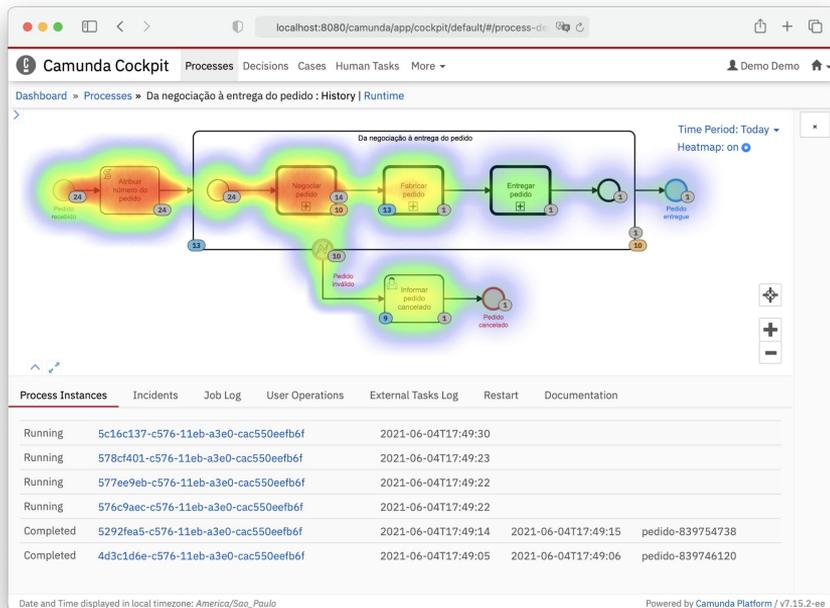
0

Features

Tags

Feature	Title	Passed	Failed	Scenarios	Time (ms)
features/RegistrarPedido.feature	Registrar Pedido	3	0	3	412
features/AvaliarPedidoValido.feature	Avaliar Pedido	19	0	19	1990
features/RegistrarPedidosOutline.feature	Registrar Pedidos	19	0	19	2495
features/E2EPedidoInvalido.feature	Registrar Pedido Inválido E2E	1	0	1	241
features/AvaliarValorPedido.feature	Avaliar Valor do Pedido	2	0	2	75
features/E2EPedidoValido.feature	Registrar Pedido Válido E2E	1	0	1	415

Histórico de Execução dos Processos



Histórico de Execução das Regras de Decisão

The screenshot displays the Camunda Cockpit interface for a Decision Requirement Definition (DRD). The diagram shows a central node 'Tipo de Cliente' (43) which branches into 'Vale a pena' (43) and 'Perfil do Pagador' (43). 'Vale a pena' leads to 'Pedido Válido' (43), and 'Perfil do Pagador' leads to 'Valor do Pedido' (46). Both 'Pedido Válido' and 'Valor do Pedido' lead to a final node (46). Below the diagram is a table of Decision Requirement Definition Instances.

Decision Requirement Definition Instances	Decision Instances			
5c9407c6-c576-11eb-a3e0...	2021-06-04T17:49:31	ValorPedidoDecision	NegociarPedidoProcess	5c8fe8e5-c576-11eb-a3e0...
5c9e4156-c576-11eb-a3e0...	2021-06-04T17:49:31	PedidoValidoDecision	NegociarPedidoProcess	5c9da504-c576-11eb-a3e0...
5ca21205-c576-11eb-a3e0...	2021-06-04T17:49:31	ValorPedidoDecision	NegociarPedidoProcess	5c9da504-c576-11eb-a3e0...
5cab6035-c576-11eb-a3e0...	2021-06-04T17:49:31	PedidoValidoDecision	NegociarPedidoProcess	5caaeaf3-c576-11eb-a3e0...
5cae6d94-c576-11eb-a3e0...	2021-06-04T17:49:31	ValorPedidoDecision	NegociarPedidoProcess	5caaeaf3-c576-11eb-a3e0...

