

Practica Calificada #1 de Desarrollo de Software

Integrantes:

- Espinoza Pari, Franklin

Parte 1

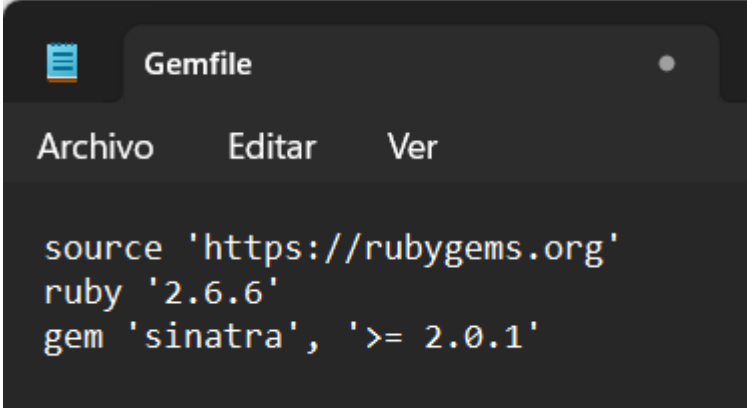
Vamos a trabajar en el directorio llamado `PrimeraPCDS` donde va a contener la aplicacion de este proyecto, ademas tambien creamos un repositorio Git llamado `PC1_CC3S2` .
Entonces realizamos los siguientes pasos:

```
user@LAPTOP-JPL0023Q MINGW64 ~/Desktop (main)
$ cd PrimeraPCDS

user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (main)
$ git init
Initialized empty Git repository in c:/Users/user/Desktop/PrimeraPCDS/.git/

user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (master)
$
```

Dentro de la carpeta creamos el archivo `Gemfile` (las mayusculas importa) sin ninguna extensión, donde vamos a insertar el contenido proporcionado:



```
source 'https://rubygems.org'
ruby '2.6.6'
gem 'sinatra', '>= 2.0.1'
```

Se usa `Gemfile` para especificar las dependencias de las gemas que tendra nuestra aplicacion, la fuente de instalacion de las gemas sera `http://rubygems.org` la siguiente linea indica la versión de ruby que va a necesitar nuestra aplicacion y finalmente `gem 'sinatra', '>= 2.0.1'` que especifica que nuestra aplicacion necesita la version 2.0.1 de la gema `sinatra`

Correr el Bundler

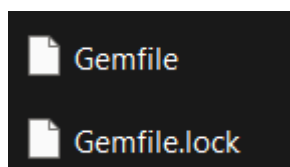
Vamos a ejecutar el comando `bundler` para verificar las dependencias del Gemfile con respecto a nuestra aplicacion

```
user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (master)
$ bundle
Your Ruby version is 3.2.2, but your Gemfile specified 2.6.6
```

Para instalar las dependencias usamos el comando `bundle install`

```
$ bundle install
Fetching gem metadata from https://rubygems.org/
....
Resolving dependencies...
Using bundler 1.17.2
Fetching ruby2_keywords 0.0.5
Installing ruby2_keywords 0.0.5
Fetching mustermann 3.0.0
Installing mustermann 3.0.0
Fetching rack 2.2.8
Installing rack 2.2.8
Fetching rack-protection 3.1.0
Installing rack-protection 3.1.0
Fetching tilt 2.3.0
Installing tilt 2.3.0
Fetching sinatra 3.1.0
Installing sinatra 3.1.0
Bundle complete! 1 Gemfile dependency, 7 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
```

Se creó el archivo `Gemfile.lock` en nuestro directorio de trabajo



Ponemos los comandos del git `git add .` y `git commit -m "Configurar el Gemfile"`

```

user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (master)
$ git add .
warning: in the working copy of '.ipynb_checkpoints/Respuestas-checkpoint.ipynb', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Gemfile.lock', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Respuestas.ipynb', LF will be replaced by CRLF the next time Git touches it

user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (master)
$ git commit -m "Configurar el Gemfile"
[master (root-commit) 474bf39] Configurar el Gemfile
 5 files changed, 214 insertions(+)
 create mode 100644 .ipynb_checkpoints/Respuestas-checkpoint.ipynb
 create mode 100644 Gemfile
 create mode 100644 Gemfile.lock
 create mode 100644 Readme.md
 create mode 100644 Respuestas.ipynb

```

Creamos el archivo `app.rb` con el siguiente contenido:

```

app.rb
1 require 'sinatra'
2 class MyApp < Sinatra::Base
3   get '/' do
4     "<!DOCTYPE html><html><head></head><body><h1>Hello World</h1></body></html>"
5   end
6 end

```

In []:

```

config.ru

Archivo  Editar  Ver

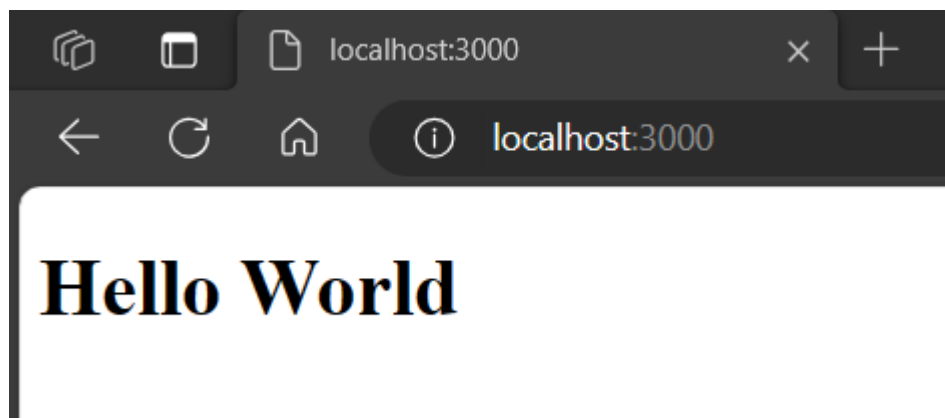
require './app'
run MyApp

```

Ejecutamos el siguiente comando `bundle exec rackup --port 3000`

```
user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (master)
$ bundle exec rackup --port 3000
[2023-09-27 15:14:25] INFO WEBrick 1.4.2
[2023-09-27 15:14:25] INFO ruby 2.6.6 (2020-03-31) [x64-mingw32]
[2023-09-27 15:14:25] INFO WEBrick::HTTPServer#start: pid=26812 port=3000
:::1 - - [27/Sep/2023:15:14:52 -0500] "GET / HTTP/1.1" 200 74 0.0337
:::1 - - [27/Sep/2023:15:14:52 -0500] "GET /favicon.ico HTTP/1.1" 404 503 0.0009
:::1 - - [27/Sep/2023:15:18:20 -0500] "GET / HTTP/1.1" 200 74 0.0007
:::1 - - [27/Sep/2023:15:18:21 -0500] "GET / HTTP/1.1" 200 74 0.0007
```

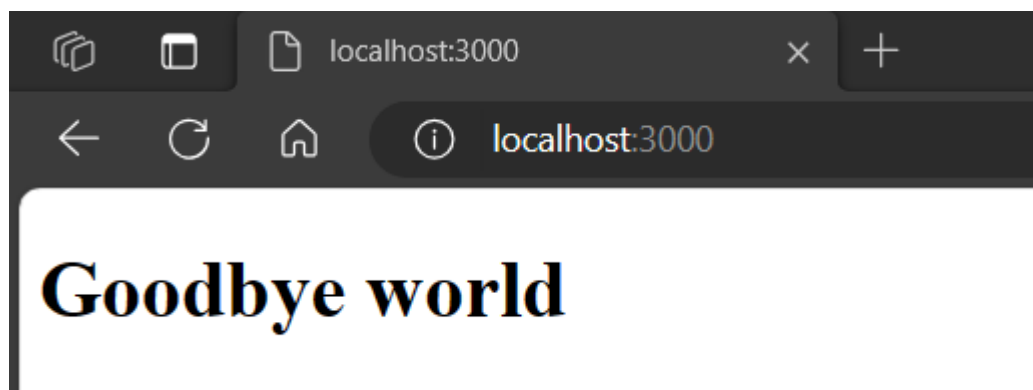
In []:



Cambiamos el "Hello Word" por "Good bye World"

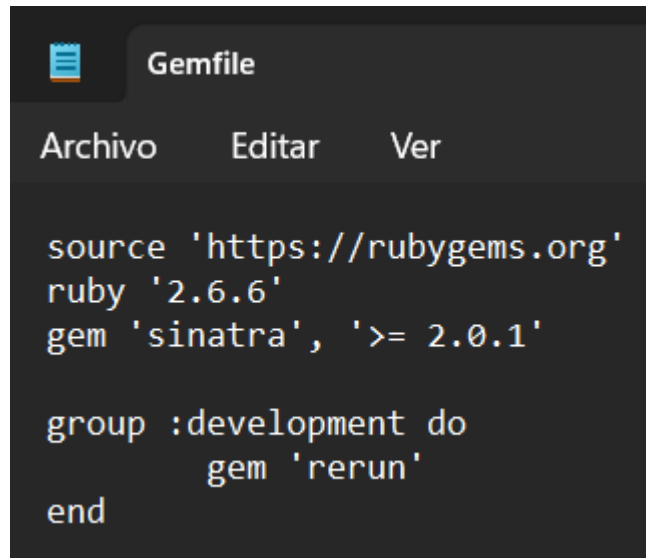
```
app.rb
1 require 'sinatra'
2 class MyApp < Sinatra::Base
3   get '/' do
4     "<!DOCTYPE html><html><head></head><body><h1>Goodbye world</h1></body></html>"
5   end
6 end
```

