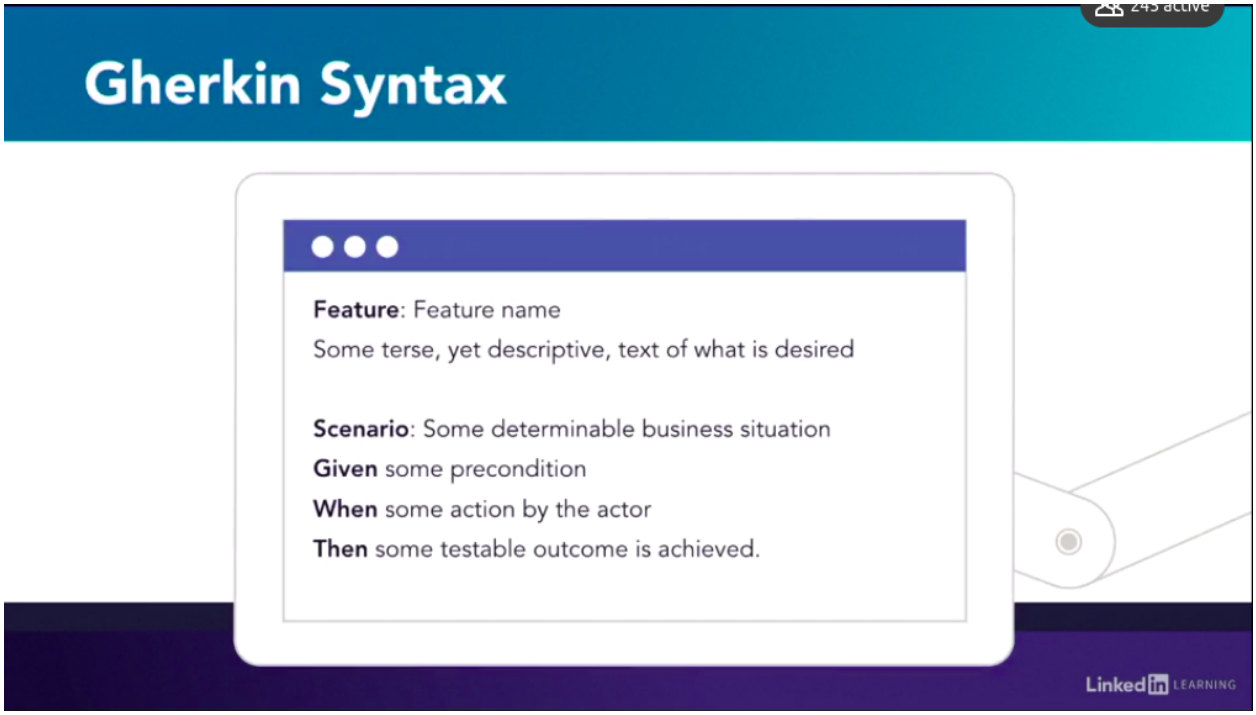
sudo snap install ruby --classic
sudo gem install bundler



sudo mkdir projectbdd
cd projectbdd/
code .

Create Gemfile that is the dependences of the project

```
#we will define the dependences
source 'https://rubygems.org'

gem 'cucumber', '~> 3.1.0'
gem 'rspec', '~> 3.7.0'
```