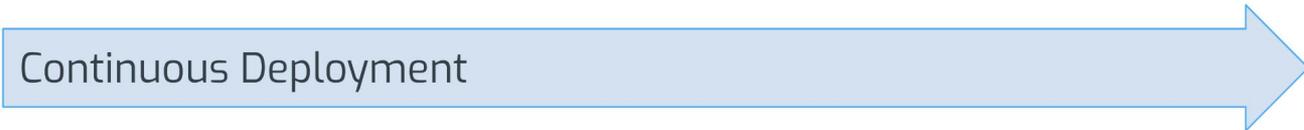
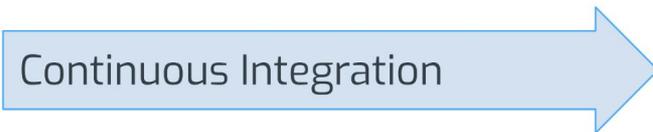
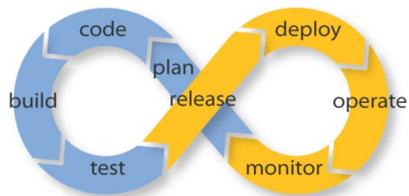
The screenshot displays the execution history for a specific decision instance. The top part shows the DRD diagram for 'Pedido Válido' with a 'Hit Policy: Unique'. Below this is a table showing the execution results for three different input scenarios.

	When = VIP	And = false	Then	And	
	Tipo Cliente	Vale a pena	Pedido Válido	Mensagem	
	string	boolean	boolean	string	
1	"VIP"		true = true	"Pedido aceito" = Pedido aceito	-
2	"Comum"	true	true	"Pedido aceito"	-
3	"Comum"	false	false	"Pedido não vale a pena"	-
	"Ruim"		false	"Pedido de cliente ruim"	-

Below the table is a table showing the inputs and outputs for the decision instance.

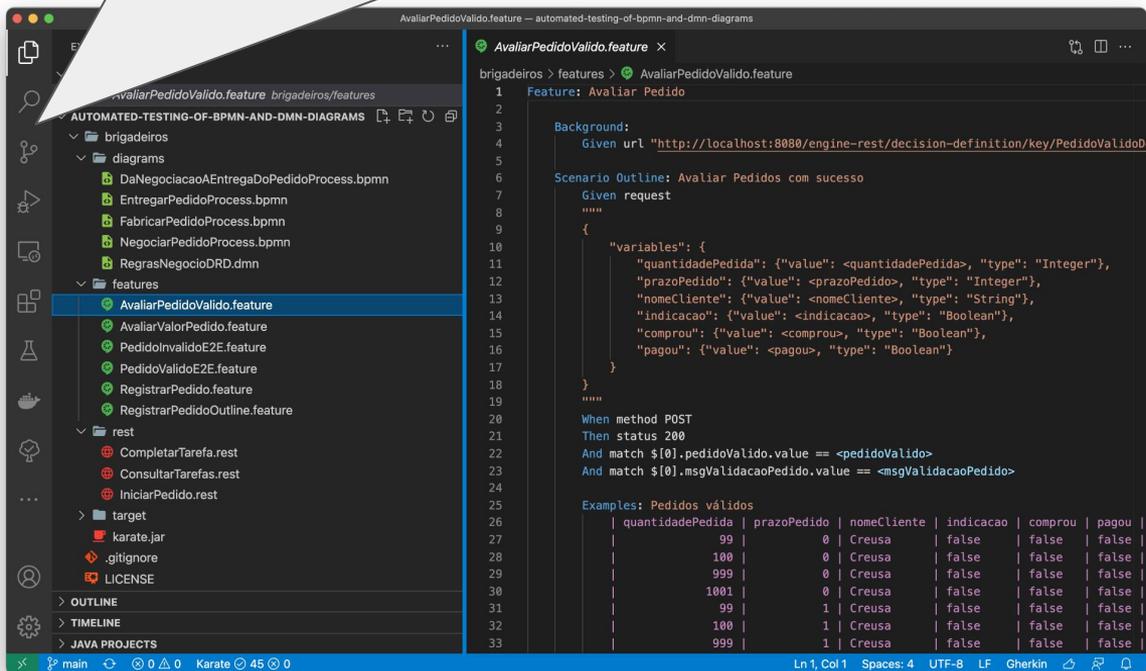
Name	Type	Value
Vale a pena	Boolean	false
Tipo Cliente	String	VIP