In []:



In []:

Agregamos el contenido adicional al Gemfile



```
Gemfile

Archivo  Editar  Ver

source 'https://rubygems.org'
ruby '2.6.6'
gem 'sinatra', '>= 2.0.1'

group :development do
  gem 'rerun'
end
```

Ejecutamos el comando `bundle install`

```
$ bundle install
Fetching gem metadata from https://rubygems.org/.....
Fetching gem metadata from https://rubygems.org/.
Resolving dependencies...
Using bundler 1.17.2
Fetching ffi 1.16.2 (x64-mingw32)
Installing ffi 1.16.2 (x64-mingw32)
Fetching rb-fsevent 0.11.2
Installing rb-fsevent 0.11.2
Fetching rb-inotify 0.10.1
Installing rb-inotify 0.10.1
Fetching listen 3.8.0
Installing listen 3.8.0
Using ruby2_keywords 0.0.5
Using mustermann 3.0.0
Using rack 2.2.8
Using rack-protection 3.1.0
Fetching rerun 0.14.0
Installing rerun 0.14.0
Using tilt 2.3.0
Using sinatra 3.1.0
Bundle complete! 2 Gemfile dependencies, 12 gems now installed
Use `bundle info [gemname]` to see where a bundled gem is installed.
```

Ahora ejecutamos el comando `bundle exec rerun -- rackup --port 3000`

```

user@LAPTOP-JPL00230 MINGW64 ~/Desktop/PrimeraPCDS (master)
$ bundle exec rerun -- rackup --port 3000
INFORMACIÓN: no se pudo encontrar ningún archivo para los patrones dados.
INFORMACIÓN: no se pudo encontrar ningún archivo para los patrones dados.
INFORMACIÓN: no se pudo encontrar ningún archivo para los patrones dados.
[2023-09-27 15:40:17] INFO WEBrick 1.4.2
[2023-09-27 15:40:17] INFO ruby 2.6.6 (2020-03-31) [x64-mingw32]
[2023-09-27 15:40:17] INFO WEBrick::HTTPServer#start: pid=32676 port=3000

```

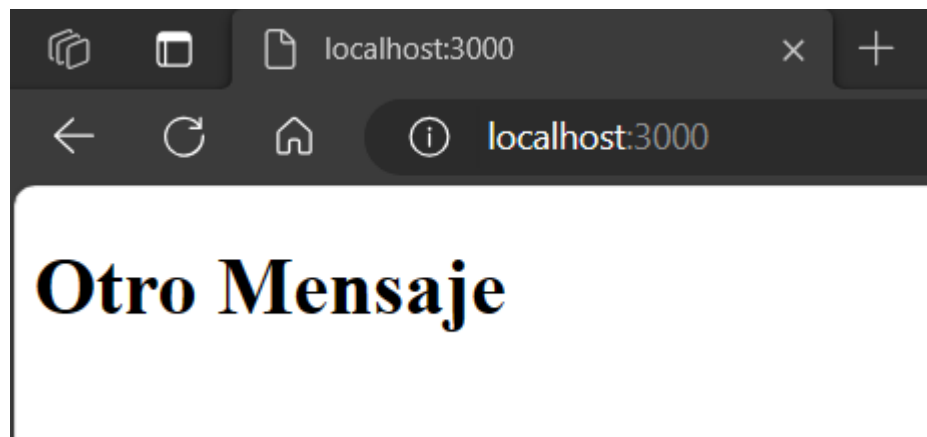
Ahora editamos directamente en nuestro archivo `app.rb` y reemplazamos el `'Goodbye Word'` por `'Otro Mensaje'`

```

app.rb
1 require 'sinatra'
2 class MyApp < Sinatra::Base
3   get '/' do
4     "<!DOCTYPE html><html><head></head><body><h1>Otro Mensaje</h1></body></html>"
5   end
6 end

```

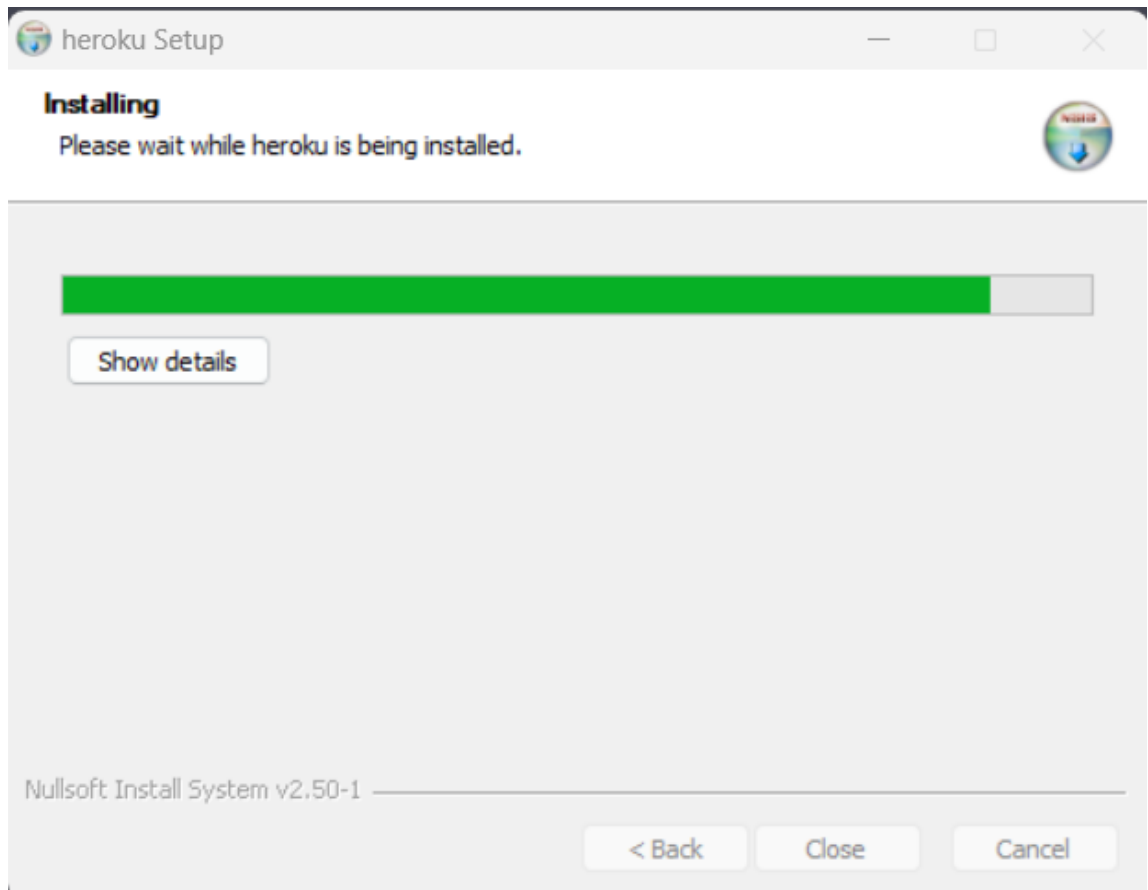
Y por ultimo refrescamos directamente la pagina para verificar los cambios



In []:

In []:

In []:



Verificamos la version de heroku

```
$ heroku --version
heroku/8.4.3 win32-x64 node-v16.19.0
```

Ahora accedemos a nuestra cuenta de heroku a traves de la terminal con el comando
heroku login -i

```
user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS (master)
$ heroku login
heroku: Press any key to open up the browser to login or q to exit:
s
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/4f7
f36e1-2e59-4eae-977f-ed60a0dea43d?requestor=SFMyNTY.g2gDbQAAAA0xODE
uMTC2LjQ5LjY1bgYA4FC12IoBYgABUYA.7KFWRBszXzXNsJD5JuDtEerNlLt0xFYYt7
t_Bx5HVrg
Logging in... done
Logged in as pldelacruzv@gmail.com
```

Verificamos que estemos conectados con el comando heroku whoami

```
$ heroku whoami
pldelacruzv@gmail.com
```

Vamos a trabajar con una version anterior de heroku, la razon es porque el proyecto se encuentra hecho con la version ruby-2.6.6 y este no es compatible con la version heroku-22, por lo que vamos a trabajar con heroku-20 para satisfacer dicha compatibilidad

```
$ heroku stack:set heroku-20
Setting stack to heroku-20... done
```

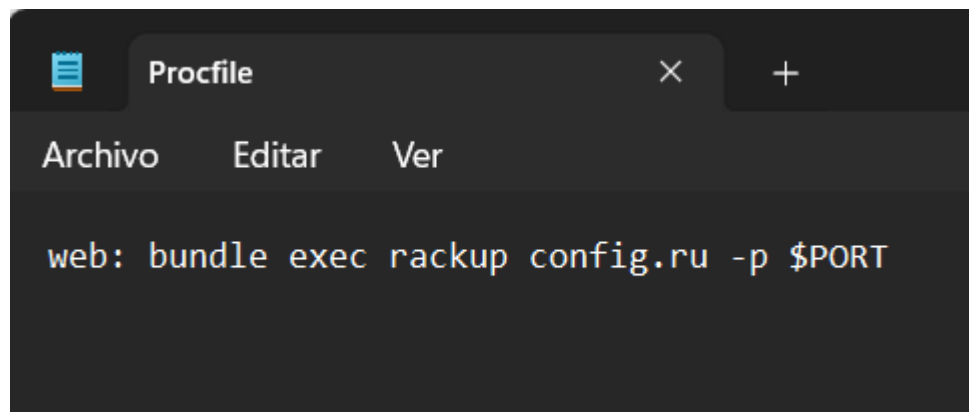
Ahora creamos un nuevo proyecto de heroku-20

```
$ heroku create --stack heroku-20
Creating app... done, aqueous-citadel-71756, stack is heroku-20
https://aqueous-citadel-71756-85546a2f9489.herokuapp.com/ | https://
git.heroku.com/aqueous-citadel-71756.git
```

Verificamos la creacion del proyecto

```
$ heroku apps
=== pldelacruzv@gmail.com Apps
aqueous-citadel-71756
```

Creamos el archivo Procfile con el siguiente contenido:

A screenshot of a code editor window titled 'Procfile'. The window has a menu bar with 'Archivo', 'Editar', and 'Ver'. The main content area shows the text 'web: bundle exec rackup config.ru -p \$PORT'.