sudo bundle install
sudo cucumber --init

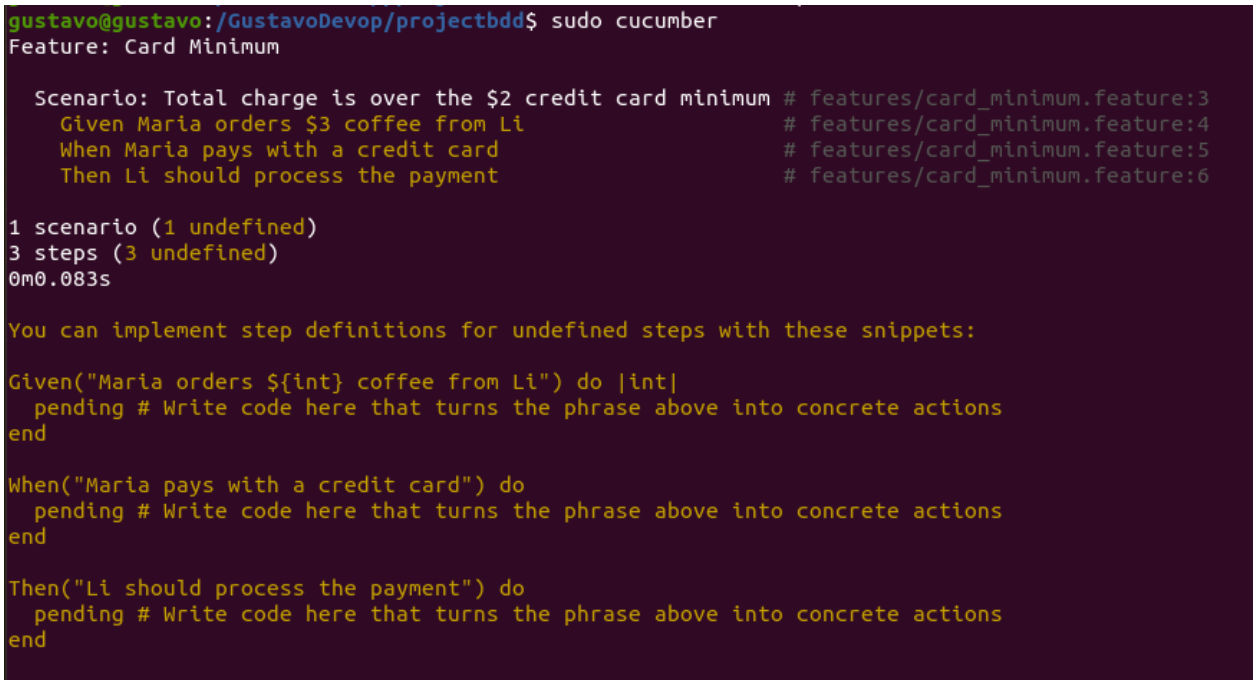
sudo chmod -R 777 ./

Create in the features directory the file card_minimum.feature and write the scenario
Feature: Card Minimum

  Scenario: Total charge is over the $2 credit card minimum
  Given Maria orders $3 coffee from Li
  When Maria pays with a credit card
  Then Li should process the payment

  Scenario: Total charge is under the $2 credit card minimum
  Given Maria orders $1 coffee from Li
  When Maria pays with a credit card
  Then Li should not process the payment

sudo cucumber # to show the scenario test as code



Create the steps.rb file in the steps directory to paste the code of the test

```
Given("Maria orders ${int} coffee from Li") do |int|
 pending # Write code here that turns the phrase above into concrete
actions
end

When("Maria pays with a credit card") do
 pending # Write code here that turns the phrase above into concrete
actions
end

Then("Li should process the payment") do
 pending # Write code here that turns the phrase above into concrete
actions
end
```

sudo cucumber # will match the steps with the scenario to run the bdd test

We will code the test first

```
require 'ourcode'
Given("Maria orders ${int} coffee from Li") do |int|
   @maria = Customer.new
   @li = Associate.new
   @price = int
   @card_minimum = 2
   @maria.orders
end
When("Maria pays with a credit card") do
   @maria.pays_with_card
end
Then("Li should process the payment") do
   expect(@li.process_payment(@price, @card_minimum)).to include(true)
end

Then("Li should not process the payment") do
```

```
   expect(@li.process_payment(@price, @card_minimum)).to include(false)
   expect(@li.process_payment(@price, @card_minimum)).to include('below
card minimum')
end
```

Then we code to the test to work
we create a lib directory inside the root folder
Then we create our script called ‘ourcode.rb’ with the following content

```
class Customer
   def orders
   end

   def pays_with_card
   end
end

class Associate
   def process_payment(price, card_minimum)
       if price >= card_minimum
           return [true, 'card processed']
       end
       if price < card_minimum
           return [false, 'below card minimum']
       end
   end
end
```

EMBRACE FAILURE!

Require our code in the test script and then run the test again to pass
Add the @ sign before the variables in ruby makes it visible in all the scopes

```
gustavo@gustavo:/GustavoDevop/projectbdd$ sudo cucumber
Feature: Card Minimum

  Scenario: Total charge is over the $2 credit card minimum # features/card_minimum.feature:3
    Given Maria orders $3 coffee from Li                    # features/step_definitions/steps.rb:3
    When Maria pays with a credit card                      # features/step_definitions/steps.rb:11
    Then Li should process the payment                      # features/step_definitions/steps.rb:15

  Scenario: Total charge is under the $2 credit card minimum # features/card_minimum.feature:8
    Given Maria orders $1 coffee from Li                     # features/step_definitions/steps.rb:3
    When Maria pays with a credit card                       # features/step_definitions/steps.rb:11
    Then Li should not process the payment                   # features/step_definitions/steps.rb:19
      expected [false, "below card minimum"] to include true
      Diff:
      @@ -1 +1 @@
      -[true]
      +[false, "below card minimum"]
       (RSpec::Expectations::ExpectationNotMetError)
      ./features/step_definitions/steps.rb:20:in `"Li should not process the payment"'
      features/card_minimum.feature:11:in `Then Li should not process the payment'

Failing Scenarios:
cucumber features/card_minimum.feature:8 # Scenario: Total charge is under the $2 credit card minimum

2 scenarios (1 failed, 1 passed)
6 steps (1 failed, 5 passed)
0m0.052s
/var/lib/gems/2.7.0/gems/cucumber-core-3.2.1/lib/cucumber/core/test/result.rb:12: warning: calling priva
y not have the intended effect
```

```
gustavo@gustavo:/GustavoDevop/projectbdd$ sudo cucumber
Feature: Card Minimum

  Scenario: Total charge is over the $2 credit card minimum # features/card_minimum.feature:3
    Given Maria orders $3 coffee from Li                    # features/step_definitions/steps.rb:3
    When Maria pays with a credit card                      # features/step_definitions/steps.rb:11
    Then Li should process the payment                      # features/step_definitions/steps.rb:15

  Scenario: Total charge is under the $2 credit card minimum # features/card_minimum.feature:8
    Given Maria orders $1 coffee from Li                     # features/step_definitions/steps.rb:3
    When Maria pays with a credit card                       # features/step_definitions/steps.rb:11
    Then Li should not process the payment                   # features/step_definitions/steps.rb:19

2 scenarios (2 passed)
6 steps (6 passed)
0m0.035s
/var/lib/gems/2.7.0/gems/cucumber-core-3.2.1/lib/cucumber/core/test/result.rb:12: warning: calling pr
y not have the intended effect
```