Continuous Integration and Delivery (CI / CD)



Compartilhamento com a Comunidade de Usuários

<https://github.com/mbitencourt/automated-testing-of-bpmn-and-dmn-diagrams.git>



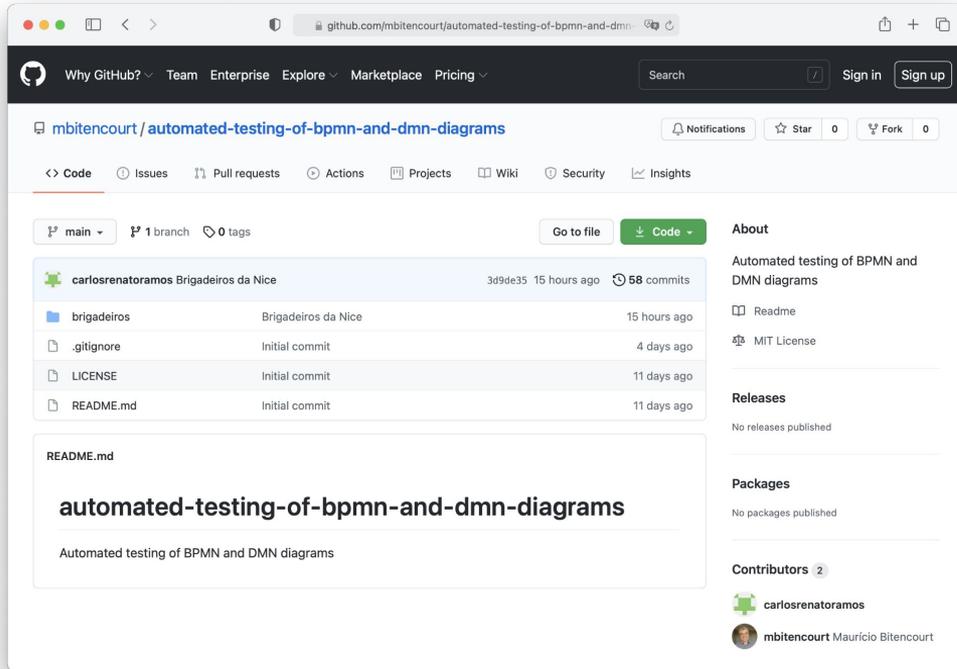
The screenshot shows an IDE with a project structure on the left and a feature file on the right. The project structure includes a 'features' folder with several feature files, including 'AvaliarPedidoValido.feature'. The feature file content is as follows:

```
1 Feature: Avaliar Pedido
2
3 Background:
4   Given url "http://localhost:8080/engine-rest/decision-definition/key/PedidoValidoD
5
6 Scenario Outline: Avaliar Pedidos com sucesso
7   Given request
8     ""
9     {
10      "variables": {
11        "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
12        "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
13        "nomeCliente": {"value": <nomeCliente>, "type": "String"},
14        "indicacao": {"value": <indicacao>, "type": "Boolean"},
15        "comprou": {"value": <comprou>, "type": "Boolean"},
16        "pagou": {"value": <pagou>, "type": "Boolean"}
17      }
18    }
19
20 When method POST
21 Then status 200
22 And match ${0}.pedidoValido.value == <pedidoValido>
23 And match ${0}.msgValidacaoPedido.value == <msgValidacaoPedido>
24
25 Examples: Pedidos válidos
26 | quantidadePedida | prazoPedido | nomeCliente | indicacao | comprou | pagou |
27 | 99 | 0 | Creusa | false | false | false |
28 | 100 | 0 | Creusa | false | false | false |
29 | 999 | 0 | Creusa | false | false | false |
30 | 1001 | 0 | Creusa | false | false | false |
31 | 99 | 1 | Creusa | false | false | false |
32 | 100 | 1 | Creusa | false | false | false |
33 | 999 | 1 | Creusa | false | false | false |
```