Ahora vamos a implementar nuestro repositorio local en Heroku, para ello seguimos los comandos que se encuentran en Deploy de nuestra aplicacion en heroku

```
$ heroku git:remote -a aqueous-citadel-71756
set git remote heroku to https://git.heroku.com/aqueous-citadel-717
56.git
```

```
$ git add .
```

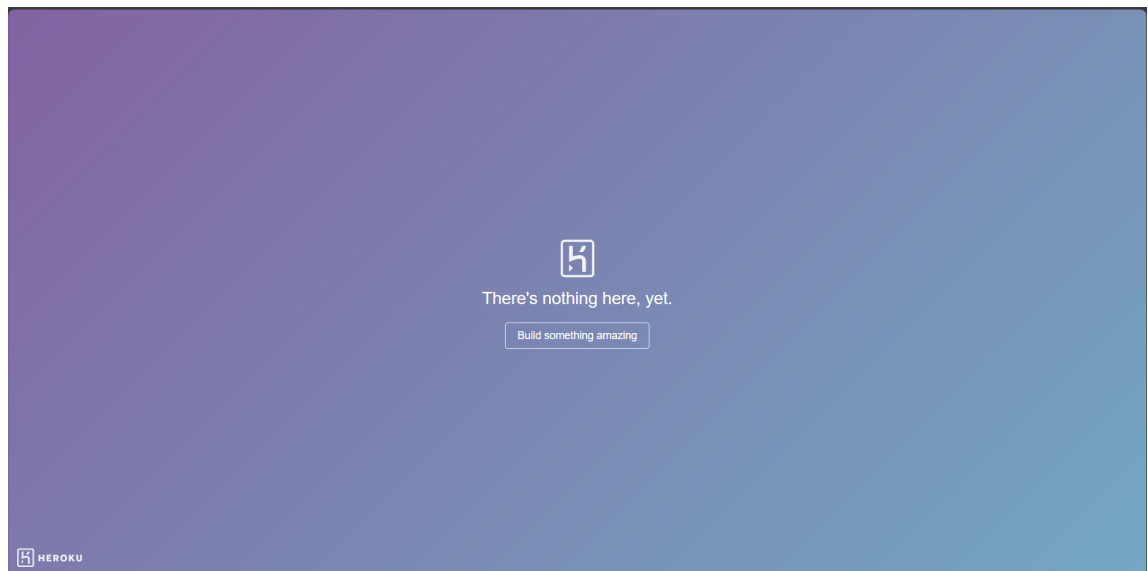


```
$ git commit -am "make it better"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
user@LAPTOP-JPL00230 MINGW64 ~/Desktop/PrimeraPCDS (main)
$ git push heroku main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (22/22), 690.21 KiB | 13.80 MiB/s, done.
Total 22 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Updated 8 paths from accb438
remote: Compressing source files... done.
remote: Building source:
remote:
remote: ----> Building on the Heroku-20 stack
remote: ----> Determining which buildpack to use for this app
remote: ----> Ruby app detected
remote: ----> Installing bundler 1.17.3
remote: ----> Removing BUNDLED WITH version in the Gemfile.lock
remote: ----> Compiling Ruby/Rack
remote: ----> Using Ruby version: ruby-2.6.6
remote: ##### WARNING:
```

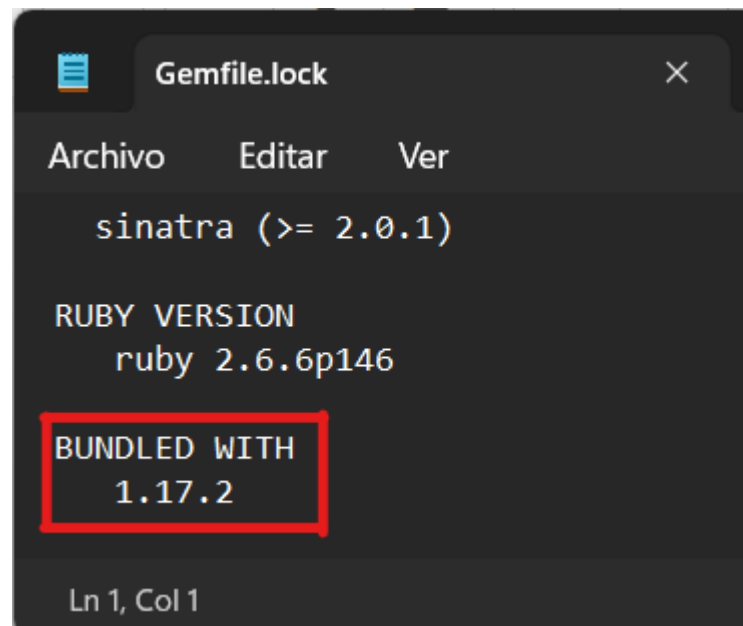
Podemos ver nuestra app visitando <https://aqueous-citadel-71756.herokuapp.com/>



Clonamos el repositorio del Wordguesser

```
$ git clone https://github.com/saasbook/hw-sinatra-saas-wordguesser
Cloning into 'hw-sinatra-saas-wordguesser'...
remote: Enumerating objects: 646, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 646 (delta 1), reused 26 (delta 0), pack-reused 608
Receiving objects: 100% (646/646), 266.64 KiB | 817.00 KiB/s, done.
Resolving deltas: 100% (350/350), done.
```

Instalamos las dependencias necesarias para el proyecto usando la version 1.17.3 del bundle (ya que así se encuentra especificado en el Gemfile.lock)



```
Gemfile.lock
Archivo  Editar  Ver

sinatra (>= 2.0.1)

RUBY VERSION
  ruby 2.6.6p146

BUNDLED WITH
  1.17.2

Ln 1, Col 1
```

```
$ bundle _1.17.3_ install
Using ZenTest 4.11.2
Using public_suffix 5.0.3
Using addressable 2.8.5
Using builder 3.2.3
Using bundler 1.17.3
Using columnize 0.9.0
Using byebug 5.0.0
Using mini_mime 1.1.5
Using racc 1.7.1
Using nokogiri 1.13.10 (x64-mingw32)
Using rack 2.2.8
Using rack-test 0.6.3
Using xpath 3.2.0
Using capybara 3.1.0
Using hitimes 2.0.0
Using timers 4.0.4
Using celluloid 0.16.0
```

Ejecutamos el comando `bundle exec autotest`

```
user@LAPTOP-JPL0023Q MINGW64 ~/Desktop/PrimeraPCDS/hw-sinatra-saas-
wordguesser (master)
$ bundle exec autotest
WARNING: You must use ANSICON 1.31 or later (http://adoxa.3eeweb.com/ansicon/) to use colour on windows. Called from C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/spec_helper.rb:40:in `block in <top (required)>'.
(Not running features. To run features in autotest, set AUTOFEATURE=true.)
loading autotest/rspec
"C:\Ruby26-x64\bin\ruby" -rrubygems -s "C:/Ruby26-x64/lib/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/wordguesser_game_spec.rb"
Run options: exclude {:pending=>true}
All examples were filtered out
Finished in 0.00042 seconds (files took 2.34 seconds to load)
0 examples, 0 failures
```

El mensaje `All examples were filtered out` se refiere a que todas las pruebas fueron filtradas y no se ejecutaron. Esto se debe a que en el archivo `wordguesser_game_spec.rb` posee un filtro en RSpec que excluye ciertas pruebas. La palabra clave `:pending => true` en el archivo `wordguesser_game_spec.rb` hace que esa prueba no se pueda ejecutar. Por ejemplo si borramos `:pending => true` de la primera prueba notamos que ahora si se ejecuta automáticamente en el Autotest

```

1  require 'spec_helper'
2  require 'wordguesser_game'
3
4  describe WordGuesserGame do
5    # helper function: make several guesses
6    def guess_several_letters(game, letters)
7      letters.chars do |letter|
8        game.guess(letter)
9      end
10   end
11
12   describe 'new', :pending => true do
13     it "takes a parameter and returns a WordGuesserGame object" do
14       @game = WordGuesserGame.new('glorp')
15       expect(@game).to be_an_instance_of(WordGuesserGame)
16       expect(@game.word).to eq('glorp')
17       expect(@game.guesses).to eq('')
18       expect(@game.wrong_guesses).to eq('')
19     end
20   end
21

```

Cuando lo borramos, automaticamente la salida del Autotest es la siguiente:

```

Run options: exclude {:pending=>true}

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object (FAILED
- 1)

Failures:

  1) wordGuesserGame new takes a parameter and returns a wordGuesse
rGame object
     Failure/Error: expect(@game.word).to eq('glorp')
     NoMethodError:
       undefined method `word' for #<WordGuesserGame:0x000000000748
b230 @word="glorp">
     # ./spec/wordguesser_game_spec.rb:16:in `block (3 levels) in <
top (required)>'

Finished in 0.04653 seconds (files took 0.6954 seconds to load)
1 example, 1 failure

Failed examples:

rspec ./spec/wordguesser_game_spec.rb:13 # WordGuesserGame new take
s a parameter and returns a WordGuesserGame object

