

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

The screenshot shows a LinkedIn profile for Carlos Renato dos Santos Ramos. At the top, there's a large profile picture of a man with dark hair and a beard. Below it, his name is listed as "Carlos Renato dos Santos Ramos 1st". Underneath his name, it says "Analista de Informática na Câmara dos Deputados" and "Brasília, Federal District, Brazil - Contact info". It also indicates he has "2 connections". There are two buttons: "Message" and "More".

In the "Highlights" section, there's a post from the "Camunda User Group Brazil - Remote Meetup #3 - 8 jun 2021" event, where Carlos Renato is speaking. There's a "Message" button next to it.

The "Activity" section shows three items:

- A post from "Camunda User Group Brazil - Remote Meetup #3 - 8 jun 2021" where Carlos Renato shared a link. A comment from Maurício Bitencourt is shown: "Parabéns Maurício, super merecido !!!" with a thumbs-up icon. Carlos Renato commented: "Carlos Renato commented".
- A post from "Camunda User Group Brazil - Remote Meetup #3 - 8 jun 2021" where Carlos Renato shared a link. A comment from Maurício Bitencourt is shown: "Muito bom, Maurício Bitencourt" with a thumbs-up icon. Carlos Renato commented: "Carlos Renato commented".

At the bottom right of the LinkedIn interface, there are icons for messaging, notifications, and more.

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/>

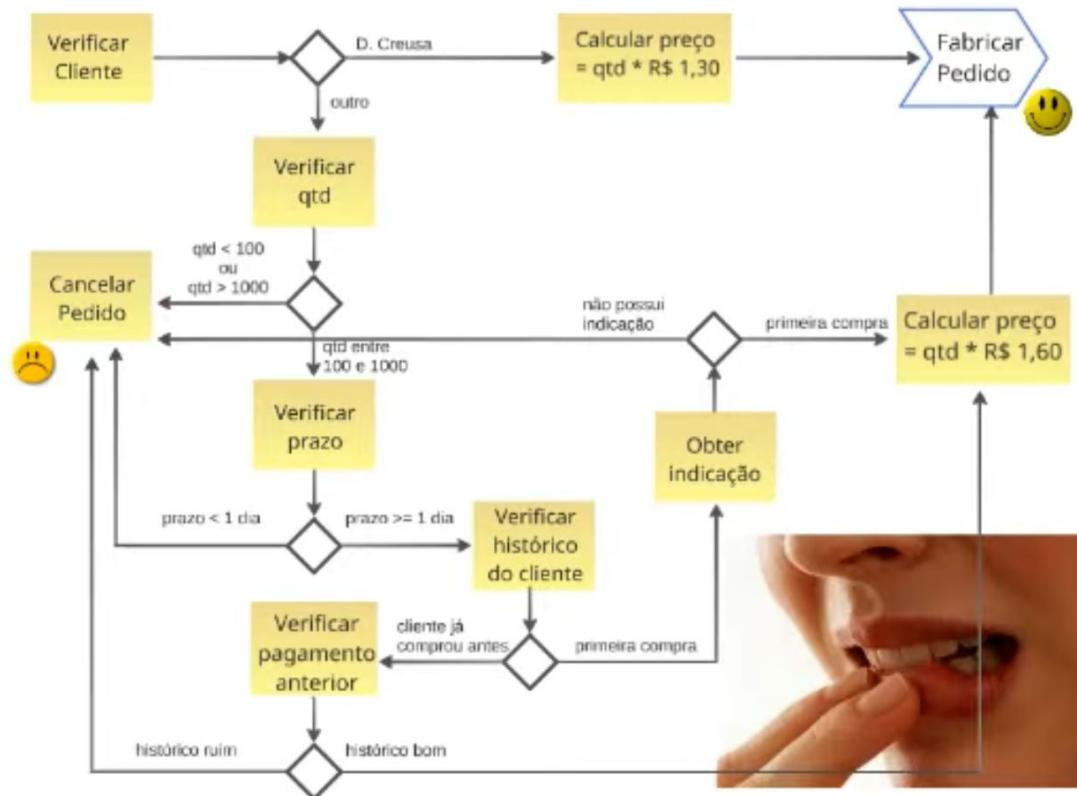
<https://www.youtube.com/watch?v=Y6sb1Eos1sl>

Processo ponta a ponta:
Do pedido à entrega de brigadeiros

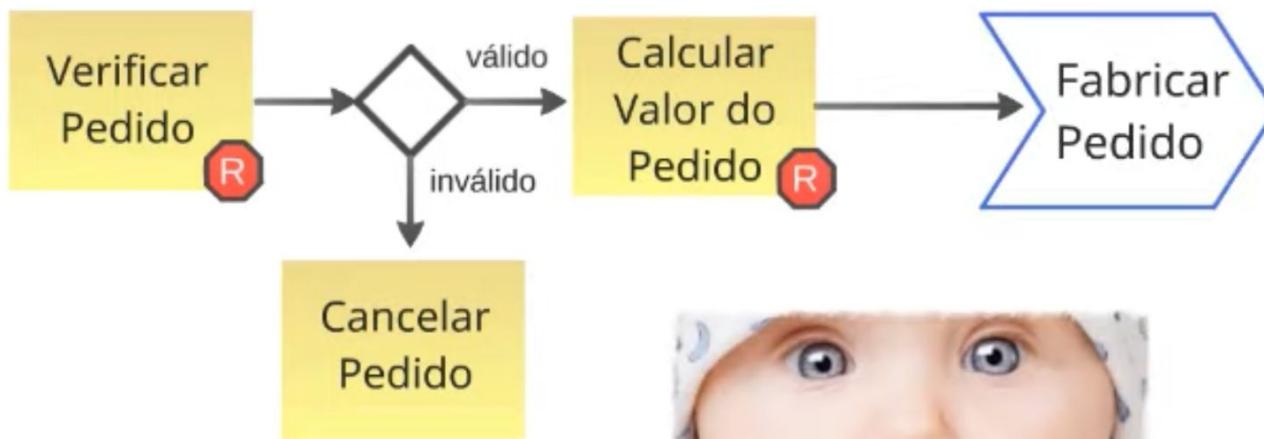
Processo de Negócio



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		

Tabela de Cliente		
Nome	100	Novo
CPF	100	
Renda	100	
Salário	100	

Vale a pena

Quantidade pedida
Prazo pedido

Tabela de Pedido		
Nome	100	Novo
CPF	100	
Valor	100	
Data	100	

Qual o valor
do pedido?



Valor do Pedido		
Tipo de Cliente		
Quantidade solicitada		

Tabela de Pedido		
Nome	100	Novo
CPF	100	
Valor	100	
Data	100	

Tipo de Cliente

Perfil do pagador
Tem indicação
Nome

Tabela de Pagador		
Nome	100	Novo
CPF	100	
Salário	100	
Endereço	100	

Perfil do pagador
Já comprou antes
Pagou no prazo

Tabela de Pagador		
Nome	100	Novo
CPF	100	
Salário	100	
Endereço	100	

Tabelas de Decisão

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

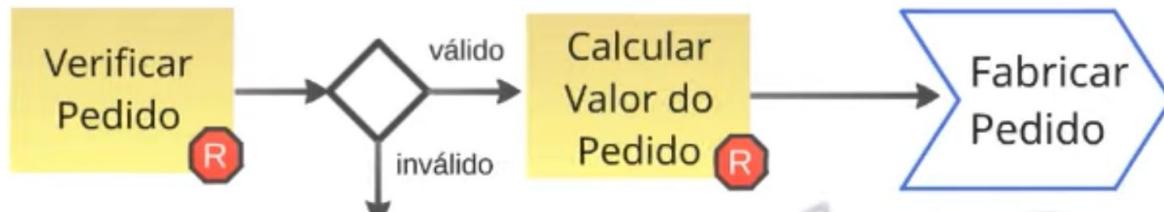
Valor do Pedido	
Tipo de Cliente	Resultado
VIP	qtd * R\$ 1,30
qualquer outro	qtd * R\$ 1,60

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

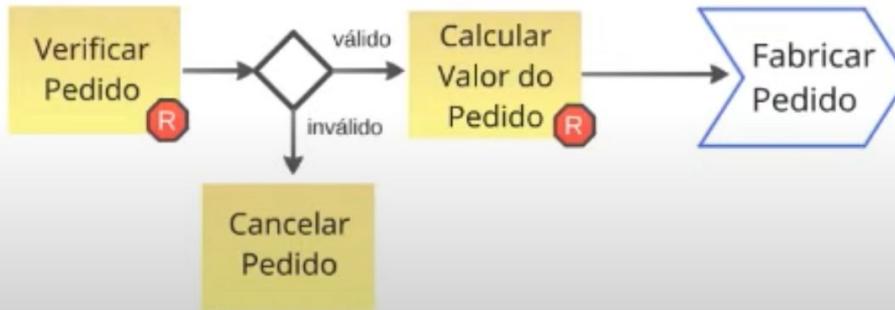
Tipo de Cliente			
Nome	Perfil pagador	Indicação	Resultado
D. Creusa	---	---	VIP
	bom pagador	---	Comum
qualquer outro	desconhecido	sim	Comum
	desconhecido	não	Ruim
	caloteiro	---	Ruim

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

Fluxo do Processo: Negociar Pedido

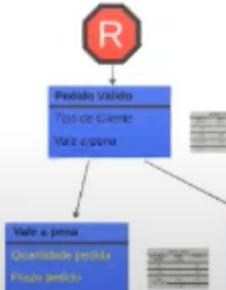


Processos



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

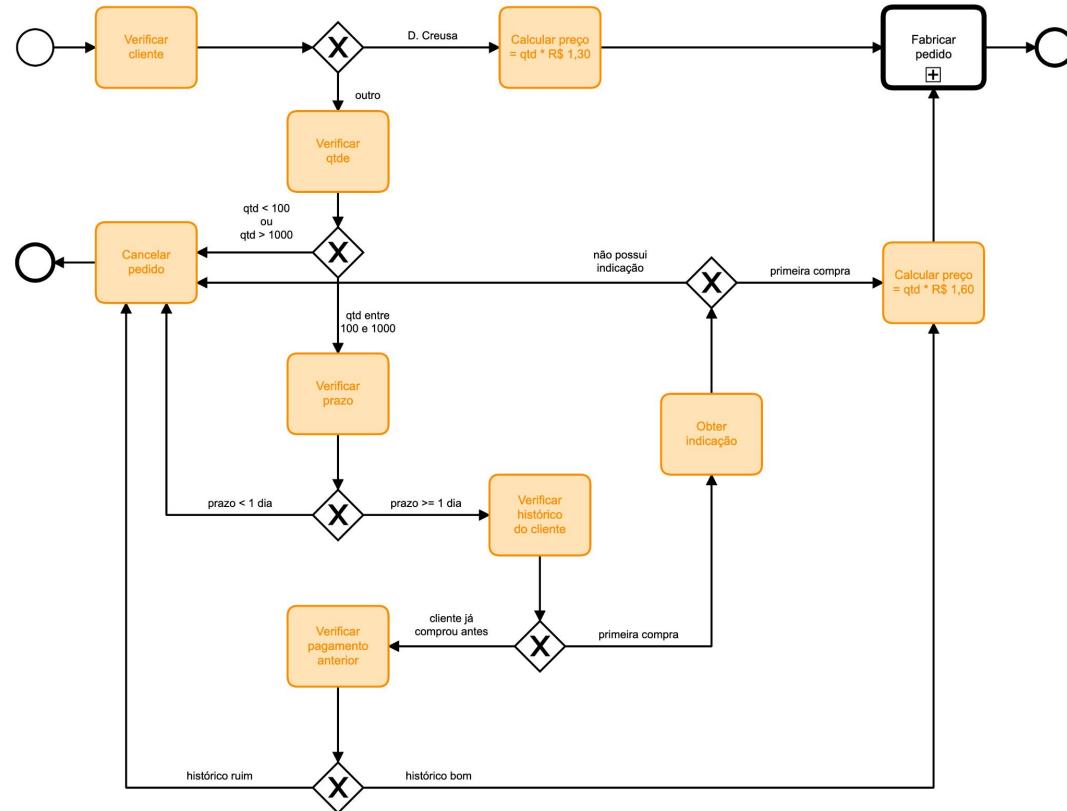
Devem ser considerados clientes vip aqueles que constam de lista específica

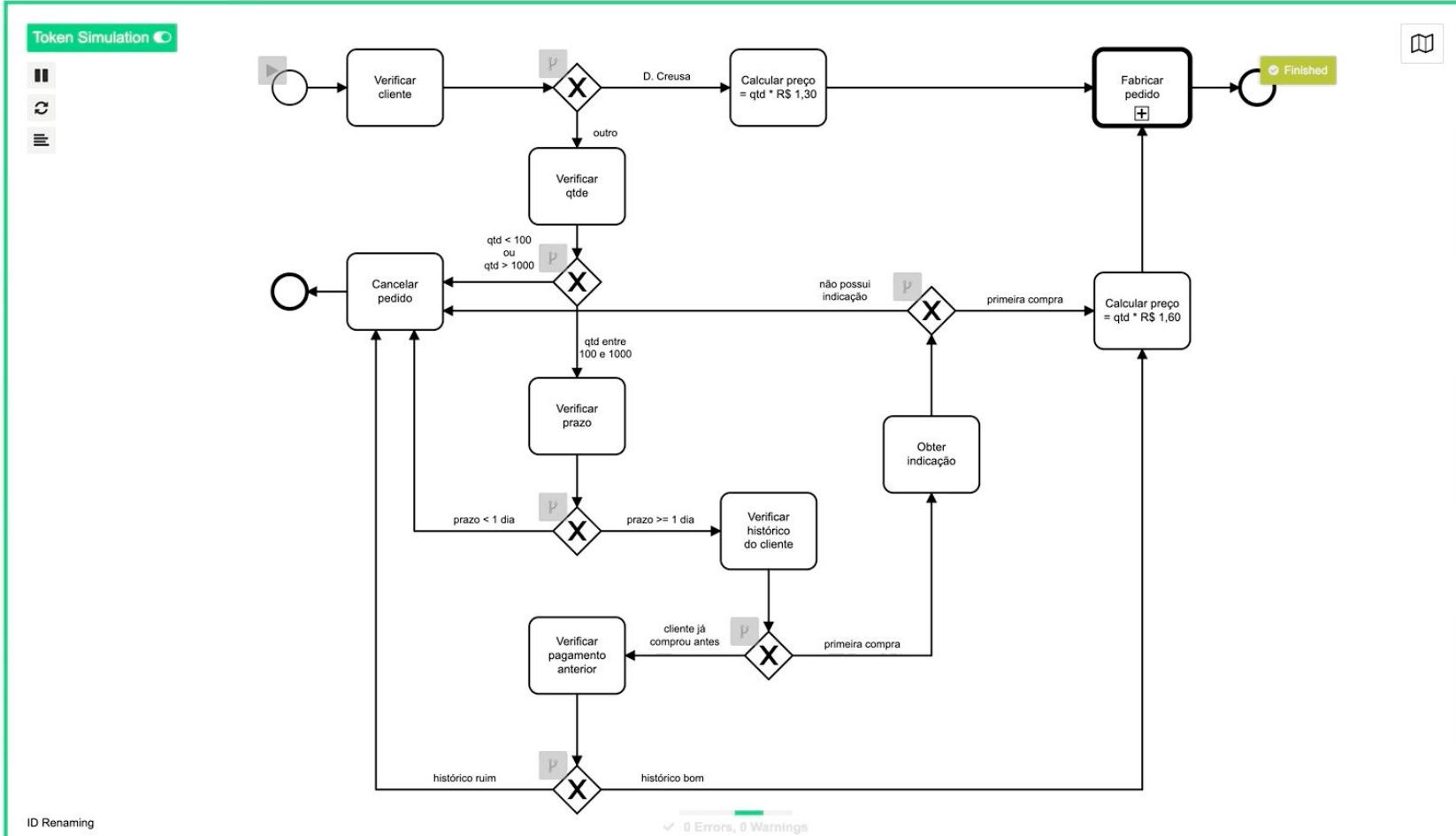
Deve ser cobrado pelo pedido o produto da quantidade pedida pelo preço unitário