<https://github.com/mbitencourt/automated-testing-of-bpmn-and-dmn-diagrams>

Projeto no GitHub



The screenshot shows the GitHub interface for the repository 'mbitencourt/automated-testing-of-bpmn-and-dmn-diagrams'. The repository is owned by 'carlosrenatoramos' and has 58 commits. The file list includes 'brigadeiros', '.gitignore', 'LICENSE', and 'README.md'. The README content is as follows:

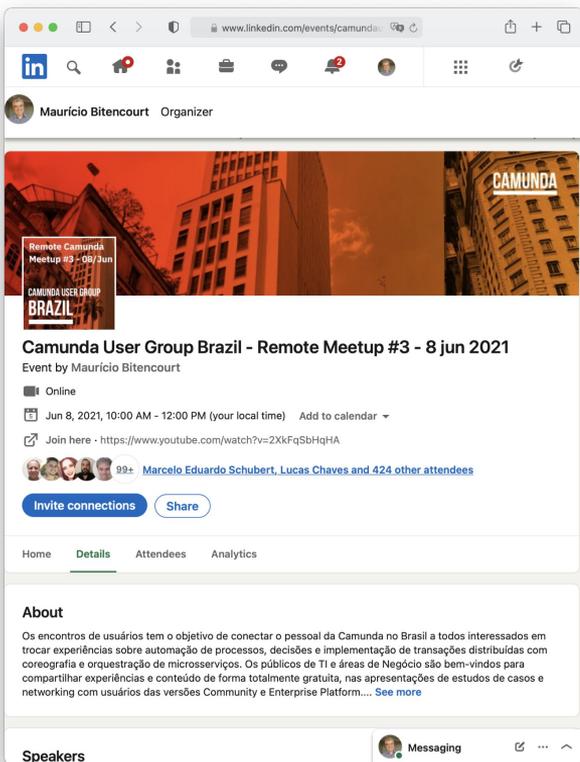
```
automated-testing-of-bpmn-and-dmn-diagrams

Automated testing of BPMN and DMN diagrams
```

The right sidebar contains sections for 'About' (Automated testing of BPMN and DMN diagrams), 'Releases' (No releases published), 'Packages' (No packages published), and 'Contributors' (2 contributors: carlosrenatoramos and mbitencourt).



Gravação desta apresentação



The screenshot shows a LinkedIn event page. At the top, it identifies the organizer as Maurício Bitencourt. The event title is 'Camunda User Group Brazil - Remote Meetup #3 - 8 jun 2021'. It is an online event scheduled for June 8, 2021, from 10:00 AM to 12:00 PM. A link to join the event is provided: <https://www.youtube.com/watch?v=2XkFqSbHqHA>. The page also shows that 99+ people are attending, including Marcelo Eduardo Schubert, Lucas Chaves, and 424 other attendees. There are buttons for 'Invite connections' and 'Share'. Below the event details, there is an 'About' section and a 'Speakers' section.