```

Esto es útil cuando queremos ejecutar las pruebas uno a uno. Seguidamente vamos a mostrar nuestras soluciones para que pasen las pruebas correspondientes

Prueba 1

```
In [ ]: describe 'new' do
  it "takes a parameter and returns a WordGuesserGame object" do
    @game = WordGuesserGame.new('glorp')
    expect(@game).to be_an_instance_of(WordGuesserGame)
    expect(@game.word).to eq('glorp')
    expect(@game.guesses).to eq('')
    expect(@game.wrong_guesses).to eq('')
  end
end
```

En esta prueba esta verificando 4 cosas de una instancia de la clase WordGuesserGame:

- se verifica que @game sea realmente una instancia de WordGuesserGame
- se verifica que el atributo word de @game sea igual a glorp
- se verifica que el atributo guesses de @game sea una cadena vacia ''
- se verifica que el atributo wrong_guesses de @game sea una cadena vacia ''

Nuestra solucion seria:

```
In [ ]: class WordGuesserGame

  # add the necessary class methods, attributes, etc. here
  # to make the tests in spec/wordguesser_game_spec.rb pass.

  # Get a word from remote "random word" service
  attr_accessor :word, :guesses, :wrong_guesses

  def initialize(word)
    @word = word
    @guesses = ''
    @wrong_guesses = ''
  end
end
```

Guardamos y verificamos el autotest

```
WARNING: You must use ANSICON 1.31 or later (http://adoxa.3eeweb.com/ansicon/) to use colour on windows. Called from C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/spec_helper.rb:40:in `block in <top (required)>'.
"C:/Ruby26-x64/bin/ruby" -rrubygems -S "C:/Ruby26-x64/lib/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/wordguesser_game_spec.rb"
Run options: exclude {:pending=>true}

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object

Finished in 0.03426 seconds (files took 0.73492 seconds to load)
1 example, 0 failures
```

Prueba 2 y 3

```
In [ ]: ...
describe 'guessing' do
  context 'correctly' do
    before :each do
      @game = WordGuesserGame.new('garply')
      @valid = @game.guess('a')
    end
    it 'changes correct guess list' do
      expect(@game.guesses).to eq('a')
      expect(@game.wrong_guesses).to eq('')
    end
    it 'returns true' do
      expect(@valid).not_to be false
    end
  end
end
...
```

En esta prueba se estan verificando 2 cosas:

- se crea una instancia con la palabra `garply` y se guarda en la variable `@valid`
- despues de una adivinanza correcta, la letra adivinada se agregue al atributo `@guesses`
- y como no hay letra incorrecta el atributo `@wrong_guesses` es una cadena vacia ''

```
In [ ]: def guess(letra)
  if @word.include?(letra)
    @guesses << letra
    true
  else
    @wrong_guesses << letra
    false
  end
end
```

Al recibir `letra` verificamos si `letra` se encuentra incluido en `@word`. Si es `true` entonces agregamos `letra` al atributo `@guesses` a menos de que ya se encuentre incluido. Sino se agregara a las palabras incorrectas a menos que ya se encuentre incluido. Finalmente retornara un booleano dependiendo si la letra fue adivinada o no.

```

WARNING: You must use ANSICON 1.31 or later (http://adoxa.3eeweb.com/ansicon/) to use colour on windows. Called from C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/spec_helper.rb:40:in `lock in <top (required)>'.
"C:/Ruby26-x64/bin/ruby" -rrubygems -S "C:/Ruby26-x64/lib/ruby/gems/.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/wordguesser_game_spec.rb"
Run options: exclude {:pending=>true}

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object
  guessing
    correctly
      changes correct guess list
      returns true

Finished in 0.03429 seconds (files took 0.75289 seconds to load)
3 examples, 0 failures

```

Prueba 4 y 5

```

In [ ]: context 'incorrectly' do
  before :each do
    @game = WordGuesserGame.new('garply')
    @valid = @game.guess('z')
  end
  it 'changes wrong guess list' do
    expect(@game.guesses).to eq('')
    expect(@game.wrong_guesses).to eq('z')
  end
  it 'returns true' do
    expect(@valid).not_to be false
  end
end

```

Aquí en el contexto de una adivinanza incorrecta se debería cumplir lo opuesto, es decir que para una letra que no se encuentra incluida en la palabra no se le aprega nada al atributo `guesses` pero sí al atributo `wrong_guesses` y la validación debe de ser `true`. Nuestro código anterior cumple estas validaciones excepto la última prueba. Esto se soluciona borrando las líneas de `true` y `false`

```

In [ ]: def guess(letra)
  if @word.include?(letra)
    @guesses << letra
  else
    @wrong_guesses << letra
  end
end

```

```

WARNING: You must use ANSICON 1.31 or later (http://adoxa.3eeweb.com
ansicon/) to use colour on windows. Called from C:/Users/user/Desktop
/PrimeraPCDS/hw-sinatra-saas-wordguesser/spec/spec_helper.rb:40:in `
lock in <top (required)>'.
"C:/Ruby26-x64/bin/ruby" -rrubygems -s "C:/Ruby26-x64/lib/ruby/gems/
.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "C:/Users/user/Desktop/P
rimeraPCDS/hw-sinatra-saas-wordguesser/spec/wordguesser_game_spec.rb"
Run options: exclude {:pending=>true}

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object
  guessing
    correctly
      changes correct guess list
      returns true
    incorrectly
      changes wrong guess list
      returns true

Finished in 0.03858 seconds (files took 0.82788 seconds to load)
5 examples, 0 failures

```

Pruebas 6, 7, 8 y 9

```

In [ ]: context 'same letter repeatedly' do
  before :each do
    @game = WordGuesserGame.new('garply')
    guess_several_letters(@game, 'aq')
  end
  it 'does not change correct guess list' do
    @game.guess('a')
    expect(@game.guesses).to eq('a')
  end
  it 'does not change wrong guess list' do
    @game.guess('q')
    expect(@game.wrong_guesses).to eq('q')
  end
  it 'returns false' do
    expect(@game.guess('a')).to be false
    expect(@game.guess('q')).to be false
  end
  it 'is case insensitive' do
    expect(@game.guess('A')).to be false
    expect(@game.guess('Q')).to be false
    expect(@game.guesses).not_to include('A')
    expect(@game.wrong_guesses).not_to include('Q')
  end
end

```

En esta prueba se estan verificando 4 cosas:

- Si se adivina la letra correcta mas de una vez, el juego no debe de añadir esta letra a guesses otra vez
- Si se adivina la letra incorrecta mas de una vez, el juego no debe de añadir esta letra a wrong_guesses
- Si se intenta adivinar una letra repetidamente, el juego debe de retornar false

- El juego no debe de ser sensible a mayusculas y minusculas

```
In [ ]: def guess(letra)
        letra.downcase!

        return false if @guesses.include?(letra) || @wrong_guesses.include?(letra)

        if @word.include?(letra)
            @guesses << letra
        else
            @wrong_guesses << letra
        end
        true
    end
```

Con este codigo estamos asegurando con `letra.downcase!` que el juego no sea sensible a minusculas ni mayusculas. Si la letra ya ha sido adivinada correcta o incorrectamente, si es asi retornamos `false` , sino se añade a la lista correspondiente y se retorna `true`

```
WordGuesserGame
new
  takes a parameter and returns a WordGuesserGame object
guessing
  correctly
    changes correct guess list
    returns true
  incorrectly
    changes wrong guess list
    returns true
  same letter repeatedly
    does not change correct guess list
    does not change wrong guess list
    returns false
    is case insensitive

Finished in 0.03345 seconds (files took 0.72643 seconds to load)
9 examples, 0 failures
```

Pruebas 10, 11 y 12

```
In [ ]: context 'invalid' do
  before :each do
    @game = WordGuesserGame.new('foobar')
  end
  it 'throws an error when empty' do
    expect { @game.guess('') }.to raise_error(ArgumentError)
  end
  it 'throws an error when not a letter' do
    expect { @game.guess('%') }.to raise_error(ArgumentError)
  end
  it 'throws an error when nil' do
    expect { @game.guess(nil) }.to raise_error(ArgumentError)
  end
end
```

Con estas pruebas se quieren manejar algunas excepciones que pueden ocurrir:

- si la entrada es una cadena vacia
- si es una entrada que no sea una letra
- si la entrada es nula

```
In [ ]: def guess(letter)
  if letter.nil? || letter.empty? || !letter.match?(/[A-Za-z]/)
    raise ArgumentError.new("Invalid guess.")
  end

  letter.downcase!

  return false if @guesses.include?(letter) || @wrong_guesses.include?(letter)

  if @word.include?(letter)
    @guesses << letter unless @guesses.include?(letter)
  else
    @wrong_guesses << letter unless @wrong_guesses.include?(letter)
  end
  true
end
```

Para cumplir estas pruebas agregamos una condicional if con las excepciones requeridas, si es nulo, si es una cadena vacia o si no es una letra entonces lanzara `ArgumentError.new("Invalid guess.")`

```

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object
  guessing
    correctly
      changes correct guess list
      returns true
    incorrectly
      changes wrong guess list
      returns true
  same letter repeatedly
    does not change correct guess list
    does not change wrong guess list
    returns false
  is case insensitive
  invalid
    throws an error when empty
    throws an error when not a letter
    throws an error when nil