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

Fluxo do Processo: Negociar Pedido

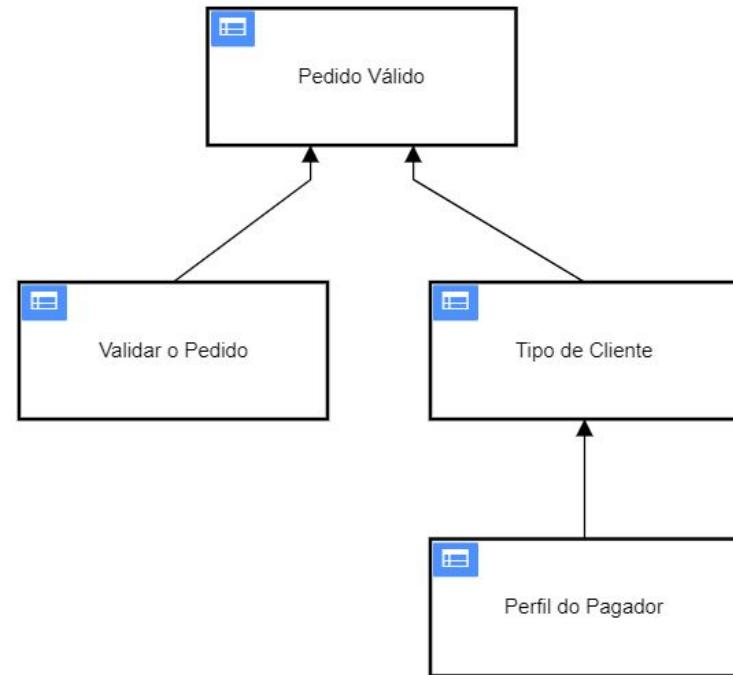
**Antes da
separação de
processos e
decisões**





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
	Tipo Cliente	Vale a pena	Resultado
1	"VIP"	-	"Fabricar pedido"
2	"Comum"	"Sim"	"Fabricar pedido"
3	"Comum"	"Não"	"Cancelar Pedido"
4	"Ruin"	-	"Cancelar pedido"
+	-	-	-



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

Tipo de Cliente		Hit Policy: First	
	When	And	Then
	Nome	Indicação	Perfil Pagador
1	"Creusa"	-	"vip"
2	-	-	"Comum"
3	-	"Sim"	"Desconhecido"
4	-	"Não"	"Desconhecido"
5	-	-	"Caloteiro"
+	-	-	-

Perfil do Pagador		Hit Policy: Unique	
	When	And	Then
	Já comprou	Pagou	Perfil do Pagador
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

DMN Simulator

Inputs:

- Decision table: All tables
- Beverages**
Guests with children: True boolean
- Dish**
Season: Enter String string
How many guests: 0 integer

Outputs:

- Beverages: []
- Dish: []

Simulate now

Dinner Decisions
dinnerDecisions

```
graph TD; subgraph dinnerDecisions [Dinner Decisions]; direction TB; DT[Dinner Decisions]; end; DT --- Dish[Dish]; Dish --> Beverages[Beverages]; Dish --> Guests[Guests with children?];
```

RegrasNegocio.dmn

Simulação do Pedido Inválido

Inputs:

Decision table: All tables

Beverages
Guests with children: True boolean

Dish
Season: Enter String string
How many guests: 0 integer

Outputs:

Beverages: []
Dish: []

Simulate now

Dinner Decisions
dinnerDecisions

```
graph TD; subgraph dinnerDecisions [Dinner Decisions]; direction TB; d1[dinnerDecisions]; end; d1 --> B[Beverages]; d1 --> D[Dish]; B --> G1{Guests with children?}; D --> S1{Season}; D --> NG1[Number of Guests]
```

Upload DMN Table | Download DMN Table

BPMN.io

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:

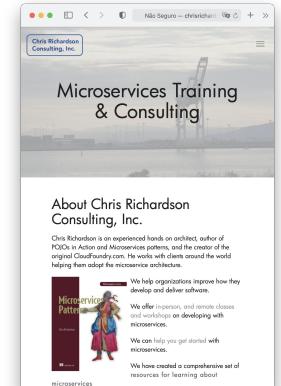
*An application has two categories of requirements. The first category includes the **functional** requirements, which define what the application must do. They're usually in the form of use cases or user stories.*

*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.*

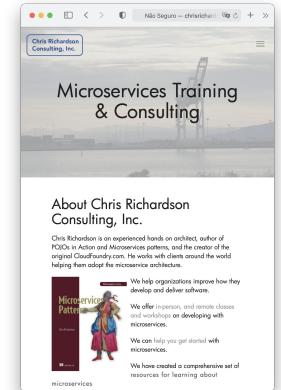
The architecture you choose for your application determines how well it meets these quality requirements.

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 quanto bem ele atende a esses requisitos de qualidade.