Event by Maurício Bitencourt

Online

Jun 8, 2021, 10:00 AM - 12:00 PM (your local time) Add to calendar

Join here - <https://www.youtube.com/watch?v=2XkFqSbHqHA>

99+ Marcelo Eduardo Schubert, Lucas Chaves and 424 other attendees

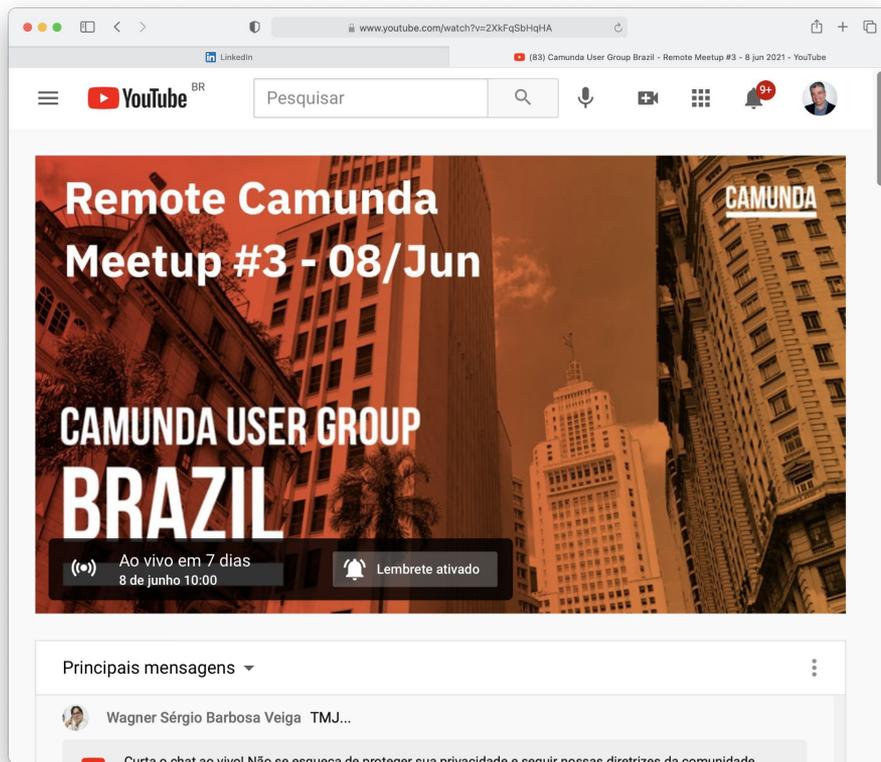
Invite connections Share

Home Details Attendees Analytics

About

Os encontros de usuários tem o objetivo de conectar o pessoal da Camunda no Brasil a todos interessados em trocar experiências sobre automação de processos, decisões e implementação de transações distribuídas com coreografia e orquestração de microsserviços. Os públicos de TI e áreas de Negócio são bem-vindos para compartilhar experiências e conteúdo de forma totalmente gratuita, nas apresentações de estudos de casos e networking com usuários das versões Community e Enterprise Platform.... [See more](#)

Speakers Messaging



The screenshot shows a YouTube video player. The video title is 'Remote Camunda Meetup #3 - 08/Jun'. The video is from the 'Camunda User Group Brazil' channel. The video player shows a thumbnail with the text 'Remote Camunda Meetup #3 - 08/Jun' and 'CAMUNDA USER GROUP BRAZIL'. The video is currently live, with a notification that says 'Ao vivo em 7 dias 8 de junho 10:00'. There is also a notification that says 'Lembrete ativado'. Below the video player, there is a section for 'Principais mensagens' (Main messages) with a dropdown arrow. The first message is from Wagner Sérgio Barbosa Veiga (TMJ...).

YouTube BR

Pesquisar

Camunda User Group Brazil - Remote Meetup #3 - 8 jun 2021 - YouTube

Remote Camunda Meetup #3 - 08/Jun

CAMUNDA USER GROUP BRAZIL

Ao vivo em 7 dias 8 de junho 10:00

Lembrete ativado

Principais mensagens

Wagner Sérgio Barbosa Veiga TMJ...

Curta o chat ao vivo! Não se esqueça de proteger sua privacidade e seguir nossas diretrizes da comunidade

<https://www.linkedin.com/events/camundausergroupbrazil-remoteme6801205810325151744>

<https://www.youtube.com/watch?v=2XkFqSbHqHA>