```

Finished in 0.04072 seconds (files took 0.73736 seconds to load)
 12 examples, 0 failures

Pruebas 13, 14 y 15

```

In [ ]: describe 'displayed word with guesses' do
  before :each do
    @game = WordGuesserGame.new('banana')
  end
  # for a given set of guesses, what should the word look like?
  @test_cases = {
    'bn' => 'b-n-n-',
    'def' => '-----',
    'ban' => 'banana'
  }
  @test_cases.each_pair do |guesses, displayed|
    it "should be '#{displayed}' when guesses are '#{guesses}'" do
      guess_several_letters(@game, guesses)
      expect(@game.word_with_guesses).to eq(displayed)
    end
  end
end
end

```

En este caso realizamos tres pruebas:

- si la entrada es 'bn' se debe desplegar 'b-n-n-'
- si la entrada es 'def' se debe desplegar '-----'
- si la entrada es 'ban' se debe de desplegar 'banana'

```
In [ ]: def word_with_guesses
  resultado = ''

  @word.chars.each do |letra|
    if @guesses.include?(letra)
      resultado += letra
    else
      resultado += '-'
    end
  end

  resultado
end
```

El metodo `word_with_guesses` va a inicializar la cadena `resultado` como una cadena vacia, luego estamos dividiendo `@word` en caracteres y luego vamos a iterar cada letra usando `.each`. Si el caracter se encuentra incluido en `@guesses` entonces se le agrega al resultado, sino se le agrega `-` y finalmente se retorna `resultado`

```
new
  takes a parameter and returns a WordGuesserGame object
guessing
  correctly
    changes correct guess list
    returns true
  incorrectly
    changes wrong guess list
    returns true
  same letter repeatedly
    does not change correct guess list
    does not change wrong guess list
    returns false
  is case insensitive
invalid
  throws an error when empty
  throws an error when not a letter
  throws an error when nil
displayed word with guesses
  should be 'b-n-n-' when guesses are 'bn'
  should be '-----' when guesses are 'def'
  should be 'banana' when guesses are 'ban'
```

```
Finished in 0.02592 seconds (files took 0.72968 seconds to load)
15 examples, 0 failures
```

Prueba 16, 17 y 18

```
In [ ]: describe 'game status' do
  before :each do
    @game = WordGuesserGame.new('dog')
  end
  it 'should be win when all letters guessed' do
    guess_several_letters(@game, 'ogd')
    expect(@game.check_win_or_lose).to eq(:win)
  end
  it 'should be lose after 7 incorrect guesses' do
    guess_several_letters(@game, 'tuvwxyz')
    expect(@game.check_win_or_lose).to eq(:lose)
  end
  it 'should continue play if neither win nor lose' do
    guess_several_letters(@game, 'do')
    expect(@game.check_win_or_lose).to eq(:play)
  end
end
```

Aqui se estan haciendo 3 verificaciones:

- Se gano el juego si todos los caracteres han sido adivinados
- Se perdio el juego ya que el numero de intentos es mayor a 7
- El juego continua (todavia no se gana ni se pierde)

```
In [ ]: def check_win_or_lose
  if @word.chars.all? { |c| @guesses.include?(c) }
    :win
  elsif @wrong_guesses.length >= 7
    :lose
  else
    :play
  end
end
```

Para resolver estas pruebas, creamos el metodo `check_win_or_lose` . Luego se aplica la condicional if con la condicion si todos los caracteres estan incluidos en `@guesses` , si es verdadero entonces se gana el juego, si es falso se verifica si tenemos mas de 7 intentos incorrectos, si lo hay se pierde sino sigue el juego.

```

incorrectly
  changes wrong guess list
  returns true
same letter repeatedly
  does not change correct guess list
  does not change wrong guess list
  returns false
  is case insensitive
invalid
  throws an error when empty
  throws an error when not a letter
  throws an error when nil
displayed word with guesses
  should be 'b-n-n-' when guesses are 'bn'
  should be '-----' when guesses are 'def'
  should be 'banana' when guesses are 'ban'
game status
  should be win when all letters guessed
  should be lose after 7 incorrect guesses
  should continue play if neither win nor lose

Finished in 0.03681 seconds (files took 0.69416 seconds)
18 examples, 0 failures

```

Ahora si revisamos el metodo `get_random_word` este retorna palabras aleatorias a traves del servidor `http://randomword.saasbook.info/RandomWord` , podemos obtener estas palabras a traves del comando `curl` de la siguiente manera:

```

user@LAPTOP-JPL0023Q MINGW64 /
$ curl --data '' http://randomword.saasbook.info/RandomWord
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %       Dload  Upload  Total   Spent    Left   Speed
100    10    100    10    0    0      47    0  --:--:-- --:--:-- --:--:--    47 possessive

user@LAPTOP-JPL0023Q MINGW64 /
$ curl --data '' http://randomword.saasbook.info/RandomWord
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %       Dload  Upload  Total   Spent    Left   Speed
100     6    100     6    0    0      25    0  --:--:-- --:--:-- --:--:--    25 flavor

user@LAPTOP-JPL0023Q MINGW64 /
$ curl --data '' http://randomword.saasbook.info/RandomWord
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %       Dload  Upload  Total   Spent    Left   Speed
100     7    100     7    0    0      31    0  --:--:-- --:--:-- --:--:--    31 married

user@LAPTOP-JPL0023Q MINGW64 /
$ curl --data '' http://randomword.saasbook.info/RandomWord
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %       Dload  Upload  Total   Spent    Left   Speed
100     6    100     6    0    0      18    0  --:--:~ --:~:~ --:~:~    18 lonely

```

Se puede agregar el `-s` para obtener solo la palabra

```
user@LAPTOP-JPL0023Q MINGW64 /  
$ curl -s --data '' http://randomword.saasbook.info/RandomWord  
machine  
user@LAPTOP-JPL0023Q MINGW64 /  
$ curl -s --data '' http://randomword.saasbook.info/RandomWord  
scientific  
user@LAPTOP-JPL0023Q MINGW64 /  
$ curl -s --data '' http://randomword.saasbook.info/RandomWord  
fortunate  
user@LAPTOP-JPL0023Q MINGW64 /  
$ curl -s --data '' http://randomword.saasbook.info/RandomWord  
spotless  
user@LAPTOP-JPL0023Q MINGW64 /  
$ curl -s --data '' http://randomword.saasbook.info/RandomWord  
loving  
user@LAPTOP-JPL0023Q MINGW64 /  
$ curl -s --data '' http://randomword.saasbook.info/RandomWord  
courageous  
user@LAPTOP-JPL0023Q MINGW64 /
```

Parte 2

Este apartado se explico en la grabación ! No hay codigo para este apartado, solo es teoría base para lo que viene en la siguiente parte.

Pate 3

Entramos a la carpeta `hw-sinatra-saas-wordguesser` y ejecutamos el comando `bundle install`

```
$ bundle install
Using ZenTest 4.11.2
Using public_suffix 5.0.3
Using addressable 2.8.5
Using builder 3.2.3
Using bundler 1.17.3
Using columnize 0.9.0
Using byebug 5.0.0
Using mini_mime 1.1.5
Using racc 1.7.1
Using nokogiri 1.13.10 (x64-mingw32)
Using rack 2.2.8
Using rack-test 0.6.3
Using xpath 3.2.0
Using capybara 3.1.0
Using hitimes 2.0.0
Using timers 4.0.4
Using celluloid 0.16.0
Using safe_yaml 1.0.4
Using crack 0.4.3
Using multi_json 1.13.1
Using gherkin 2.12.2
Using cucumber-core 1.1.3
```

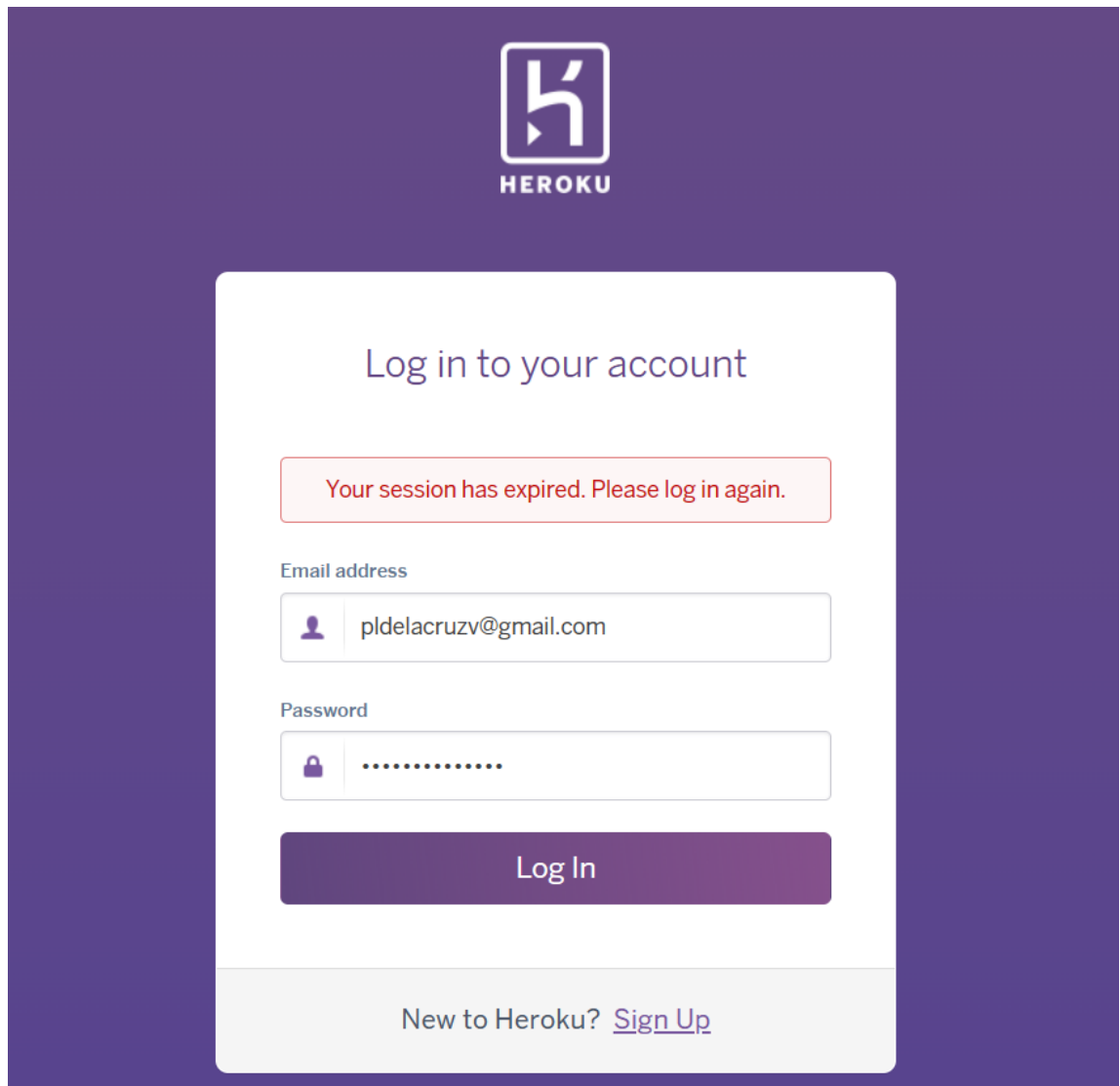
Agregamos todos los archivos al git

```
$ git add .
```