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 Platform 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

```
[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/
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] o.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

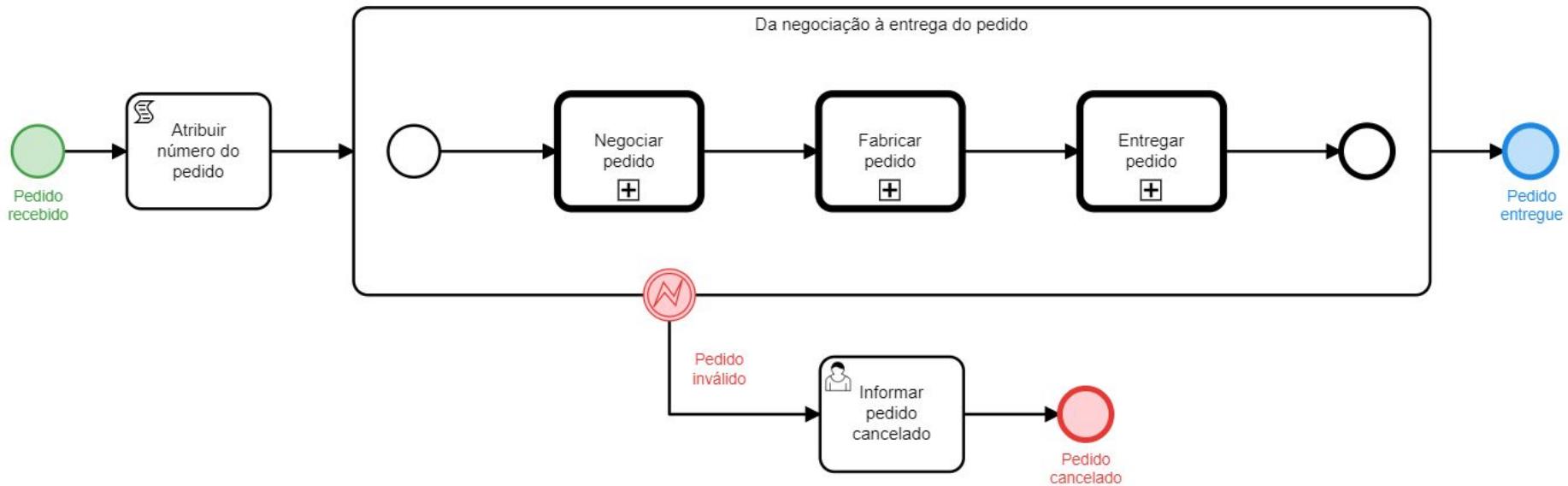


Diagrama em BPMN 2.0

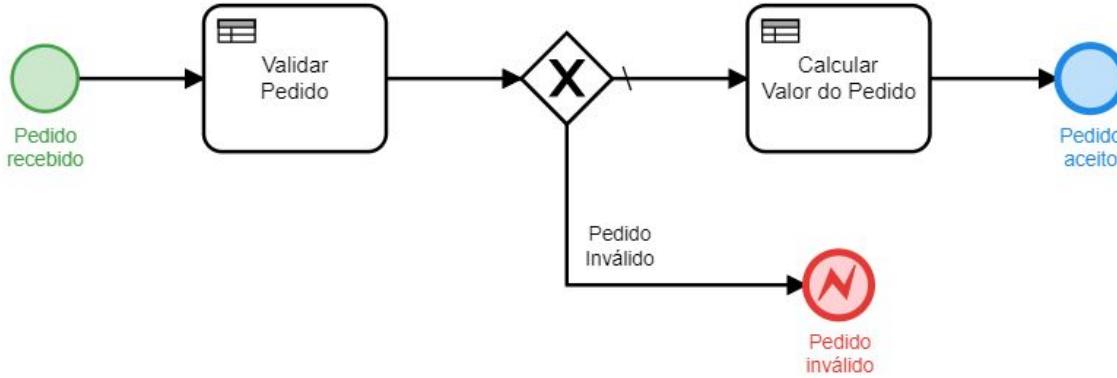


Diagrama em BPMN 2.0

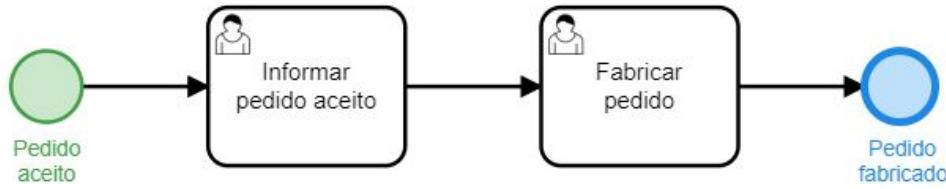
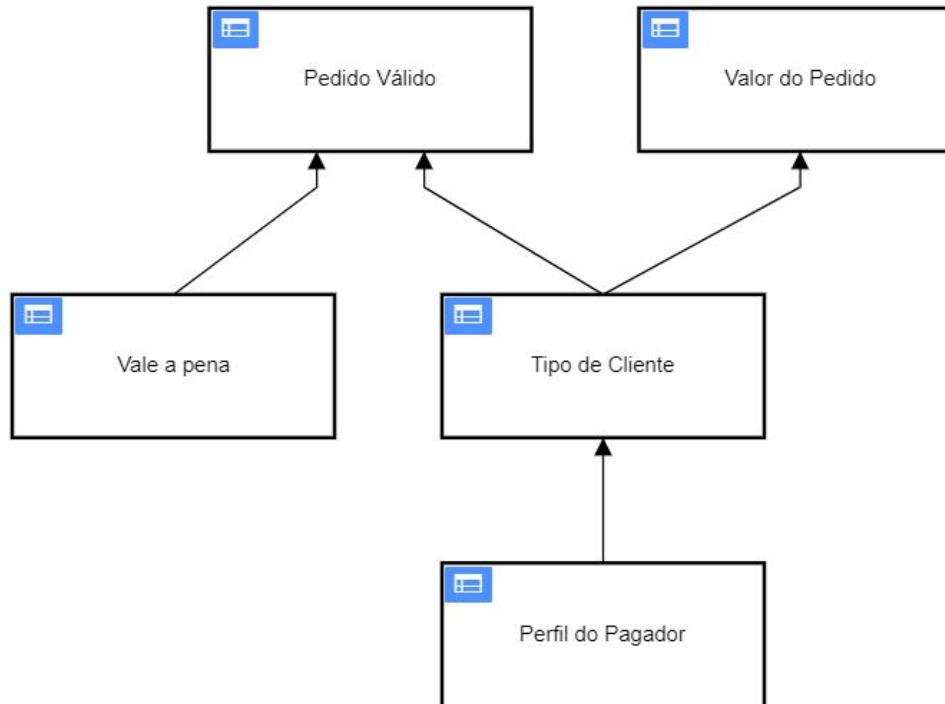


Diagrama em BPMN 2.0



Diagrama em DMN 1.3



Tabelas de Decisão em DMN

Pedido Válido		Hit Policy: Unique			
	When	And	Then	And	Annotations
	Tipo Cliente	Vale a pena	Pedido Válido	Mensagem	
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	"Ruin"	-	false	"Pedido de cliente ruim"	
+	-	-	-		

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

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

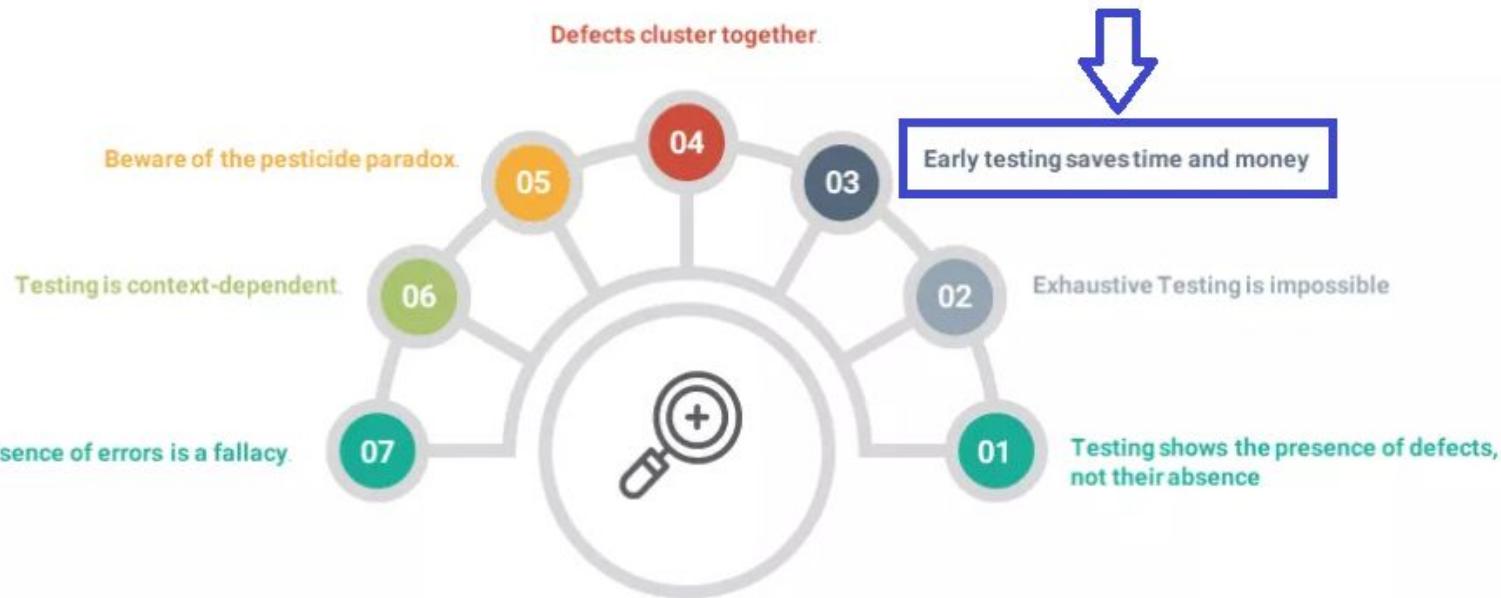
Tipo de Cliente		Hit Policy: First			
	When	And	Then		Annotations
	Nome Cliente	Perfil Pagador	Indicação	Tipo do Cliente	
1	"Creusa"	-	-	"VIP"	
2	-	"Bom pagador"	-	"Comum"	
3	-	"Desconhecido"	true	"Comum"	
4	-	"Desconhecido"	false	"Ruin"	
5	-	"Caloteiro"	-	"Ruin"	
+	-	-	-	-	

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

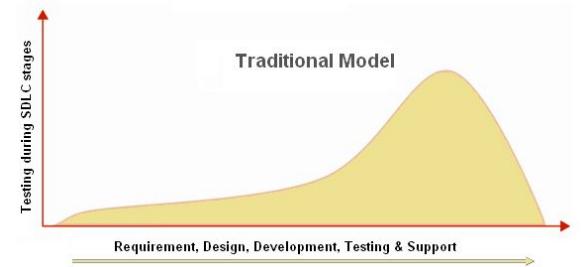
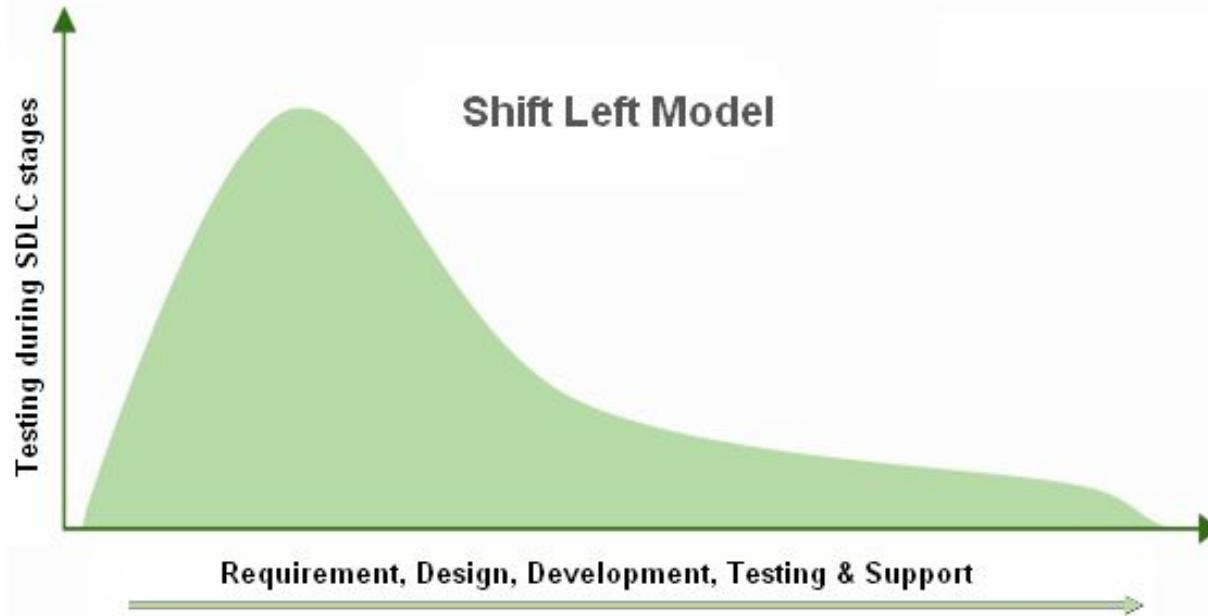
Testes no processo Negociar Pedido

- 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



David Ries
Shift Left Testing: What, Why &
How To Shift Left

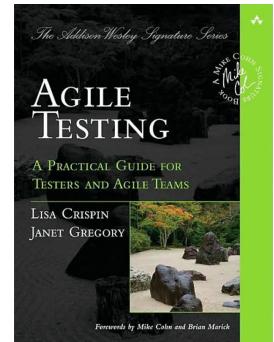
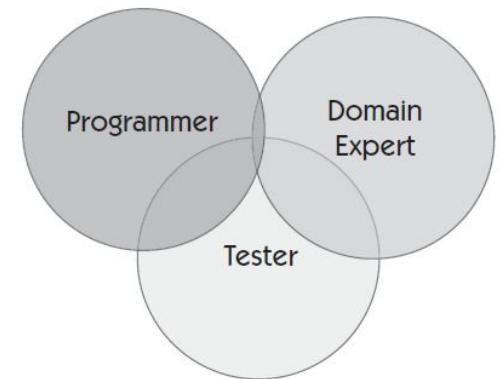


November 30, 2020
5 minute read
Kindle Magowan

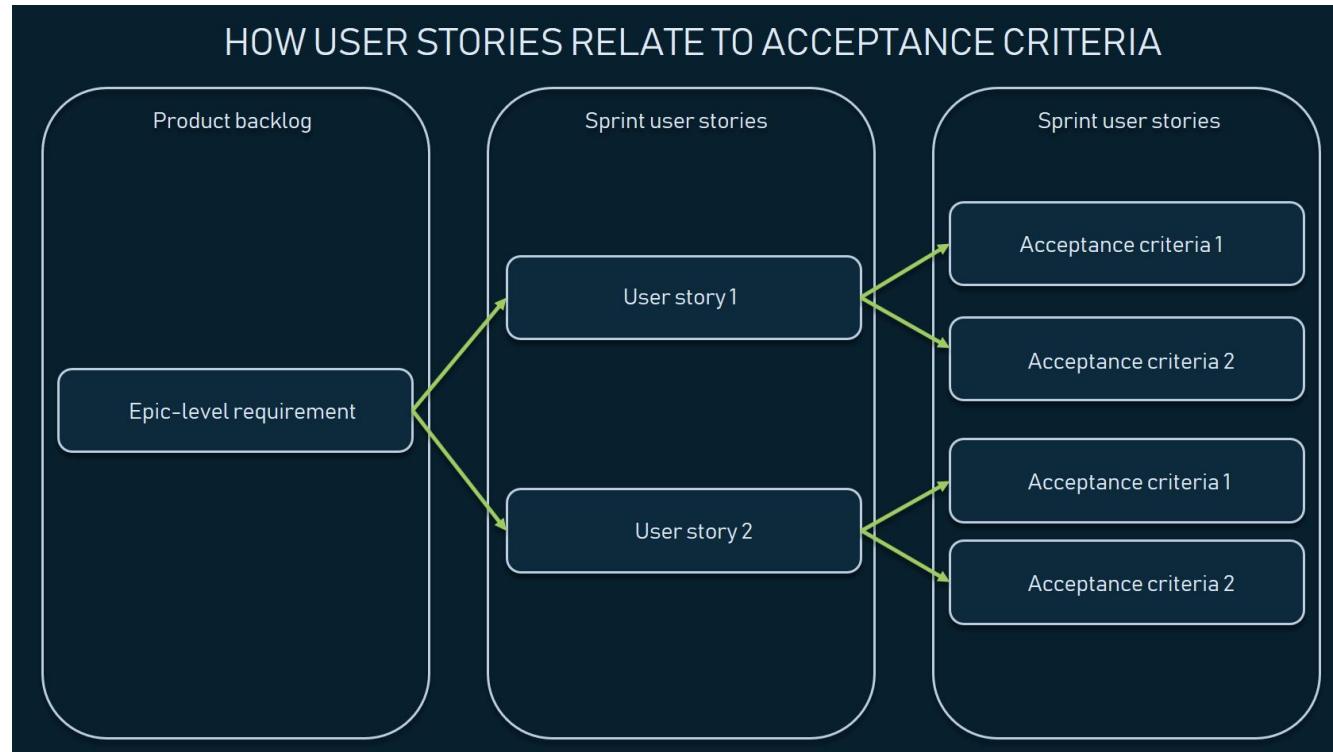
Os “três amigos”



Interaction of Roles

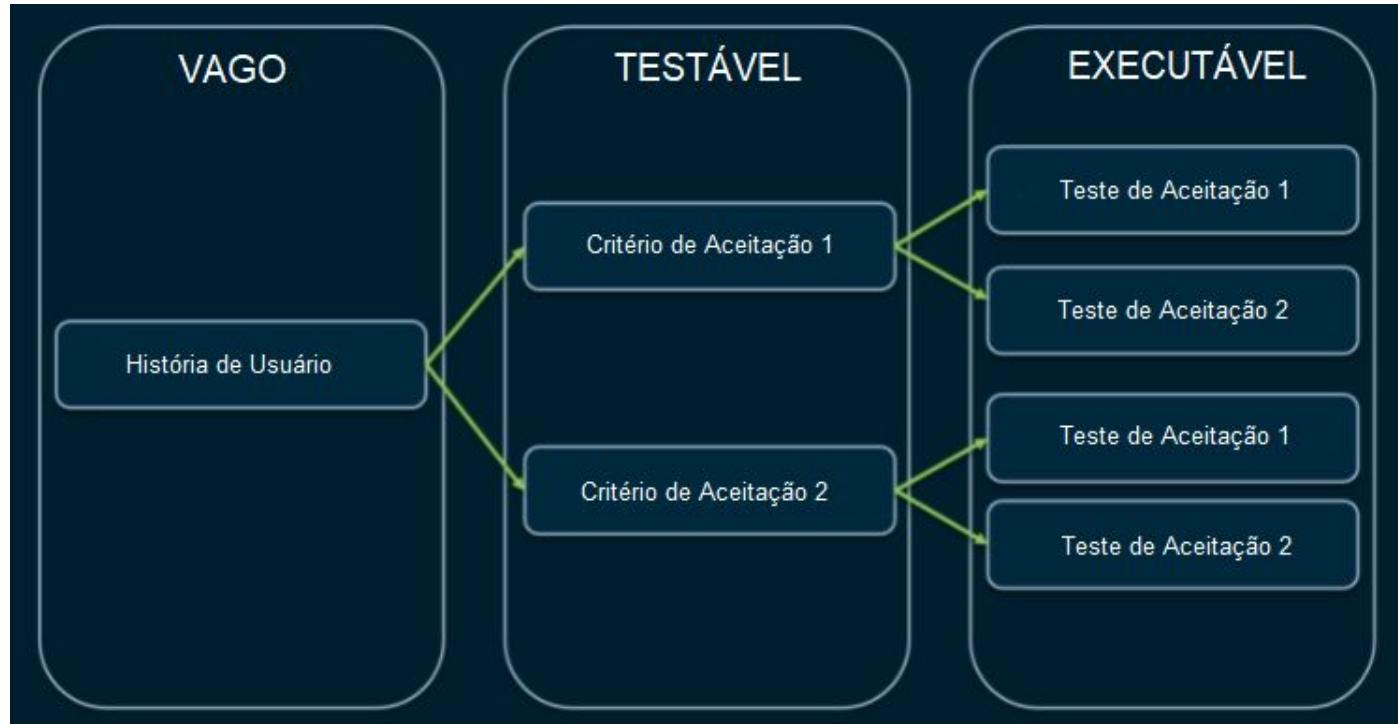


Critérios de aceitação



The screenshot shows a blog post titled "Acceptance Criteria for User Stories: Purposes, Formats, Examples, and Best Practices". The post was last updated on 18 May, 2021. It includes a summary, social sharing options, and a comment section. The content discusses the purpose of acceptance criteria, their formats, examples, and best practices, emphasizing their role in preventing scope creep and ensuring product quality.

Testes de aceitação



Especificações executáveis

Texto citado:

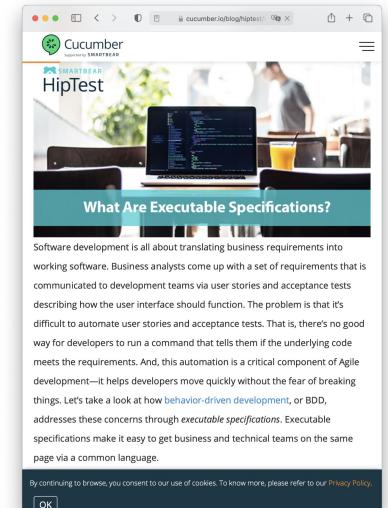
Executable specifications make it easy to get business and technical teams on the same page via a common language.

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*.



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

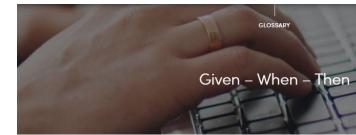
Tradução livre:

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



Agile Essentials Overview Agile 101 Agile Manifesto 12 Principles Agile Subw

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 withdrawls 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 Behave, 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:

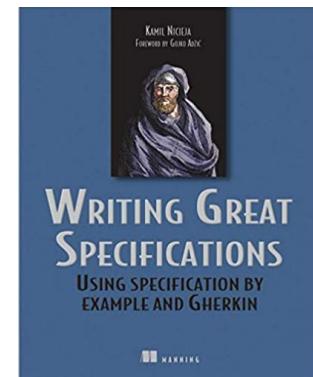
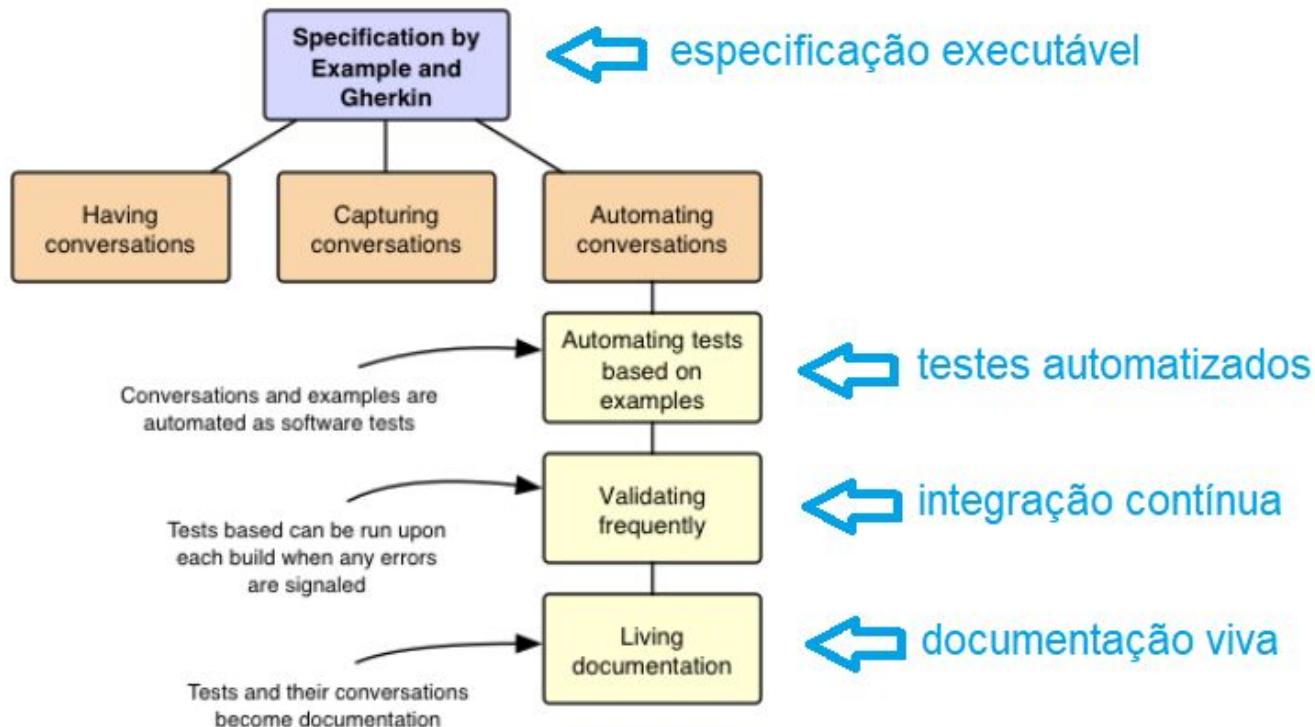
- Feature
- Rule (as of Gherkin 6)
- Example (or Scenario)
- Given, When, Then, And, But for steps (or *)
- Background
- Scenario Outline (or Scenario Template)
- Examples (or Scenarios)

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





www.growingagile.co.za

@growingAgile

A screenshot of a web browser showing the "Meet the Growing Agile Founders" page. It features two profiles: Karen Greaves (Agile Coach) and Sam Laing (Agile Coach). Each profile includes a photo, a title, a brief bio, and social media links.

Karen Greaves
Agile Coach
Karen is a recovered project manager who loves the simplicity and fun that agile brings to work. She's usually the loudest person in the room, and loves anything to do with numbers or email: karen@growingagile.co.za

Sam Laing
Agile Coach

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 sections for 'Vale a pena' (with fields for Quantidade pedida: 500 and Prazo Pedido: 2), 'Tipo de Cliente' (with Nome Cliente: Creusa and Indicação: True), and 'Perfil do Pagador' (with Já comprou: True and Pagou: True). 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. At the bottom, a 'Simulate now' button is visible. Below the main form, a 'View DRD' section shows the internal structure of the decision table 'Pedido Válido'. The table has columns for When, And, Then, And, and Annotations. It contains two rows: one for 'VIP' clients with annotations 'Pedido aceito' and another for 'Comum' clients with the same annotation.

	When	And	Then	And	Annotations
1	Tipo Cliente string	Vale a pena boolean	Pedido Válido boolean	Mensagem string	"Pedido aceito"
2	"Comum" true	- true	true true		"Pedido aceito"

← conferência visual

Cliente REST

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** IniciarPedido.rest - brigadeiros - Visual Studio Code.
- Explorer:** Shows a tree view of files under the BRIGADEIROS folder, including diagrams, features, rest (with sub-items like CompletarTarefa.rest, ConsultarTarefas.rest, and IniciarPedido.rest), target, and karate.jar.
- Editor:** The main editor area displays a JSON configuration for a REST request to start a process definition. The URL is POST http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start. The JSON payload defines variables for the process:

```
POST http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start
Content-Type: application/json

{
  "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"
    }
  },
  "withVariablesInReturn": true
}
```
- Toolbar:** Includes icons for file operations, search, and run.
- Bottom Status Bar:** Shows file paths (main, Karate), line count (Ln 1, Col 1), tab size (Tab Size: 4), encoding (UTF-8), and environment (No Environment).

Cliente REST

The screenshot shows a Visual Studio Code interface with two main panes. The left pane is the Explorer view, showing a file structure under the 'rest' folder. The right pane displays a REST client request and its corresponding response.

Request (IniciarPedido.rest):

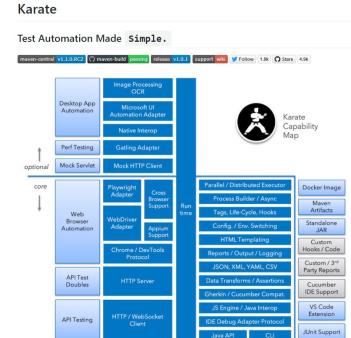
```
POST http://localhost:8080/engine-rest/process-definition/key/...  
Content-Type: application/json  
  
{  
    "variables": {  
        "quantidadePedida": {  
            "value": 100,  
            "type": "Integer"  
        },  
        "prazoPedido": {  
            "value": 1,  
            "type": "Integer"  
        },  
        "nomeCliente": {  
            "value": "creusa",  
            "type": "String"  
        },  
        "indicacao": {  
            "value": true,  
            "type": "Boolean"  
        },  
        "comprou": {  
            "value": true,  
            "type": "Boolean"  
        },  
        "pagou": {  
            "value": true,  
            "type": "Boolean"  
        }  
    },  
    "withVariablesInReturn": true  
}
```

Response (1823ms):

```
HTTP/1.1 200  
Content-Type: application/json  
Transfer-Encoding: chunked  
Date: Sun, 06 Jun 2021 21:31:09 GMT  
Connection: close  
  
{  
    "id": "81dee535-c70e-11eb-9431-6432a88054ea",  
    "definitionId": "90774906-c4c5-11eb-bfa3-6432a88054ea",  
    "businessKey": "pedido-015068948",  
    "caseInstanceId": null,  
    "ended": false,  
    "suspended": false,  
    "tenantId": null,  
    "variables": {  
        "pedidoValido": {  
            "type": "Boolean",  
            "value": true,  
            "valueInfo": {}  
        },  
        "quantidadePedida": { ... },  
        "indicacao": { ... },  
        "msgValidacaoPedido": {  
            "type": "String",  
            "value": "Pedido aceito",  
            "valueInfo": {}  
        },  
        "numeroPedido": {  
            "type": "String",  
            "value": "015068948",  
            "valueInfo": {}  
        }  
    }  
}  
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](#)



Karate Framework

Hello World

For API Testing

Scenario: create and retrieve a cat

Given url '<http://myhost.com/v1/cats>'

```
And request { name: 'Billie' }
```

When method post

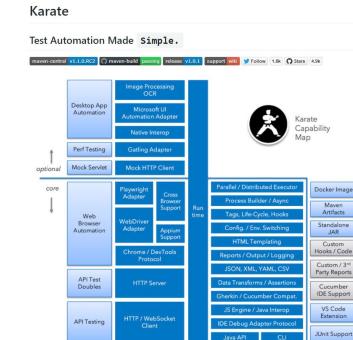
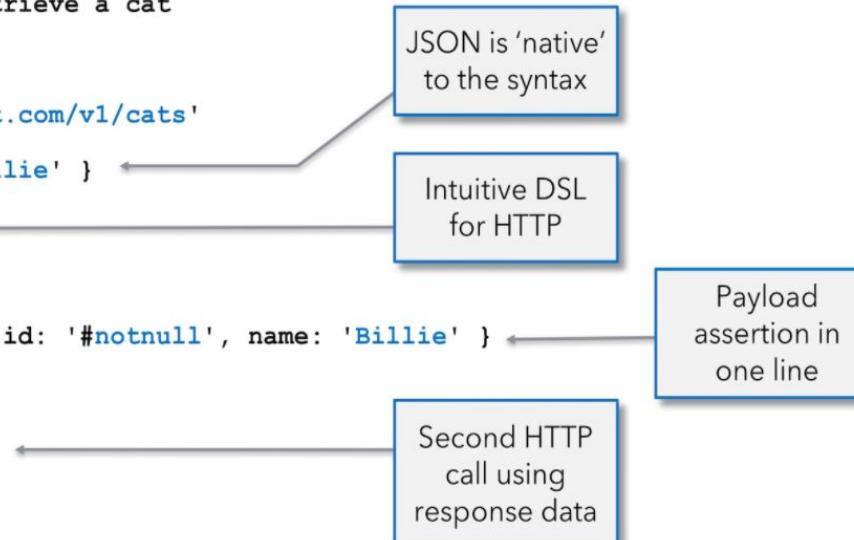
Then status 201

And match response == { id: '#notnull', name: 'Billie' }

Given path response.id

When method get

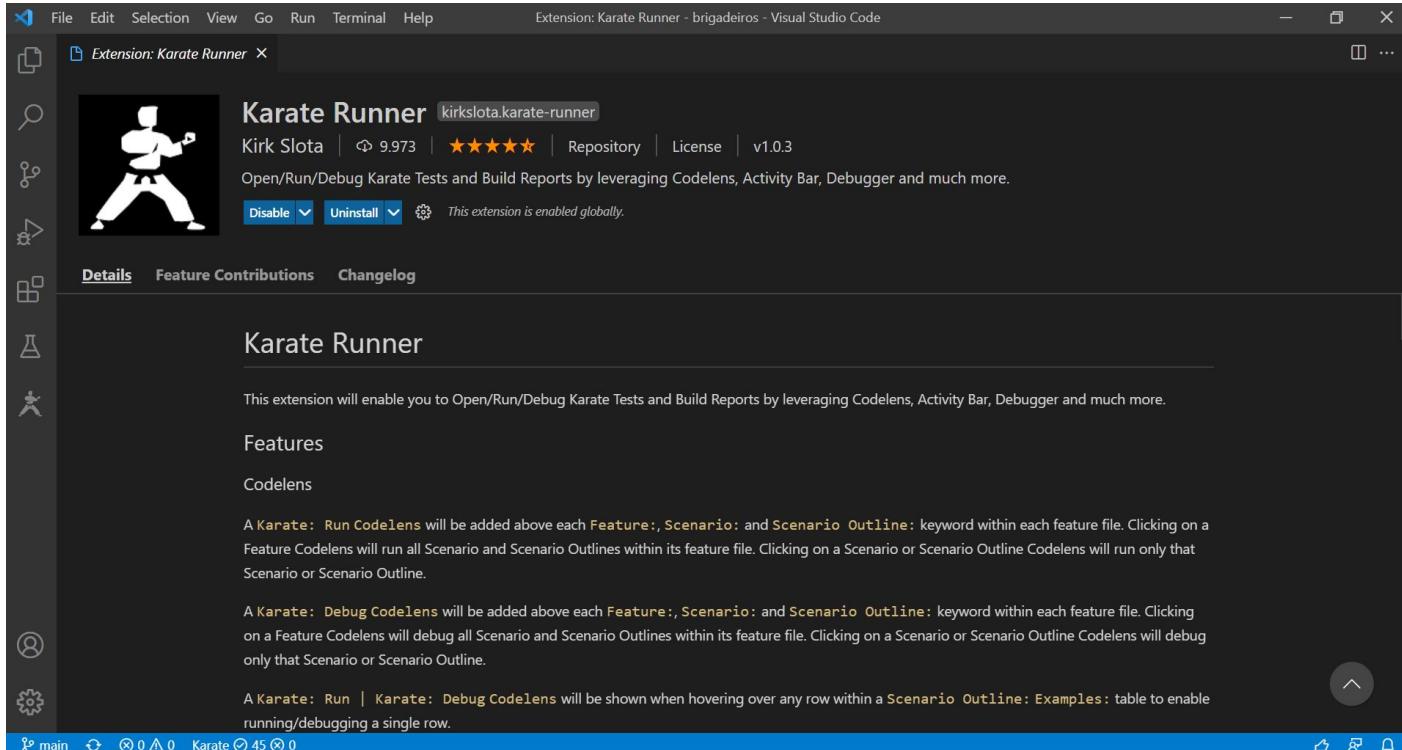
Then status 200



Testes no processo Negociar Pedido

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

Visual Studio Code (extensão Karate Runner)



Registrar Pedido (Scenario)

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** RegistrarPedido.feature - brigadeiros - Visual Studio Code.
- Explorer:** Shows the project structure under "BRIGADEIROS".
- Editor:** Displays the content of "RegistrarPedido.feature".
- Bottom Status Bar:** main, Karate, Ln 72, Col 47, Spaces: 4, UTF-8, CRLF, Karate.

The "RegistrarPedido.feature" file contains the following scenarios:

```
Feature: Registrar Pedido
  Background: ...
  Scenario: Registrar Pedido de cliente vip ...
  Scenario: Registrar Pedido de cliente comum ...
  Scenario: Registrar Pedido que não vale a pena ...
  Scenario: Registrar Pedido de cliente ruim ...
```

A detailed table view is shown for the first scenario:

Pedido Válido	Hit Policy:	Unique		
When	And	Then	And	Mensagem
Tipo Cliente	Vale a pena	Pedido Válido		
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"	
+ -	-	-		

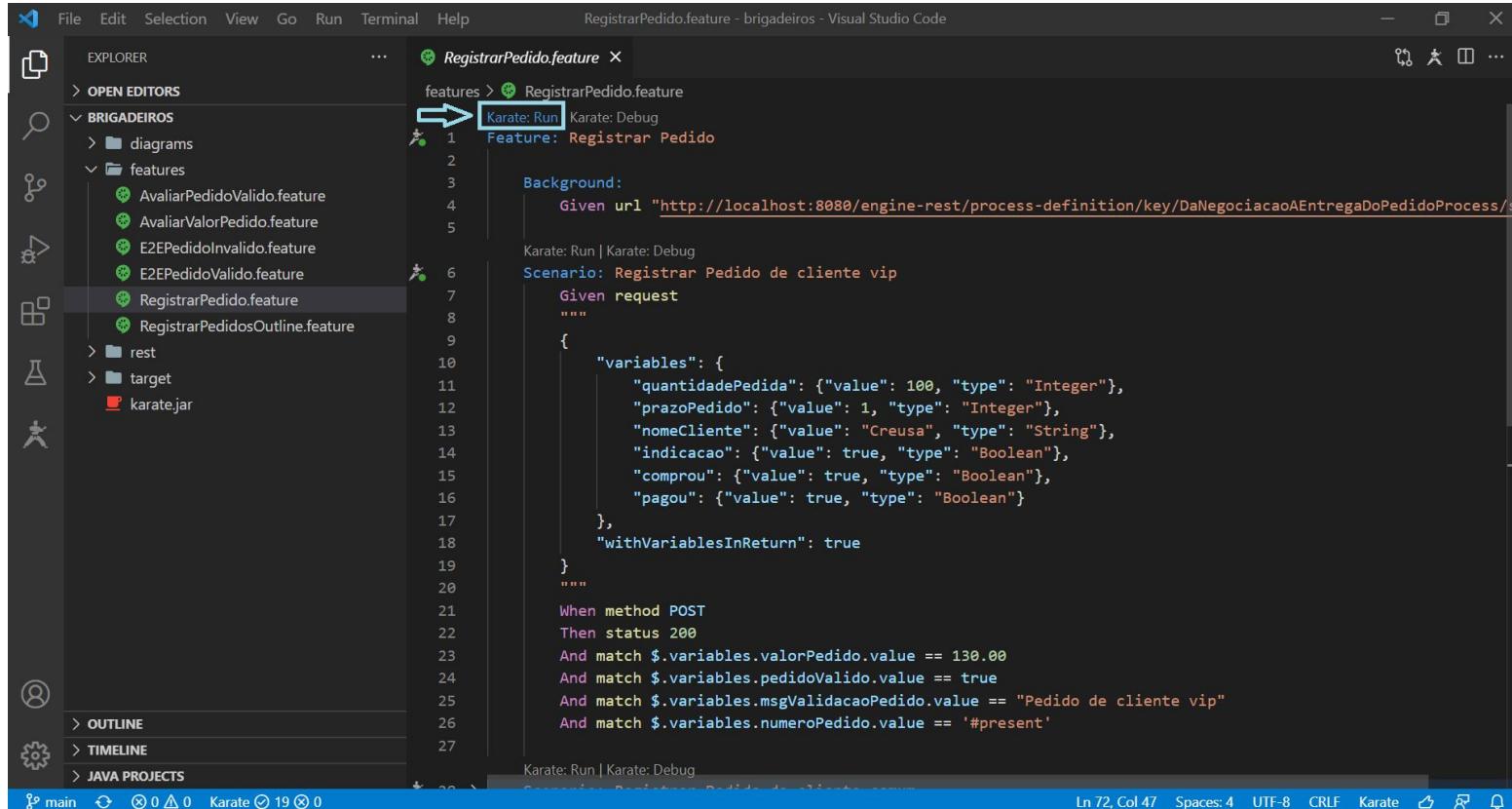
Registrar Pedido (Scenario)

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** RegistrarPedido.feature - brigadeiros - Visual Studio Code.
- Explorer:** Shows a tree view of files under the 'BRIGADEIROS' folder, including 'diagrams', 'features', and several feature files like 'AvaliarPedidoValido.feature', 'AvaliarValorPedido.feature', etc. The file 'RegistrarPedido.feature' is selected.
- Editor:** Displays the content of 'RegistrarPedido.feature'. The code is a Karate feature file with the following structure:

```
Feature: Registrar Pedido
  Background:
    Given url "http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEEntregaDoPedidoProcess"
  Scenario: Registrar Pedido de cliente vip
    Given request
    """
    {
      "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"}
      },
      "withVariablesInReturn": true
    }
    """
    When method POST
    Then status 200
    And match $.variables.valorPedido.value == 130.00
    And match $.variables.pedidoValido.value == true
    And match $.variables.msgValidacaoPedido.value == "Pedido de cliente vip"
    And match $.variables.numeroPedido.value == '#present'
```
- Annotations:** Three arrows point to specific parts of the code:
 - An arrow points to the 'Background' section with the label "situação inicial".
 - An arrow points to the 'When method POST' section with the label "ação a ser testada".
 - An arrow points to the 'Then' section with the label "resultado esperado".
- Bottom Status Bar:** main, Karate, Ln 72, Col 47, Spaces: 4, UTF-8, CRLF, Karate.

Registrar Pedido (Scenario)



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Title Bar:** RegistrarPedido.feature - brigadeiros - Visual Studio Code
- Explorer:** Shows a tree view of files and folders under the 'BRIGADEIROS' project. The 'features' folder contains several feature files: AvaliarPedidoValido.feature, AvaliarValorPedido.feature, E2EPedidoInvalido.feature, E2EPedidoValido.feature, RegistrarPedido.feature (selected), and RegistrarPedidosOutline.feature.
- Editor:** The main editor area displays the content of the selected feature file, 'RegistrarPedido.feature'. A red arrow points to the 'Karate: Run' button in the toolbar above the editor.
- Editor Content:**

```
1 Feature: Registrar Pedido
2
3 Background:
4   Given url "http://localhost:8080/engine-rest/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/stage"
5
6 Scenario: Registrar Pedido de cliente vip
7   Given request
8   """
9   {
10     "variables": {
11       "quantidadePedido": {"value": 100, "type": "Integer"},
12       "prazoPedido": {"value": 1, "type": "Integer"},
13       "nomeCliente": {"value": "Creusa", "type": "String"},
14       "indicacao": {"value": true, "type": "Boolean"},
15       "comprou": {"value": true, "type": "Boolean"},
16       "pagou": {"value": true, "type": "Boolean"}
17     },
18     "withVariablesInReturn": true
19   }
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'
```
- Bottom Status Bar:** main, Karate, Ln 72, Col 47, Spaces: 4, UTF-8, CRLF, Karate

Registrar Pedido (Scenario)

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Terminal:** RegistrarPedido.feature - brigadeiros - Visual Studio Code
- Explorer:** Shows a file tree under the BRIGADEIROS folder:
 - diagrams
 - features
 - AvaliarPedidoValido.feature
 - AvaliarValorPedido.feature
 - E2EPedidoInvalido.feature
 - E2EPedidoValido.feature
 - RegistrarPedido.feature
 - RegistrarPedidosOutline.feature
 - rest
 - target
 - karate.jar
- Terminal Content:**

```
1 < Connection: keep-alive
{
  "variables": {
    "pedidoValido": {
      "valueInfo": {},
      "type": "Boolean",
      "value": false
    },
    "quantidadePedido": {
      "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": "040891879"
    },
    "nomeCliente": {
      "valueInfo": {},
      "type": "String",
      "value": "Frias"
    },
    "pagou": {
      "valueInfo": {},
      "type": "Boolean",
      "value": false
    },
    "prazoPedido": {
      "valueInfo": {},
      "type": "Integer",
      "value": 1
    },
    "comprou": {
      "valueInfo": {},
      "type": "Boolean",
      "value": true
    }
  },
  "caseInstanceId": null,
  "businessKey": "pedido-040891879",
  "ended": false,
  "tenantId": null,
  "links": [
    {
      "method": "GET",
      "rel": "self",
      "href": "http://localhost:8080/engine-rest/process-instance/a1a8494d-c74a-11eb-9431-6432a88054ea"
    }
  ],
  "id": "a1a8494d-c74a-11eb-9431-6432a88054ea",
  "suspended": false,
  "definitionId": "21a967db-c730-11eb-9431-6432a88054ea"
}

feature: features/RegistrarPedido.feature
scenarios: 4 | passed: 4 | failed: 0 | time: 1,7748
```

=====
0.11.35.103 [main] INFO com.intuit.karate.Suite - <><> feature 1 of 1 (0 remaining) features/RegistrarPedido.feature
Karate version: 1.1.0.RC1
=====
elapsed: 3,63 | threads: 1 | thread time: 1,78
features: 1 | skipped: 0 | efficiency: 0,49
scenarios: 4 | passed: 4 | failed: 0
=====

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

Terminal will be reused by tasks, press any key to close it.
- Bottom Status Bar:** Ln 72, Col 47 Spaces: 4 UTF-8 CRLF Karate

Registrar Pedido (Scenario)

Scenarios		4	0
[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			
Scenario: [1:6] Registrar Pedido de cliente vip		ms: 1215	
>> Background:			
7 Given request		31	
{			
"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			
}			
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)

Scenarios	4	0	2
[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			
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 screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** RegistrarPedidosOutline.feature - brigadeiros - Visual Studio Code.
- Explorer:** Shows the project structure under 'BRIGADEIROS':
 - diagrams
 - features
 - AvaliarPedidoValido.feature
 - AvaliarValorPedido.feature
 - E2EPedidoInvalido.feature
 - E2EPedidoValido.feature
 - RegistrarPedido.feature
 - RegistrarPedidosOutline.feature
 - rest
 - target
 - karate.jar
- Editor:** Displays the feature file content:

```
Feature: Registrar Pedidos
Background: ...
  Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"
  Given request
  """
  {
  }
  """

  When method POST
  Then status 200
  And match $.variables.valorPedido.value == <valorPedido>
  And match $.variables.pedidoValido.value == <pedidoValido>
  And match $.variables.msgValidacaoPedido.value == <msgValidacaoPedido>
  And match $.variables.numeroPedido.value == '#present'

Examples: Pedidos de cliente vip...
Examples: Pedidos de cliente comum...
Examples: Pedidos que não valem a pena...
Examples: Pedidos de cliente ruim...
```
- Annotations:**
 - An arrow points from the text "Scenario Outline: Registrar Pedidos com sucesso" to the heading "Scenario Outline".
 - An arrow points from the placeholder text "<valorPedido>" to the label "variáveis".
 - An arrow points from the "Examples" block to the label "Examples".
- Bottom Status Bar:** main, Karate, 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

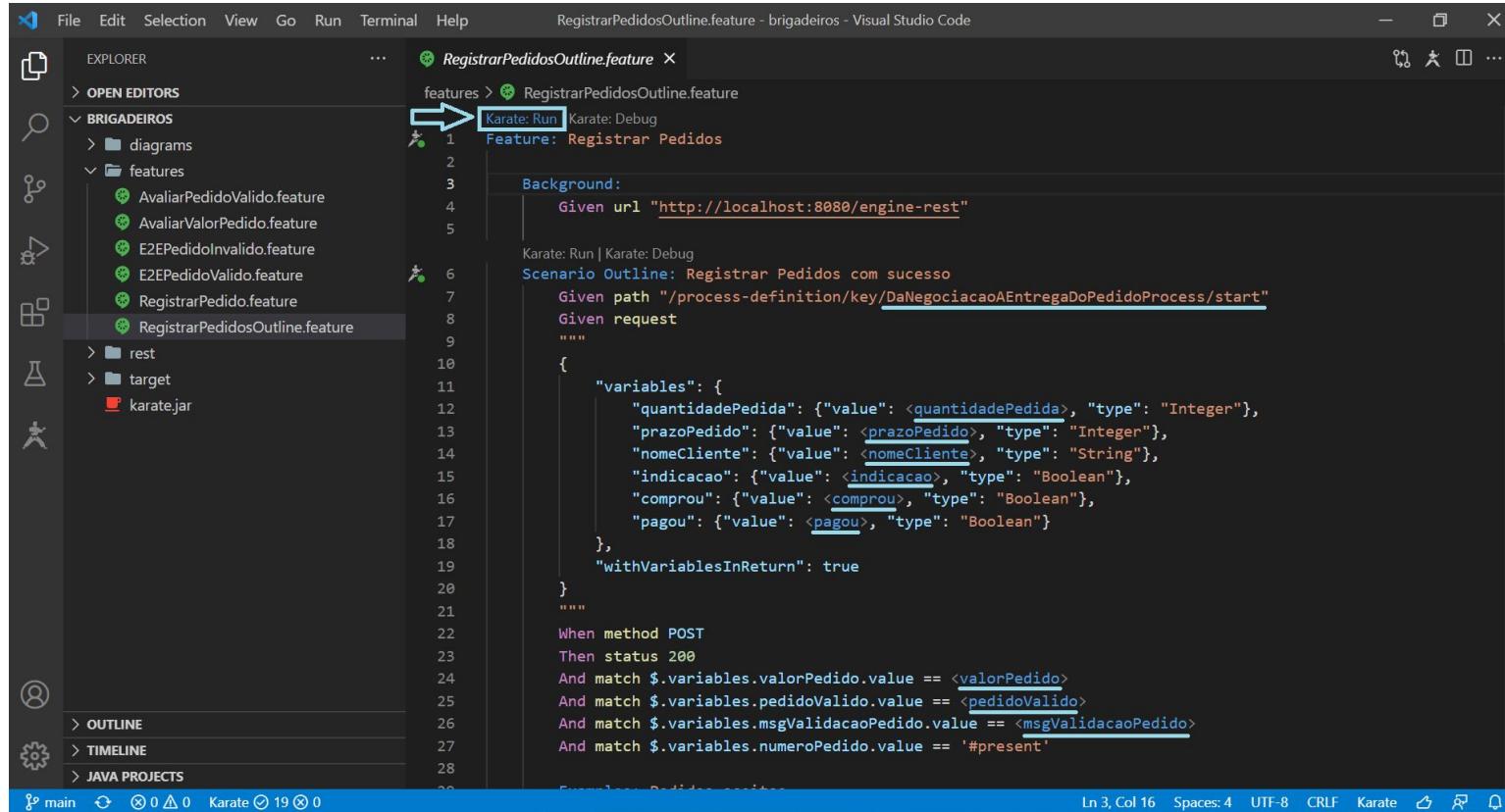
RegistrarPedidosOutline.feature x

features > RegistrarPedidosOutline.feature

Examples: Pedidos de cliente vip

	quantidadePedida	prazoPedido	nomeCliente	indicacao	comprou	pagou	pedidoValido	msgValidacaoPedido	valorPedido
29	99	0	Creusa	false	false	false	true	"Pedido de cliente vip"	128.70
30	100	0	Creusa	false	false	false	true	"Pedido de cliente vip"	130.00
31	999	0	Creusa	false	false	false	true	"Pedido de cliente vip"	1298.70
32	1001	0	Creusa	false	false	false	true	"Pedido de cliente vip"	1301.30
33	99	1	Creusa	false	false	false	true	"Pedido de cliente vip"	128.70
34	100	1	Creusa	false	false	false	true	"Pedido de cliente vip"	130.00
35	999	1	Creusa	false	false	false	true	"Pedido de cliente vip"	1298.70
36	1001	1	Creusa	false	false	false	true	"Pedido de cliente vip"	1301.30
37	99	1	Maria	true	false	false	true	"Pedido de cliente comum"	160.00
38	100	1	Maria	true	false	false	true	"Pedido de cliente comum"	1598.40
39	999	1	Maria	false	true	true	true	"Pedido de cliente comum"	160.00
40	1001	1	Maria	false	true	true	true	"Pedido de cliente comum"	1598.40
41	99	1	Penna	true	true	false	false	"Pedido não vale a pena"	'#notpresent'
42	100	0	Penna	true	true	true	false	"Pedido não vale a pena"	'#notpresent'
43	999	0	Penna	true	true	true	false	"Pedido não vale a pena"	'#notpresent'
44	99	1	Penna	true	true	true	false	"Pedido não vale a pena"	'#notpresent'
45	1001	1	Penna	true	true	true	false	"Pedido não vale a pena"	'#notpresent'
46	99	1	Frias	false	true	false	false	"Pedido de cliente ruim"	'#notpresent'
47	100	1	Frias	false	true	false	false	"Pedido de cliente ruim"	'#notpresent'
48	999	1	Frias	false	true	false	false	"Pedido de cliente ruim"	'#notpresent'
49	99	1	Frias	false	true	false	false	"Pedido de cliente ruim"	'#notpresent'
50	1001	1	Frias	false	true	false	false	"Pedido de cliente ruim"	'#notpresent'
51	99	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
52	1001	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
53	99	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
54	100	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
55	999	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
56	99	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
57	1001	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
58	99	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'
59	1001	1	Frias	false	false	false	false	"Pedido de cliente ruim"	'#notpresent'

Registrar Pedidos (Scenario Outline)



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** RegistrarPedidosOutline.feature - brigadeiros - Visual Studio Code.
- Explorer:** Shows the project structure under "BRIGADEIROS":
 - diagrams
 - features
 - AvaliarPedidoValido.feature
 - AvaliarValorPedido.feature
 - E2EPedidoInvalido.feature
 - E2EPedidoValido.feature
 - RegistrarPedido.feature
 - RegistrarPedidosOutline.feature
 - rest
 - target
 - karate.jar
- Editor:** Displays the content of "RegistrarPedidosOutline.feature". A blue arrow points to the "Karate: Run" button in the top right of the editor area.
- Content:** The feature file contains a Scenario Outline for "Registrar Pedidos com sucesso". It includes Given steps for a URL and a path, a request body with variables, and a When step for a POST method. There are also Then and And steps for status codes and response values.
- Bottom Status Bar:** Shows "main" and "Karate" along with other status indicators.
- Bottom Right:** Status bar with "Ln 3, Col 16" and other file information.

Registrar Pedidos (Scenario Outline)

The screenshot shows a user interface for a testing or reporting tool. On the left, there's a sidebar with a character icon, a summary box (19 green, 0 red), and a list of scenarios. The main area displays a scenario outline table.

Scenarios:

- [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: *features/RegistrarPedidosOutline.feature* | 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
{	
"variables": {	
"quantidadePedida": {"value": 99, "type": "Integer"},	
"prazoPedido": {"value": 0, "type": "Integer"},	
"nomeCliente": {"value": Creusal, "type": "String"},	
"indicacao": {"value": false, "type": "Boolean"},	
"comprou": {"value": false, "type": "Boolean"},	
"pagou": {"value": false, "type": "Boolean"}},	
},	
"withVariablesInReturn": true	
}	
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 shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AvaliarPedidoValido.feature - brigadeiros - Visual Studio Code.
- Explorer:** Shows the project structure under 'BRIGADEIROS':
 - diagrams
 - features
 - AvaliarPedidoValido.feature (selected)
 - AvaliarValorPedido.feature
 - E2EPedidoInvalido.feature
 - E2EPedidoValido.feature
 - RegistrarPedido.feature
 - RegistrarPedidosOutline.feature
 - rest
 - target
 - karate.jar
- Editor:** Displays the content of 'AvaliarPedidoValido.feature'. The 'Feature' section is highlighted. The 'Background' section contains:

```
Given path "/decision-definition/key/PedidoValidoDecision/evaluate"
Given request
"""
{
    "variables": {
        "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
        "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
        "nomeCliente": {"value": <nomeCliente>, "type": "String"},
        "indicacao": {"value": <indicacao>, "type": "Boolean"},
        "comprou": {"value": <comprou>, "type": "Boolean"},
        "pagou": {"value": <pagou>, "type": "Boolean"}
    }
}
"""
When method POST
Then status 200
And match $[0].pedidoValido.value == <pedidoValido>
And match $[0].msgValidacaoPedido.value == <msgValidacaoPedido>
```

The 'Examples' section at the bottom is also highlighted.- Bottom Status Bar:** main, Karate, Ln 6, Col 51, Spaces: 4, UTF-8, CRLF, Karate.