Hacemos el commit

```
$ git commit -m "¡Listo para Heroku!"
[master 278b6d0] ¡Listo para Heroku!
 4 files changed, 129 insertions(+), 42 deletions(-)
```

Realizamos el login en heroku



```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit: s
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/aec3290b-005c-4eea-9a48-fa59c8dbce6d?requestor=SFMyNTY.g2gDbQAAAA4xOTAUMjMzLjIwNi44MG4GAGAAHuqKAWIAAVGA.JaWIIBcS3I52ErXmXbzLZfZM0WINAVKWMhK1FFbnsVQ
Logging in... done
Logged in as pldelacruzv@gmail.com
```

Asi como hicimos anteriormente, vamos a crear un proyecto de heroku-20

```
$ heroku create --stack heroku-20
creating app... done, safe-beach-32833, stack is heroku-20
https://safe-beach-32833-ab16c02a534f.herokuapp.com/ | https://git.heroku.com/safe-beach-32833.git
```

El nombre de nuestro proyecto es safe-beach-32833

Para asegurarnos que nuestra aplicacion se conecto con heroku podemos escribir el comando `git remote -v`

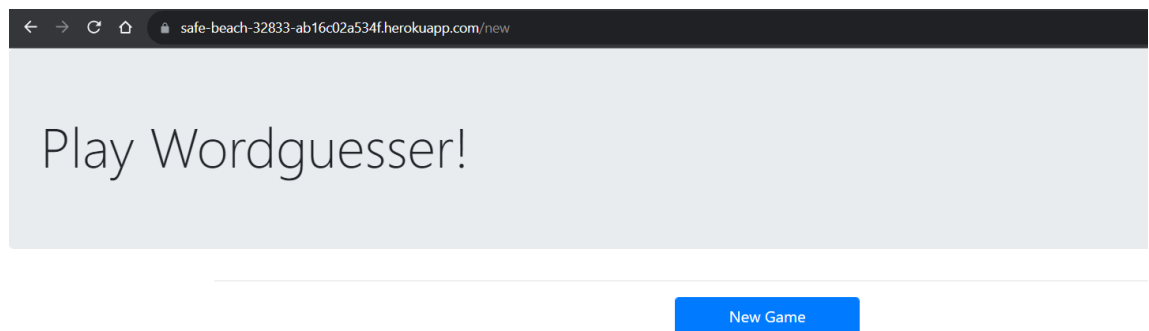
```
$ git remote -v
heroku https://git.heroku.com/safe-beach-32833.git (fetch)
heroku https://git.heroku.com/safe-beach-32833.git (push)
origin https://github.com/saasbook/hw-sinatra-saas-wordguesser (fetch)
origin https://github.com/saasbook/hw-sinatra-saas-wordguesser (push)
```

De esa manera cuando escribimos `git push heroku master` enviamos nuestro código a Heroku. Heroku recibe el código, detecta el tipo de aplicación e instala las dependencias necesarias y lanza la aplicación en uno o más dynos (contenedores que ejecutan aplicaciones en Heroku)

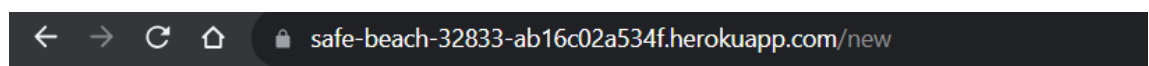
Subimos los cambios a Heroku con el comando `git push heroku master`

```
$ git push heroku master
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 12 threads
Compressing objects: 100% (44/44), done.
Writing objects: 100% (53/53), 36.16 KiB | 18.08 MiB/s, done.
Total 53 (delta 6), reused 36 (delta 0), pack-reused 0
remote: Updated 36 paths from e6185f6
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Ruby app detected
remote: -----> Installing bundler 1.17.3
remote: -----> Removing BUNDLED WITH version in the Gemfile.lock
remote: -----> Compiling Ruby/Rack
remote: -----> Using Ruby version: ruby-2.6.6
remote:
remote: ##### WARNING:
remote:
```

Por último nos dirigimos al URL que nos proporciona Heroku: <https://safe-beach-32833-ab16c02a534f.herokuapp.com/new> (<https://safe-beach-32833-ab16c02a534f.herokuapp.com/new>)



Nos saldrá la misma página que estábamos desarrollando de manera local, pero esta vez puede ser accedida por cualquier computador con internet, si le damos al botón 'New Game' nos saldrá:



Not Found

Una Solucion

Los archivos HTML se encuentran en la carpeta `views` , si en abrimos el archivo `new.erb` nos encontramos con el siguiente contenido:

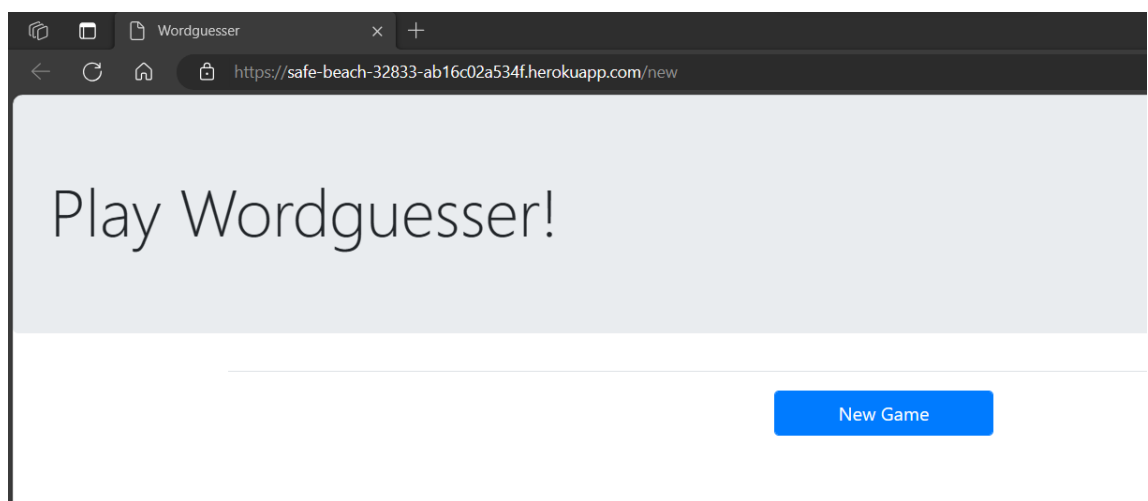
```
new.erb
Archivo  Editar  Ver

<!-- This form is incomplete--it needs a destination URL as well as a method: -->
<form method="post">
  <div class="form-row py-3 border-top">
    <input type="submit" value="New Game" class="col-md-2 offset-md-5 btn btn-primary form-control"/>
  </div>
</form>
```

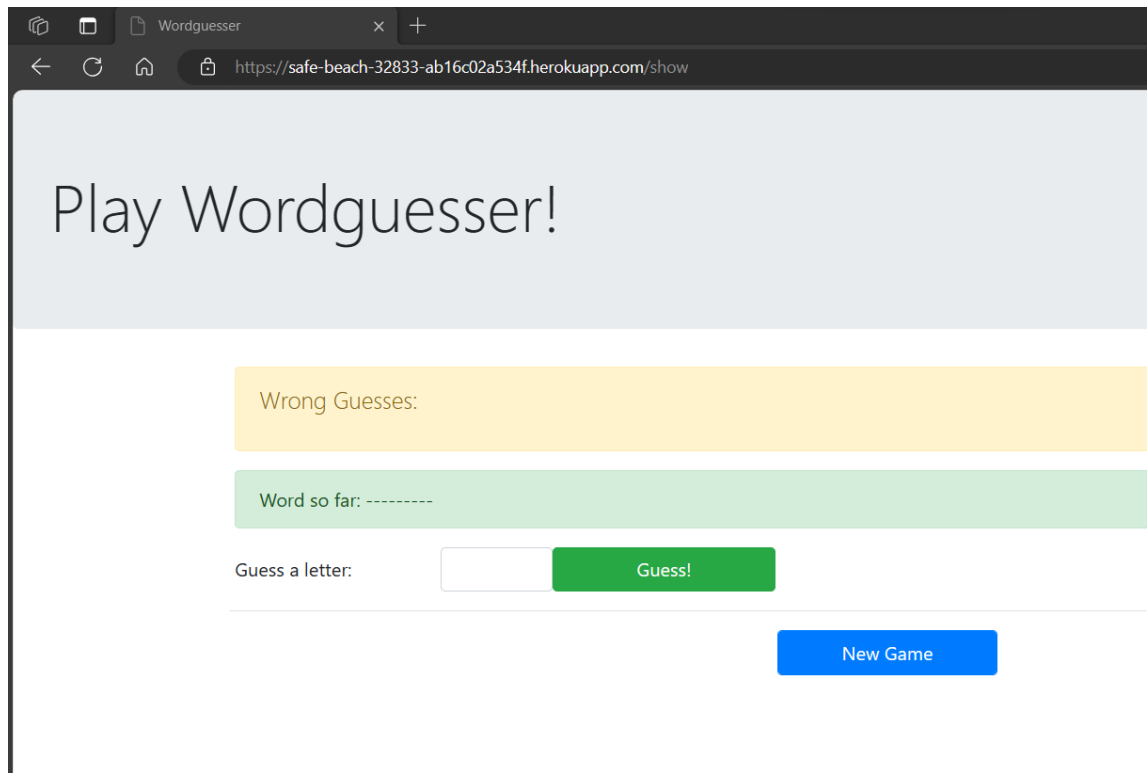
Esto es un problema, ya que falta el atributo `action` , que define la URL a la que se le enviara el formulario cuando se presione el boton. Nuestra intencion es enviar este formulario a la accion `/create` , etonces le agregamos ese atributo:

```
new.erb
Archivo  Editar  Ver

<!-- This form now has a destination URL (the /create route) and the POST method: -->
<form method="post" action="/create">
  <div class="form-row py-3 border-top">
    <input type="submit" value="New Game" class="col-md-2 offset-md-5 btn btn-primary form-control"/>
  </div>
</form>
```



Ahora presionamos `New Game`



Ahora estamos en el juego pero no podemos jugar, cuando tratamos de adivinar la palabra, simplemente no aparece nada, eso es porque falta completar el código del archivo `app.rb` (Donde vamos a agregar la lógica de nuestro juego). Pero ello lo vamos a hacer utilizando Cucumber de la siguiente forma:

Pruebas con Cucumber

Bien ahora si ejecutamos el comando `cucumber features/start_new_game.feature` nos saldrá la siguiente salida:

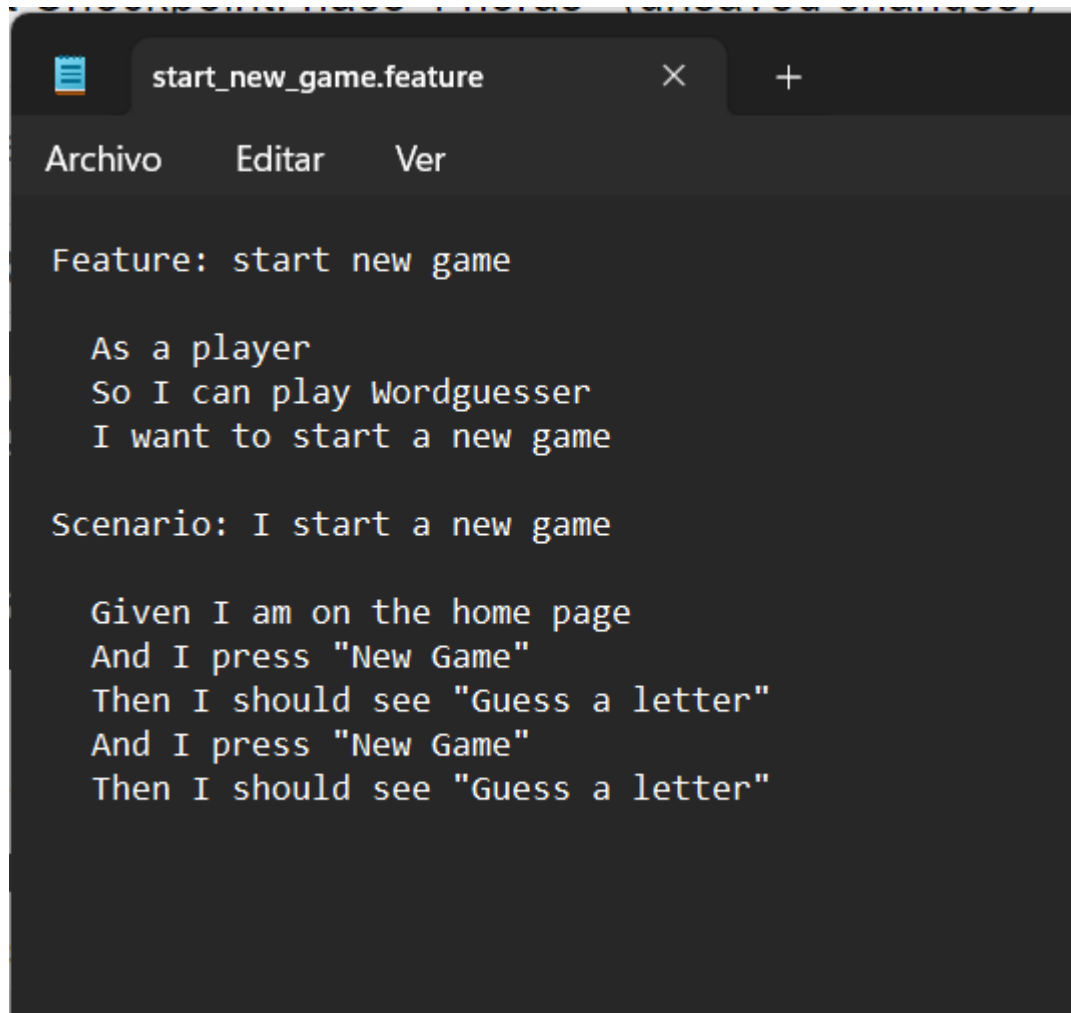
```
Feature: start new game
  As a player
  So I can play wordguesser
  I want to start a new game

  Scenario: I start a new game # features/start_new_game.feature:7
    Given I am on the home page # features/step_definitions/game_steps.rb:61
    And I press "New Game" # features/step_definitions/game_steps.rb:74
    Then I should see "Guess a letter" # features/step_definitions/game_steps.rb:70
    And I press "New Game" # features/step_definitions/game_steps.rb:74
    Then I should see "Guess a letter" # features/step_definitions/game_steps.rb:70

1 scenario (1 passed)
5 steps (5 passed)
0m0.069s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 43 / 61 LOC (70.49%) covered.
```

¿Que significa esto?

Cuando se ejecuta el comando `cucumber features/start_new_game.feature` se está usando la herramienta cucumber para ejecutar pruebas de comportamiento basadas en escenarios que se han definido en el archivo `start_new_game.feature`



Como notamos, este archivo se encuentra escrito en lenguaje natural pero estructurado. Cucumber empieza a ejecutar cada escenario (es un conjunto de pasos que describen un comportamiento específico que se espera de nuestra aplicación) del archivo

El escenario es `I start a new game` (es la característica que se quiere probar) y tiene 5 pasos, si lo comparamos con nuestro archivo `features/step_definitions/game_steps.rb` podemos inferir que se encuentran conectados de la siguiente manera:

Given I am on the home

Esto se relaciona con:

```
In [ ]: ...
Given /^(?:|I )am on (.+)$/ do |page_name|
  visit path_to(page_name)
end
...
```

Este paso te lleva a la página de inicio

- `Given` es una palabra clave en Cucumber para describir un estado inicial o precondición.

- La expresion regular `/^(?:|I)am on (.+)$/` el primer grupo es un grupo no capturador, indica que la frase se puede expresar como `I am o am`. El segundo grupo es un grupo capturador que captura uno o mas caracteres y los pasa como argumentos al bloque
- `visit` es un metodo de Capybara que permite navegar a una URL
- y por ultimo se usa el metodo `path_to()` que se encuentra definido en `features/support/paths.rb`. Este metodo se encarga de determinar que ruta o URL corresponde al nombre de la pagina proporcionado.

And I press "New Game"

Esto se relaciona con:

```
In [ ]: ...
When /^(?:|I )press "([^"]*)"(?: within "([^"]*)"?)?$/ do |button, selector|
  filtered_page(selector).click_button(button)
end
...
```

Este paso simula un click en el boton `New Game`

Then I should see "Guess a letter"

Esto se relaciona con:

```
In [ ]: ...
Then /^(?:|I )should see "([^"]*)"(?: within "([^"]*)"?)?$/ do |text, selector|
  expect(filtered_page(selector)).to have_text(text)
end
...
```

Bien regresando a la salida del comando `cucumber`

`features/start_new_game.feature`, al ultimo nos indica lo siguiente:

```
Feature: start new game

  As a player
  So I can play wordguesser
  I want to start a new game

  Scenario: I start a new game # features/start_new_game.feature:7
    Given I am on the home page # features/step_definitions/game_steps.rb:61
    And I press "New Game" # features/step_definitions/game_steps.rb:74
    Then I should see "Guess a letter" # features/step_definitions/game_steps.rb:70
    And I press "New Game" # features/step_definitions/game_steps.rb:74
    Then I should see "Guess a letter" # features/step_definitions/game_steps.rb:70

1 scenario (1 passed)
5 steps (5 passed)
0m0.069s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 43 / 61 LOC (70.49%) covered.
```

Esto muestra que el escenario `start_new_game.feature` cuyos 5 pasos pasaron las pruebas exitosamente. Lo podemos verificar visitando nuestro juego que ya se encuentra en internet gracias a heroku, pues al presionar `New Game` podemos empezar el juego, pero el boton sigue presente, si el jugador presiona otra vez el boton se reinicia el juego.