Avaliar Pedido (Scenario Outline)

File Edit Selection View Go Run Terminal Help

AvaliarPedidoValido.feature - brigadeiros - Visual Studio Code

features > AvaliarPedidoValido.feature

```
When method POST
Then status 200
And match ${0}.pedidoValido.value == <pedidoValido>
And match ${0}.msgValidacaoPedido.value == <msgValidacaoPedido>

Examples: Pedidos válidos
| quantidadePedido | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido |
| 99 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 100 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 999 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 1001 | 0 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 99 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 100 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 999 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 1001 | 1 | Creusa | false | false | false | true | "Pedido de cliente vip" |
| 100 | 1 | Maria | true | false | false | true | "Pedido de cliente comum" |
| 999 | 1 | Maria | true | false | false | true | "Pedido de cliente comum" |
| 100 | 1 | Maria | false | true | true | true | "Pedido de cliente comum" |
| 999 | 1 | Maria | false | true | true | true | "Pedido de cliente comum" |

Examples: Pedidos inválidos
| quantidadePedido | prazoPedido | nomeCliente | indicacao | comprou | pagou | pedidoValido | msgValidacaoPedido |
| 100 | 0 | Penna | true | true | true | false | "Pedido não vale a pena" |
| 999 | 0 | Penna | true | true | true | false | "Pedido não vale a pena" |
| 99 | 1 | Penna | true | true | true | false | "Pedido não vale a pena" |
| 1001 | 1 | Penna | true | true | true | false | "Pedido não vale a pena" |
| 100 | 1 | Frias | true | true | false | false | "Pedido de cliente ruim" |
| 100 | 1 | Frias | false | true | false | false | "Pedido de cliente ruim" |
| 100 | 1 | Frias | false | false | false | false | "Pedido de cliente ruim" |
```

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

Avaliar Pedido (Scenario Outline)



AvaliarPedidoValido.feature - brigadeiros - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

> OPEN EDITORS

> BRIGADEIROS

> diagrams

> features

 AvaliarPedidoValido.feature

 AvaliarValorPedido.feature

 E2EPedidoValido.feature

 E2EPedidoValido.feature

 RegistrarPedido.feature

 RegistrarPedidosOutline.feature

> rest

> target

karate.jar

AvaliarPedidoValido.feature X

features > AvaliarPedidoValido.feature

Karate: Run | Karate: Debug

Feature: Avaliar Pedido

Background:

```
Given url "http://localhost:8080/engine-rest"
```

Karate: Run | Karate: Debug

Scenario Outline: Avaliar Pedidos com sucesso

```
Given path "/decision-definition/key/PedidoValidoDecision/evaluate"
Given request
"""
{
  "variables": {
    "quantidadePedido": {"value": <quantidadePedido>, "type": "Integer"},
    "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
    "nomeCliente": {"value": <nomeCliente>, "type": "String"},
    "indicacao": {"value": <indicacao>, "type": "Boolean"},
    "comprou": {"value": <comprou>, "type": "Boolean"},
    "pagou": {"value": <pagou>, "type": "Boolean"}
  }
"""
When method POST
Then status 200
And match ${[0].pedidoValido.value == <pedidoValido>}
And match ${[0].msgValidacaoPedido.value == <msgValidacaoPedido>}
```

Examples: Pedidos válidos

quantidadePedido	prazoPedido	nomeCliente	indicacao	comprou	pagou	pedidoValido	msgValidacaoPedido
99	0	Creusa	false	false	false	true	"Pedido
100	0	Creusa	false	false	false	true	"Pedido

main Karate 19

Ln 50, Col 1 Spaces: 4 UTF-8 CRLF Karate

Avaliar Pedido (Scenario Outline)

The screenshot shows a JBehave reporting interface with a sidebar on the left and a main summary table on the right.

Left Sidebar:

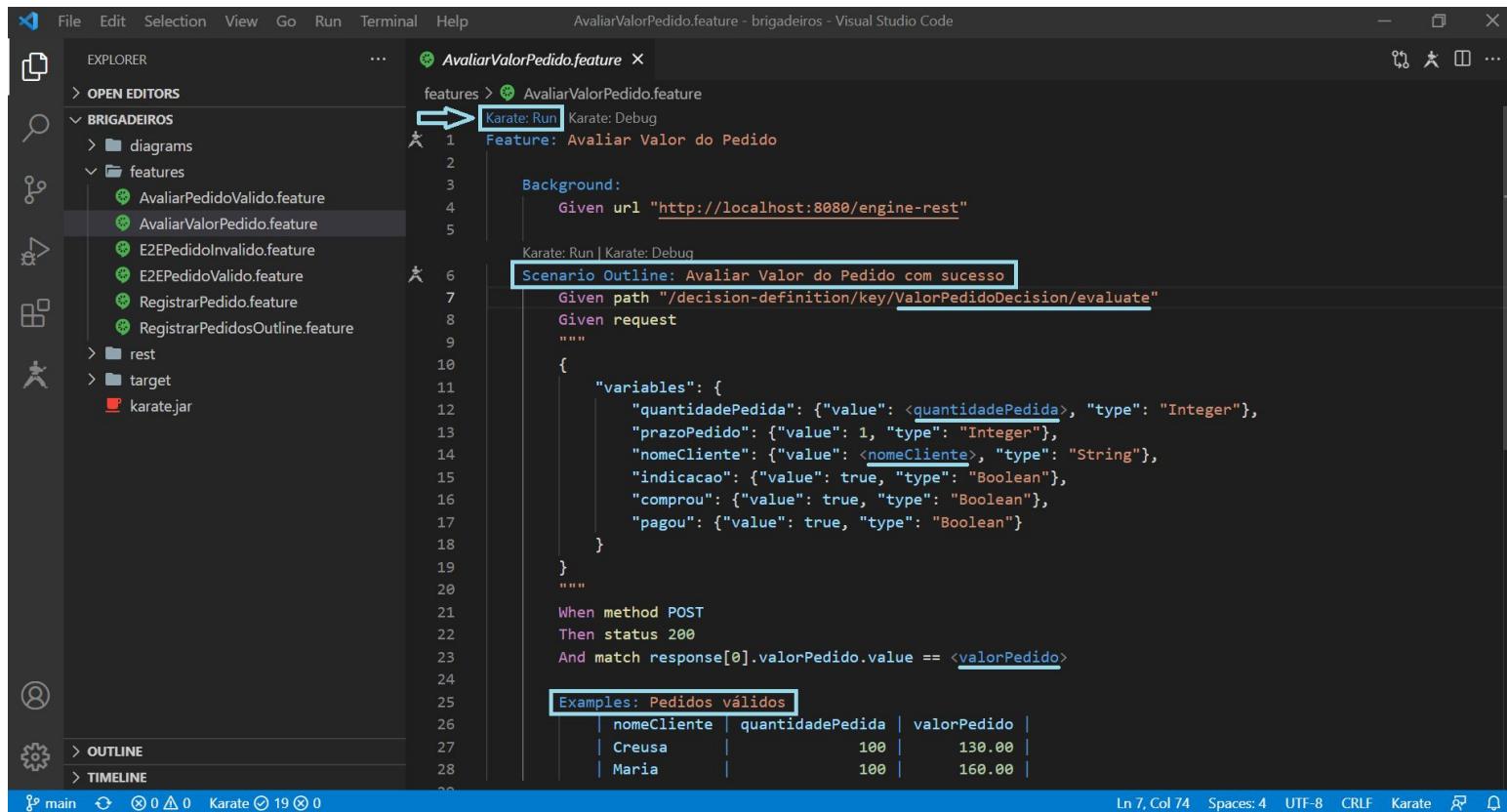
- Icon: Karate player
- Score: 19 (green)
- Score: 0 (red)
- Section: Scenarios
- List:
 - [1.1:28] Avaliar Pedidos com sucesso
 - [1.2:29] Avaliar Pedidos com sucesso
 - [1.3:30] Avaliar Pedidos com sucesso
 - [1.4:31] Avaliar Pedidos com sucesso
 - [1.5:32] Avaliar Pedidos com sucesso
 - [1.6:33] Avaliar Pedidos com sucesso
 - [1.7:34] Avaliar Pedidos com sucesso
 - [1.8:35] Avaliar Pedidos com sucesso

Main Summary Table:

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
{	"variables": { "quantidadePedida": {"value": 99, "type": "Integer"}, "prazoPedido": {"value": 0, "type": "Integer"}, "nomeCliente": {"value": Creusal, "type": "String"}, "indicacao": {"value": false, "type": "Boolean"}, "comprou": {"value": false, "type": "Boolean"}, "pagou": {"value": false, "type": "Boolean"} } }	
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)



AvaliarValorPedido.feature - brigadeiros - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
BRIGADEIROS
diagrams
features
AvaliarPedidoValido.feature
AvaliarValorPedido.feature
E2EPedidoInvalido.feature
E2EPedidoValido.feature
RegistrarPedido.feature
RegistrarPedidosOutline.feature
rest
target
karate.jar

Karate: Run | Karate: Debug

Feature: Avaliar Valor do Pedido

Background:
Given url "http://localhost:8080/engine-rest"

Karate: Run | Karate: Debug

Scenario Outline: Avaliar Valor do Pedido com sucesso

```
Given path "/decision-definition/key/ValorPedidoDecision/evaluate"
Given request
"""
{
    "variables": {
        "quantidadePedida": {"value": <quantidadePedida>, "type": "Integer"},
        "prazoPedido": {"value": 1, "type": "Integer"},
        "nomeCliente": {"value": <nomeCliente>, "type": "String"},
        "indicacao": {"value": true, "type": "Boolean"},
        "comprou": {"value": true, "type": "Boolean"},
        "pagou": {"value": true, "type": "Boolean"}
    }
}
"""
When method POST
Then status 200
And match response[0].valorPedido.value == <valorPedido>
```

Examples: Pedidos válidos

nomeCliente	quantidadePedida	valorPedido
Creusa	100	130.00
Maria	100	160.00

Ln 7, Col 74 Spaces: 4 UTF-8 CRLF Karate

Avaliar Valor Pedido (Scenario Outline)

The screenshot shows a UI interface for a testing or reporting tool. On the left, there's a sidebar with a 'Scenarios' section containing two items: '[1.1:27] Avaliar Valor do Pedido com sucesso' and '[1.2:28] Avaliar Valor do Pedido com sucesso'. The main area displays two scenarios outlined in tables.

Scenario: [1.1:27] Avaliar Valor do Pedido com sucesso

Background:		ms: 1294
7	Given path "/decision-definition/key/ValorPedidoDecision/evaluate"	15
8	Given request	33

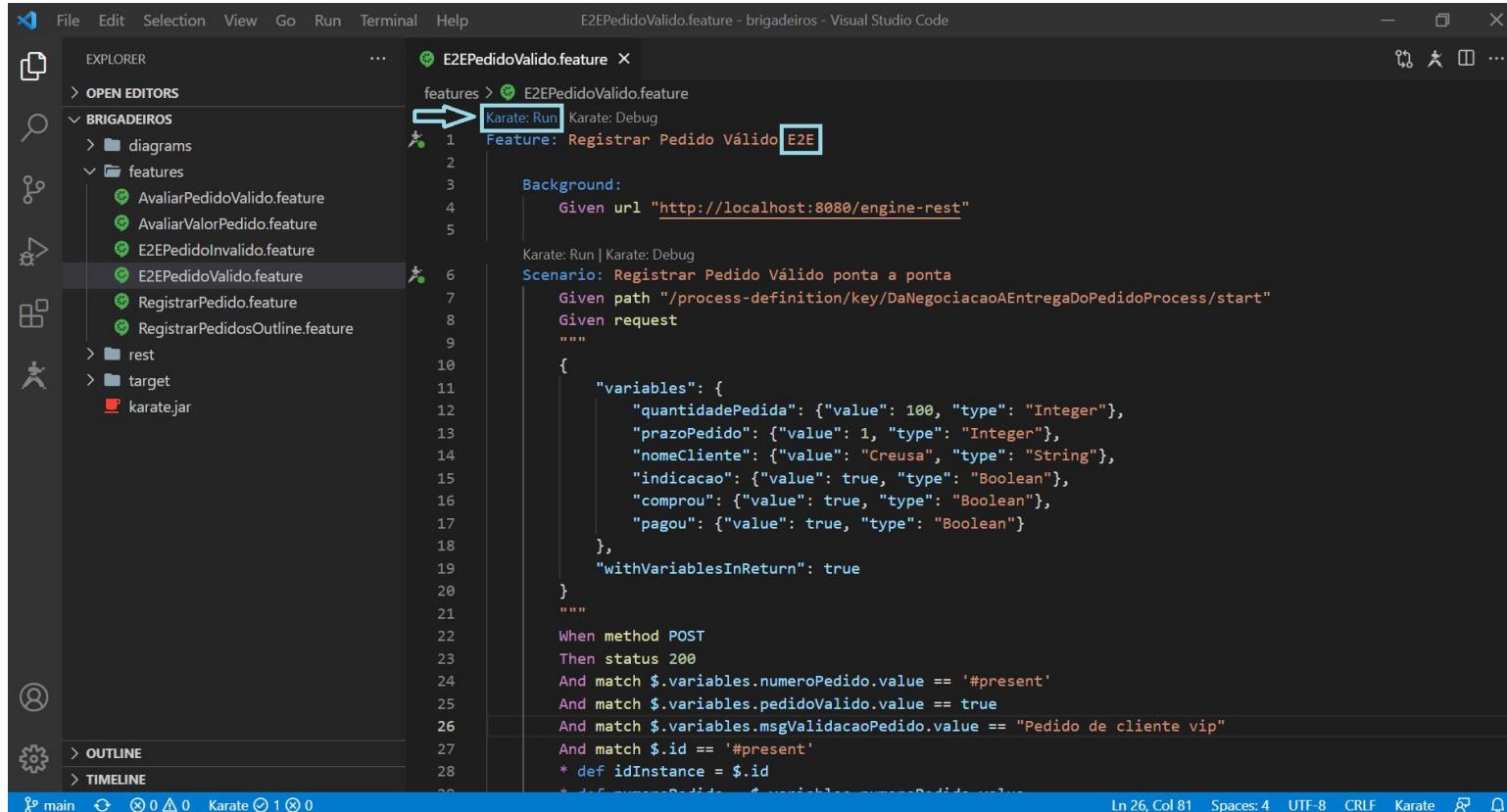
```
{  
    "variables": {  
        "quantidadePedida": {"value": 100, "type": "Integer"},  
        "prazoPedido": {"value": 1, "type": "Integer"},  
        "nomeCliente": {"value": Creusa, "type": "String"},  
        "indicacao": {"value": true, "type": "Boolean"},  
        "comprou": {"value": true, "type": "Boolean"},  
        "pagou": {"value": true, "type": "Boolean"}  
    }  
}  
21 When method POST  
22 Then status 200  
23 And match response[0].valorPedido.value == 130.00
```

Scenario: [1.2:28] Avaliar Valor do Pedido com sucesso

Background:		ms: 99
7	Given path "/decision-definition/key/ValorPedidoDecision/evaluate"	0
8	Given request	0

```
21 When method POST  
22 Then status 200  
23 And match response[0].valorPedido.value == 160.00
```

Registrar Pedido Válido (ponta-a-ponta)



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Title Bar:** E2EPedidoValido.feature - brigadeiros - Visual Studio Code
- Left Sidebar (Explorer):**
 - OPEN EDITORS
 - BRIGADEIROS
 - diagrams
 - features
 - AvaliarPedidoValido.feature
 - AvaliarValorPedido.feature
 - E2EPedidoInvalido.feature
 - E2EPedidoValido.feature** (highlighted with a blue arrow)
 - RegistrarPedido.feature
 - RegistrarPedidosOutline.feature
 - rest
 - target
 - karate.jar
- Central Area (Editor):**

Karate: Run | Karate: Debug

Feature: Registrar Pedido Válido [E2E]

```
Background:
    Given url "http://localhost:8080/engine-rest"

Karate: Run | Karate: Debug
Scenario: Registrar Pedido Válido ponta a ponta
    Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"
    Given request
    """
    {
        "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"}
        },
        "withVariablesInReturn": true
    }
    """
    When method POST
    Then status 200
    And match $.variables.numeroPedido.value == '#present'
    And match $.variables.pedidoValido.value == true
    And match $.variables.msgValidacaoPedido.value == "Pedido de cliente vip"
    And match $.id == '#present'
    * def idInstance = $.id
    * def msgValidacaoPedido = $.msgValidacaoPedido
    * def numeroPedido = $.variables.numeroPedido
    * def pedidoValido = $.variables.pedidoValido
    * def status = $.status
```
- Bottom Status Bar:** main, Karate, In 26, Col 81, Spaces: 4, UTF-8, CRLF, Karate

Registrar Pedido Válido (ponta-a-ponta)

The screenshot shows a test report for a valid order registration scenario. On the left, there's a sidebar with a karate icon, a summary table (1 success, 0 failures), and a list of scenarios. The main area displays the test steps and their execution details.

Scenarios		Summary Tags Feature: features/E2EPedidoValido.feature Registrar Pedido Válido E2E
		Scenario: [1:6] Registrar Pedido Válido ponta a ponta
>> Background:		ms: 2892
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

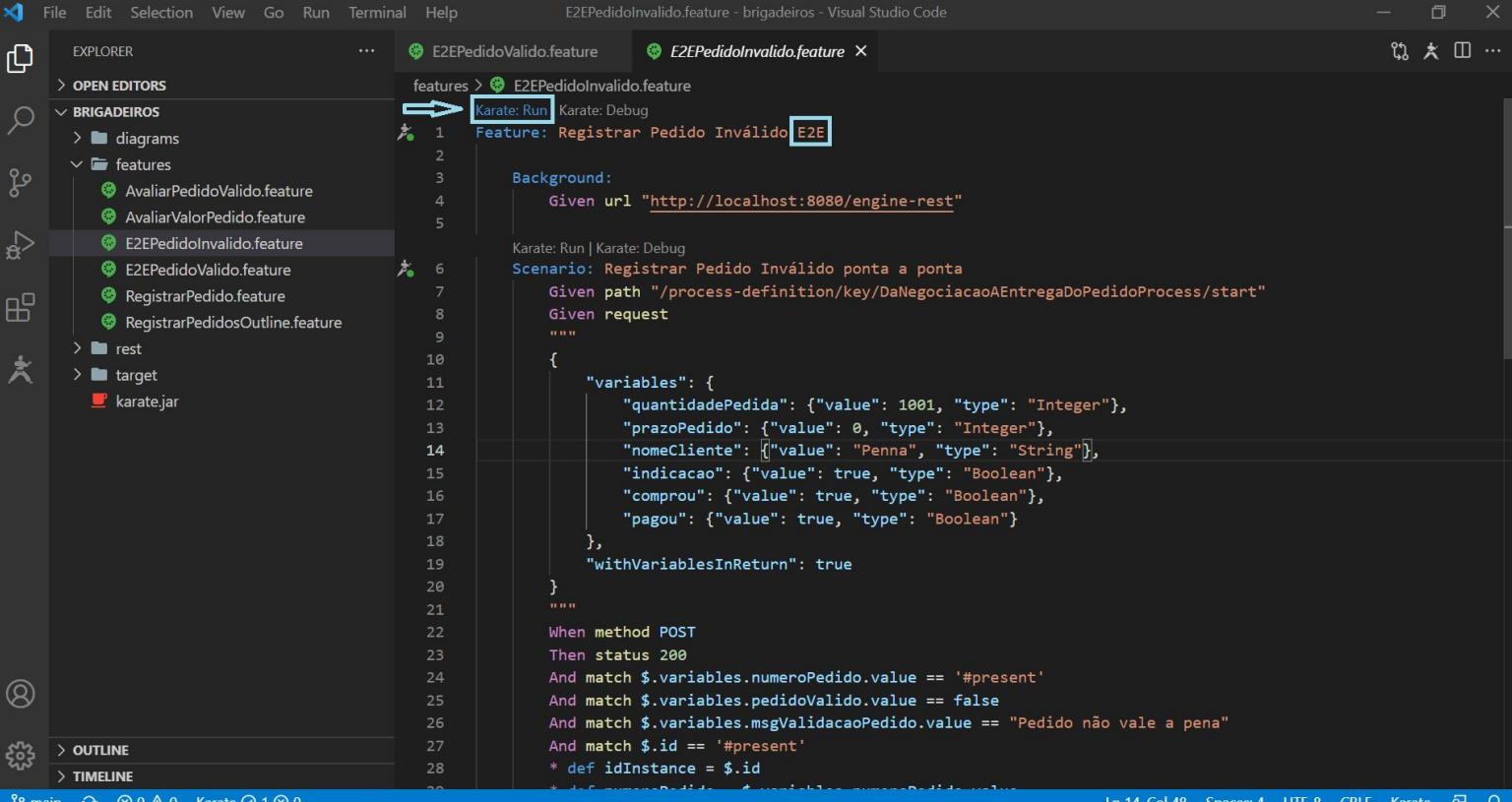
Annotations in the screenshot:

- A green box highlights the first set of steps (7-31). A green arrow points from the text "Iniciar instância" to step 7.
- A green box highlights the second set of steps (33-46). A green arrow points from the text "Verificar posição da instância" to step 33.
- A green box highlights the third set of steps (40-46). A green arrow points from the text "Completar tarefa" to step 40.

Registrar Pedido Válido (ponta-a-ponta)

Scenarios	1	0
[1:6] Registrar Pedido Válido ponta a ponta		
	Given path "/task/" + idTask + "/complete"	2
	And header Content-Type = 'application/json'	2
	When method POST	15
	Then status 204	0
	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	0
	When method GET	19
	Then status 200	0
	And match \$[0].id == '#present'	0
	And match \$[0].taskDefinitionKey == 'FabricarPedidoTask'	1
	* def idTask = \$[0].id	1
	Given path "/task/" + idTask + "/complete"	0
	And header Content-Type = 'application/json'	0
	When method POST	35
	Then status 204	0
	Given path "/task?processVariables=numeroPedido_eq_" + numeroPedido	0
	When method GET	29
	Then status 200	0
	And match \$[0].id == '#present'	1
	And match \$[0].taskDefinitionKey == 'EntregarPedidoTask'	1
	* def idTask = \$[0].id	1
	Given path "/task/" + idTask + "/complete"	0
	And header Content-Type = 'application/json'	0
	When method POST	28
	Then status 204	0
	Given path "/history/process-instance/" + idInstance	1
	When method GET	17
	Then status 200	0
	And match \$.state == 'COMPLETED'	1

Registrar Pedido Inválido (ponta-a-ponta)



The screenshot shows a Visual Studio Code interface with the following details:

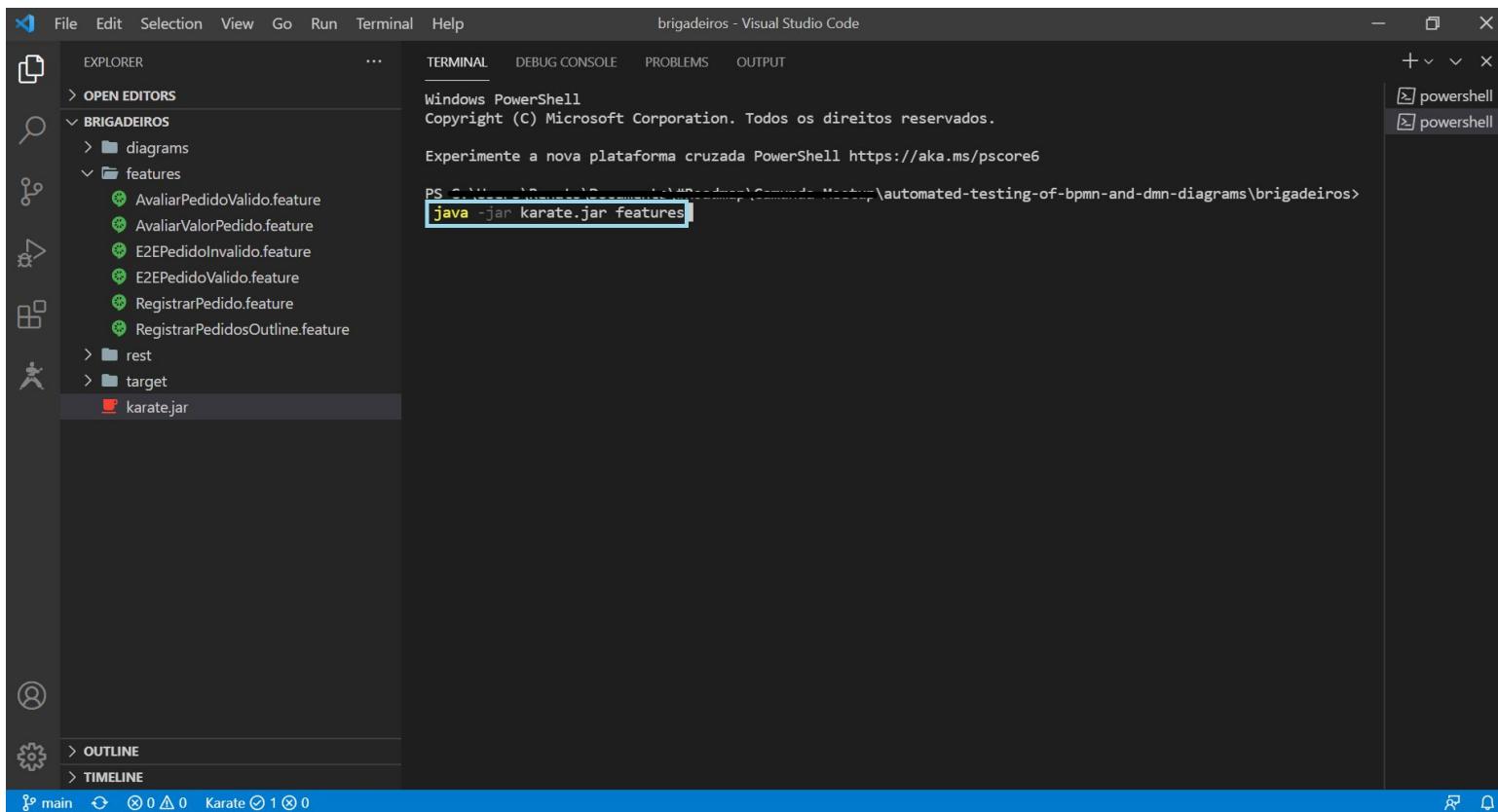
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Title Bar:** E2EPedidoInvalido.feature - brigadeiros - Visual Studio Code
- Explorer:** Shows a tree view of files under the BRIGADEIROS folder, including E2EPedidoValido.feature, AvaliarPedidoValido.feature, AvaliarValorPedido.feature, and others.
- Editor:** The main pane displays a Karate feature file named E2EPedidoInvalido.feature. A red arrow points to the "Karate: Run" button in the toolbar above the editor.
- Content:** The feature file defines a scenario for registering an invalid order. It includes a background section with a given URL, a scenario title, and a step definition for a POST method with specific assertions about response status and variable values.
- Bottom Status Bar:** Shows file statistics (main 0, Karate 1) and other developer information (Ln 14, Col 48, Spaces: 4, UTF-8, CRLF, Karate).

```
Feature: Registrar Pedido Inválido [E2E]
  Background:
    Given url "http://localhost:8080/engine-rest"
  Scenario: Registrar Pedido Inválido ponta a ponta
    Given path "/process-definition/key/DaNegociacaoAEntregaDoPedidoProcess/start"
    Given request
    """
    {
      "variables": {
        "quantidadePedida": {"value": 1001, "type": "Integer"},
        "prazoPedido": {"value": 0, "type": "Integer"},
        "nomeCliente": [{"value": "Penna", "type": "String"}],
        "indicacao": {"value": true, "type": "Boolean"},
        "comprou": {"value": true, "type": "Boolean"},
        "pagou": {"value": true, "type": "Boolean"}
      },
      "withVariablesInReturn": true
    }
    """
    When method POST
    Then status 200
    And match $.variables.numeroPedido.value == '#present'
    And match $.variables.pedidoValido.value == false
    And match $.variables.msgValidacaoPedido.value == "Pedido não vale a pena"
    And match $.id == '#present'
    * def idInstance = $.id
    * def id = idInstance
```

Registrar Pedido Inválido (ponta-a-ponta)

Scenarios		Summary Tags Feature: features/E2EPedidoInvalido.feature Registrar Pedido Inválido E2E
 1		
0		
[1:6] Registrar Pedido Inválido ponta a ponta		ms: 2309
Scenario: [1:6] Registrar Pedido Inválido ponta a ponta		
>> 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

Suite de tests completa



Suite de testes completa

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "BRIGADEIROS".
 - "features" folder contains: AvaliarPedidoValido.feature, AvaliarValorPedido.feature, E2EPedidoInvalido.feature, E2EPedidoValido.feature, RegistrarPedido.feature, RegistrarPedidosOutline.feature.
 - "rest" folder.
 - "target" folder.
 - "karate.jar" file.
- Terminal:** Displays the output of a Karate test run.

```
1 < 200
1 < Content-Type: application/json
1 < Transfer-Encoding: chunked
1 < Date: Mon, 27 Jun 2021 01:00:00 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
=====
01:00:00.000 [main] INFO com.intuit.karate.Suite - <>pass>> feature 6 of 6 (0 remaining) features/RegistrarPedidosOutline.feature
01:00:00.000 [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
=====
```
- Output Panel:** Shows a summary of the test results:

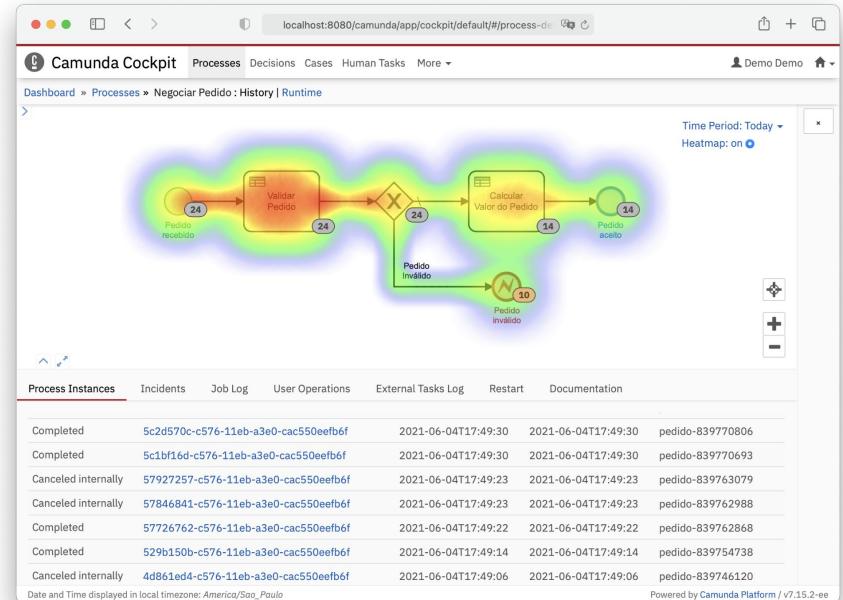
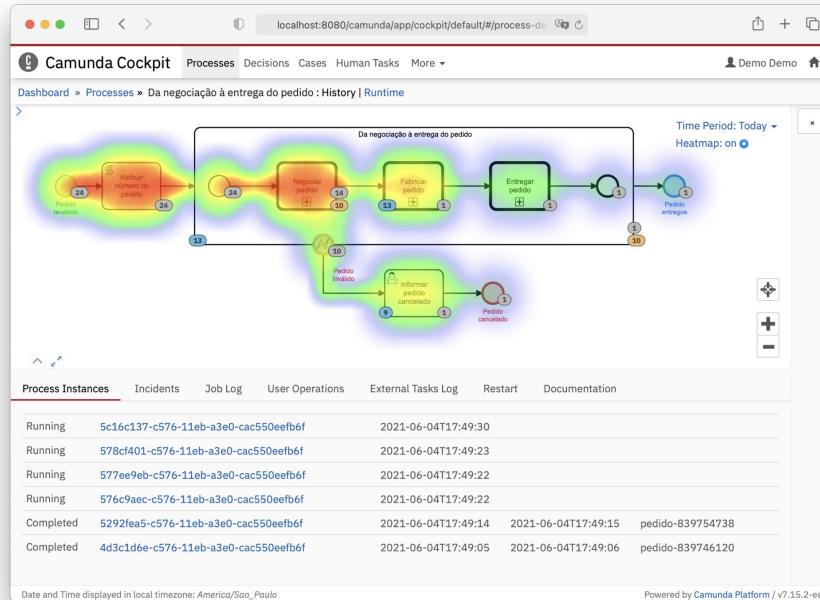
```
HTML report: (paste into browser to view) | Karate version: 1.1.0.RC1
file:///C:/Users/.../Desktop/.../automated-testing-of-bpmn-and-dmn-diagrams/brigadeiros/target/karate-reports/karate-summary.html
=====
```
- Bottom Status Bar:** Shows "PS C:\Users\...\Desktop\...\automated-testing-of-bpmn-and-dmn-diagrams\brigadeiros>".

Suite de testes completa



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



<https://docs.camunda.org/manual/latest/webapps/cockpit/bpmn/process-history-views/>
<https://camunda.com/products/camunda-platform/cockpit/#features>

Histórico de Execução das Regras de Decisão

localhost:8080/camunda/app/cockpit/default#/decision-re

Camunda Cockpit

Processes Decisions Cases Human Tasks More

Demo Demo

Dashboard » Decisions » Regras de Negócio

Decision Requirement Definition Instances

ID	Data	Nome	Processo	Valor
5c9407c6-c576-11eb-a3e0...	2021-06-04T17:49:31	ValorPedidoDecision	NegociarPedidoProcess	5c8fe8e5-c576-11eb-a3e0...
5c9e4156-c576-11eb-a3e...	2021-06-04T17:49:31	PedidoValidoDecision	NegociarPedidoProcess	5c9da504-c576-11eb-a3e0...
5ca21205-c576-11eb-a3e...	2021-06-04T17:49:31	ValorPedidoDecision	NegociarPedidoProcess	5c9da504-c576-11eb-a3e0...
5cab6035-c576-11eb-a3e0...	2021-06-04T17:49:31	PedidoValidoDecision	NegociarPedidoProcess	5caeaf3-c576-11eb-a3e0...
5cae6d94-c576-11eb-a3e0...	2021-06-04T17:49:31	ValorPedidoDecision	NegociarPedidoProcess	5caeaf3-c576-11eb-a3e0...

First Previous 1 2 Next Last

Date and Time displayed in local timezone: America/Sao_Paulo

Powered by Camunda Platform / v7.15.2-ee

localhost:8080/camunda/app/cockpit/default#/decision-in

Camunda Cockpit

Processes Decisions Cases Human Tasks More

Demo Demo

Dashboard » Decisions » Pedido Válido » 3797cca4-c576-11eb-a3e0-cac550eefb6

View DRD

Pedido Válido Hit Policy: Unique

When = VIP	And = false	Then	And	Mensagem
Type Cliente	Vale a pena	Pedido Válido	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"	-
4 "Rúim"		false	"Pedido de cliente ruim"	-

Inputs Outputs

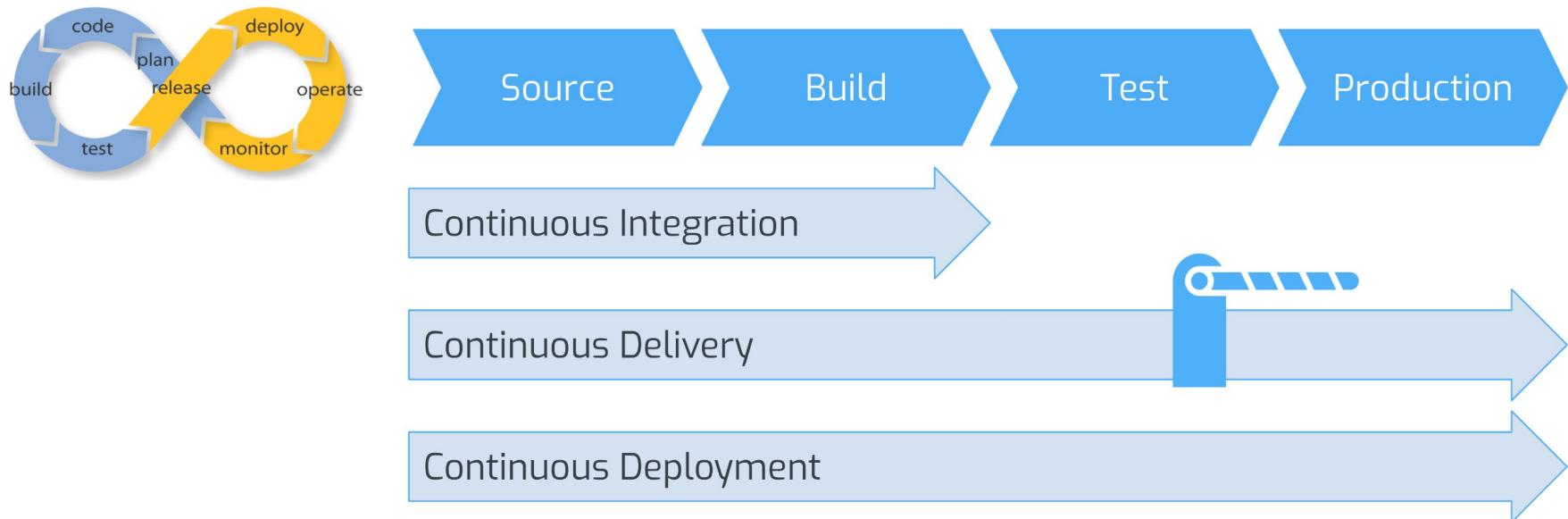
Name	Type	Value
Vale a pena	Boolean	false
Tipo Cliente	String	VIP

Date and Time displayed in local timezone: America/Sao_Paulo

Powered by Camunda Platform / v7.15.2-ee

<https://docs.camunda.org/manual/latest/webapps/cockpit/dmn/decision-requirements-definition-view/>
<https://camunda.com/products/camunda-platform/cockpit/#features>

Continuous Integration and Delivery (CI / CD)

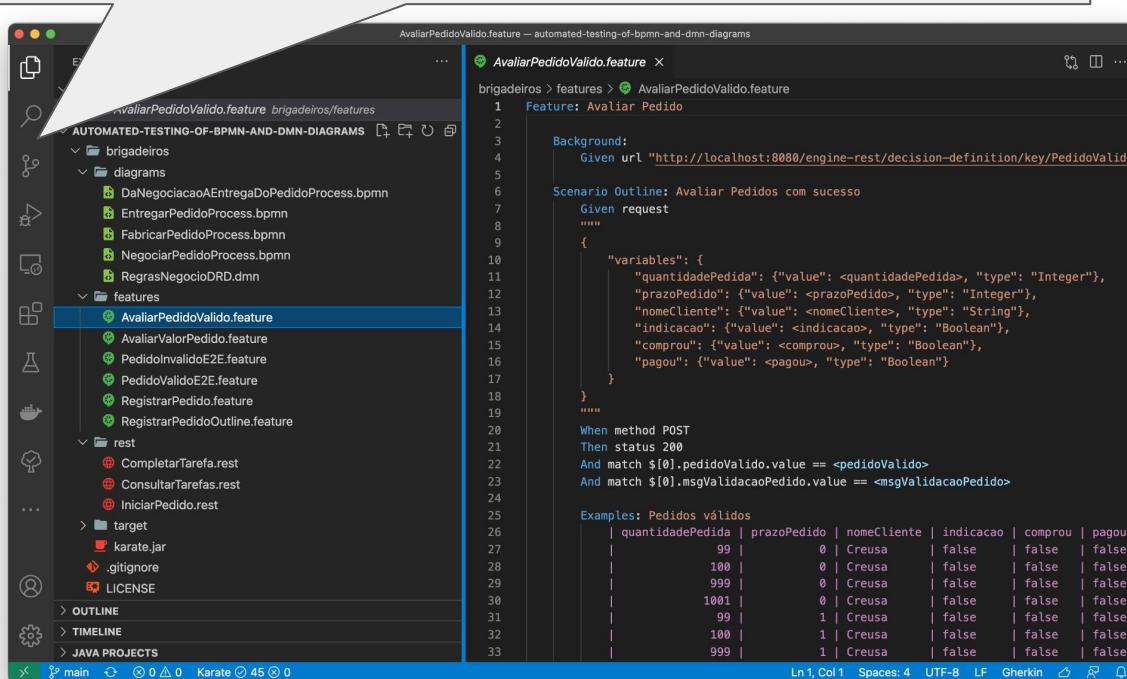


<https://www.roadmap.com.br/om>

<https://dzone.com/articles/what-is-cicd>

Compartilhamento com a Comunidade de Usuários

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



The screenshot shows a terminal window with the following content:

```
AvaliarPedidoValido.feature -- automated-testing-of-bpmn-and-dmn-diagrams
AvaliarPedidoValido.feature
brigadeiros > features > AvaliarPedidoValido.feature
  Feature: Avaliar Pedido
    Background:
      Given url "http://localhost:8080/engine-rest/decision-definition/key/PedidoValido"
    Scenario Outline: Avaliar Pedidos com sucesso
      Given request
      """
      {
        "variables": {
          "quantidadePedido": {"value": <quantidadePedido>, "type": "Integer"},
          "prazoPedido": {"value": <prazoPedido>, "type": "Integer"},
          "nomeCliente": {"value": <nomeCliente>, "type": "String"},
          "indicacao": {"value": <indicacao>, "type": "Boolean"},
          "comprou": {"value": <comprou>, "type": "Boolean"},
          "pagou": {"value": <pagou>, "type": "Boolean"}
        }
      }
      """
      When method POST
      Then status 200
      And match ${0}.pedidoValido.value == <pedidoValido>
      And match ${0}.msgValidacaoPedido.value == <msgValidacaoPedido>
    Examples: Pedidos válidos
      | quantidadePedido | prazoPedido | nomeCliente | indicacao | comprou | pagou |
      | 99 | 0 | Creusa | false | false | false |
      | 100 | 0 | Creusa | false | false | false |
      | 999 | 0 | Creusa | false | false | false |
      | 1001 | 0 | Creusa | false | false | false |
      | 99 | 1 | Creusa | false | false | false |
      | 100 | 1 | Creusa | false | false | false |
      | 999 | 1 | Creusa | false | false | false |
```



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

Projeto no GitHub

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for Why GitHub?, Team, Enterprise, Explore, Marketplace, and Pricing. Below the navigation is a search bar and buttons for Sign in and Sign up. The main title of the repository is "mbitencourt / automated-testing-of-bpmn-and-dmn-diagrams". The repository has 0 stars and 0 forks. The "Code" tab is selected, showing a main branch with 58 commits from "carlosrenatoramos". The commit list includes "Brigadeiros da Nice" (3d9de35, 15 hours ago), "Brigadeiros da Nice" (initial commit, 15 hours ago), ".gitignore" (initial commit, 4 days ago), "LICENSE" (initial commit, 11 days ago), and "README.md" (initial commit, 11 days ago). Below the code area is a "README.md" section containing the project title and a brief description: "Automated testing of BPMN and DMN diagrams". On the right side of the page, there are sections for "About", "Releases", "Packages", and "Contributors". The "About" section describes the project as "Automated testing of BPMN and DMN diagrams". The "Releases" section indicates "No releases published". The "Packages" section says "No packages published". The "Contributors" section lists "carlosrenatoramos" and "mbitencourt Maurício Bitencourt".



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

Gravação desta apresentação

The screenshot shows a LinkedIn event page. At the top, it displays the event details: "Remote Camunda Meetup #3 - 08 Jun" and "Camunda User Group BRAZIL". Below this, the title is "Camunda User Group Brazil - Remote Meetup #3 - 8 jun 2021" organized by Maurício Bitencourt. It indicates the event is online and scheduled for June 8, 2021, from 10:00 AM to 12:00 PM. A link to the YouTube video is provided: "Join here - https://www.youtube.com/watch?v=2XkFqSbHqHA". There are 99+ attendees listed, including Marcelo Eduardo Schubert, Lucas Chaves, and 424 other attendees. Buttons for "Invite connections" and "Share" are visible. The "Details" tab is selected. In the "About" section, it states that the meetings aim to connect users in Brazil to those interested in process automation, distributed transactions, and microservices. It welcomes IT professionals and business managers. The "Speakers" section is partially visible at the bottom.

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

The screenshot shows a YouTube channel page for "Camunda User Group BRAZIL". The channel has 83 subscribers. The main video thumbnail features a cityscape with the text "Remote Camunda Meetup #3 - 08/Jun" and "CAMUNDA USER GROUP BRAZIL". Below the thumbnail, a button says "Ao vivo em 7 dias 8 de junho 10:00" and another says "Lembrete ativado". Underneath the video, there is a section titled "Principais mensagens" with a message from Wagner Sérgio Barbosa Veiga. A note at the bottom encourages users to "Curta o chat ao vivo! Não se esqueça de proteger sua privacidade e seguir nossas diretrizes da comunidade."

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