Dado que nuestra aplicacion paso con exito la primera prueba, vamos a la siguiente, entonces vamos a escribir el comando `cucumber features/guess.feature`

La salida es la siguiente:

```
Scenario: multiple correct and incorrect guesses # features/guess.feature:25
  Given I start a new game with word "foobar" # features/step_definitions/game_steps.rb:9
  When I make the following guesses: a,z,x,o # features/step_definitions/game_steps.rb:22
  Then the word should read "-oo-a-" # features/step_definitions/game_steps.rb:38

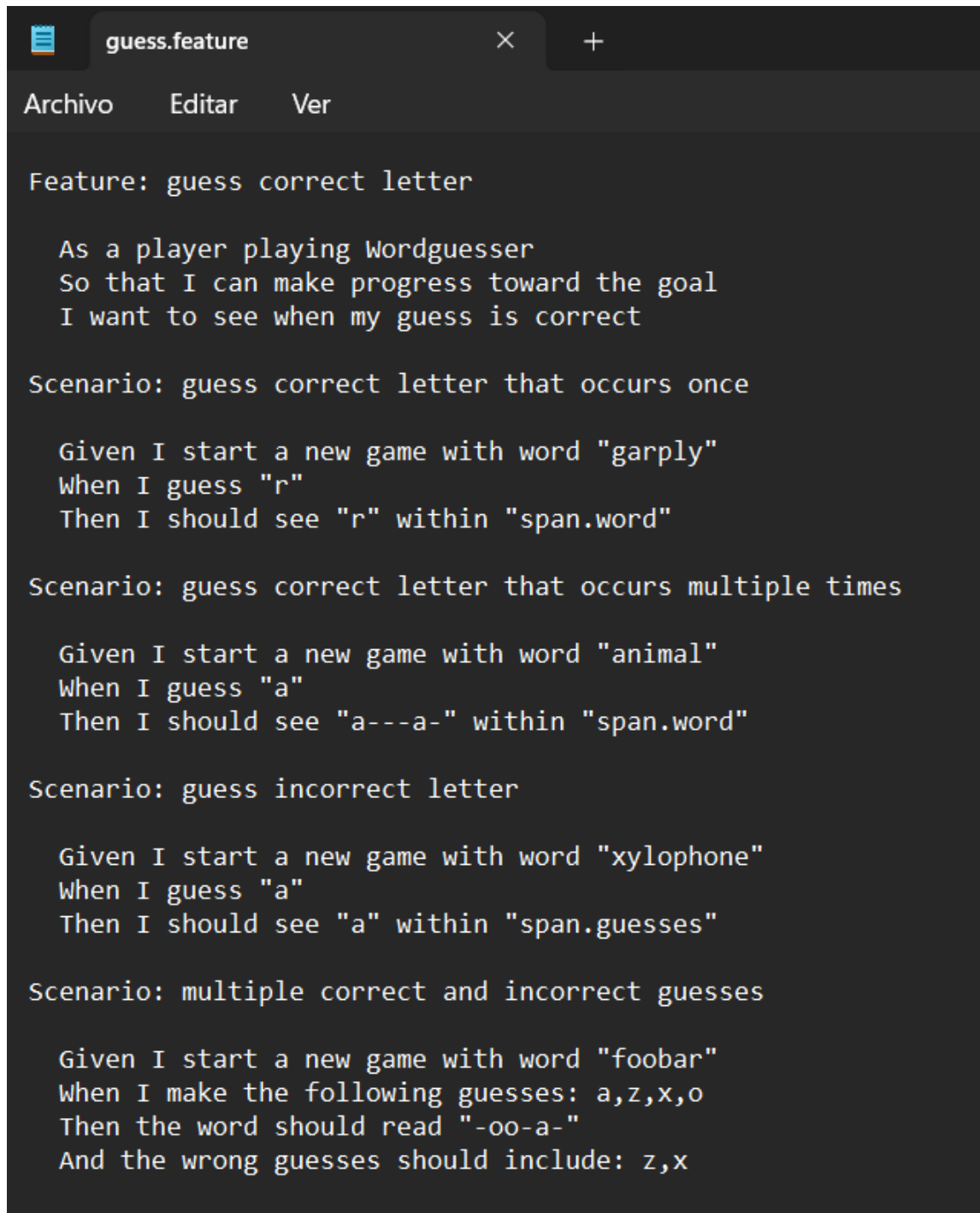
    expected: "-oo-a-"
    got: "-----"

    (compared using eql?)
    (RSpec::Expectations::ExpectationNotMetError)
    ./features/step_definitions/game_steps.rb:39:in `^the word should read "(.*)"$/'
    features/guess.feature:29:in `Then the word should read "-oo-a-"'
    And the wrong guesses should include: z,x # features/step_definitions/game_steps.rb:42

Failing Scenarios:
cucumber features/guess.feature:7 # Scenario: guess correct letter that occurs once
cucumber features/guess.feature:13 # Scenario: guess correct letter that occurs multiple times
cucumber features/guess.feature:19 # Scenario: guess incorrect letter
cucumber features/guess.feature:25 # Scenario: multiple correct and incorrect guesses

4 scenarios (4 failed)
13 steps (4 failed, 1 skipped, 8 passed)
0m0.155s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 44 / 61 LOC (72.13%) covered.
```

El resultado indica que tenemos 4 escenarios con 13 pasos, el cual 4 fallaron, 1 fue omitido y 8 pasaron con éxito. Para entender como podemos resolver, abrimos el archivo `guess.feature` para ver que es lo que se espera de nuestra aplicación



```
Feature: guess correct letter

  As a player playing Wordguesser
  So that I can make progress toward the goal
  I want to see when my guess is correct

Scenario: guess correct letter that occurs once

  Given I start a new game with word "garply"
  When I guess "r"
  Then I should see "r" within "span.word"

Scenario: guess correct letter that occurs multiple times

  Given I start a new game with word "animal"
  When I guess "a"
  Then I should see "a---a-" within "span.word"

Scenario: guess incorrect letter

  Given I start a new game with word "xylophone"
  When I guess "a"
  Then I should see "a" within "span.guesses"

Scenario: multiple correct and incorrect guesses

  Given I start a new game with word "foobar"
  When I make the following guesses: a,z,x,o
  Then the word should read "-oo-a-"
  And the wrong guesses should include: z,x
```

Para solucionar esto, vamos a agregar el siguiente contenido a nuestro bloque `post /guess` de la siguiente manera:


```
In [ ]: post '/guess' do
  letra = params[:guess].to_s[0]

  if letra.nil? || !letra.match(/[a-zA-Z]/)
    flash[:message] = "Adivinanza incorrecta :("
  elsif !@game.guess(letra)
    flash[:message] = "ya usaste esta letra :/"
  end

  redirect '/show'
end
```

1. Capturamos la letra ingresada por el jugador, y lo guardamos en letra

- params es un hash provisto por sinatra que contiene todos los parametros pasados a la aplicacion, en este caso estamos obteniendo el parametro guess que contiene la letra que el jugador esta tratando de adivinar
- to_s[0] esta convirtiendo la entrada del jugador en string y solo esta tomando la primera letra del string, ya que puede ser que la entrada del jugador sea una cadena.

2. Comprobar si la letra ingresada es valida

- la letra ingresada puede ser que sea una cadena vacia o un simbolo que no sea una letra, en esos casos le mandamos un mensaje al jugador Adivinanza incorrecta :(
- en el caso contrario verificamos si esa letra ya se encontraba adivinada, si es asi le enviamos el mensaje ya usaste esta letra :/ al jugador

3. Redireccionar al usuario

- Despues de procesar la letra adivinada independientemente si fue valida o invalida o ya adivinada previamente, redireccionamos al usuario (jugador) a la ruta /show

Verificamos:

```
Feature: guess correct letter

  As a player playing Wordguesser
  So that I can make progress toward the goal
  I want to see when my guess is correct

  Scenario: guess correct letter that occurs once # features/guess.feature:7
    Given I start a new game with word "garply" # features/step_definitions/game_steps.rb:9
    When I guess "r" # features/step_definitions/game_steps.rb:16
    Then I should see "r" within "span.word" # features/step_definitions/game_steps.rb:70

  Scenario: guess correct letter that occurs multiple times # features/guess.feature:13
    Given I start a new game with word "animal" # features/step_definitions/game_steps.rb:9
    When I guess "a" # features/step_definitions/game_steps.rb:16
    Then I should see "a--a-" within "span.word" # features/step_definitions/game_steps.rb:70

  Scenario: guess incorrect letter # features/guess.feature:19
    Given I start a new game with word "xylophone" # features/step_definitions/game_steps.rb:9
    When I guess "a" # features/step_definitions/game_steps.rb:16
    Then I should see "a" within "span.guesses" # features/step_definitions/game_steps.rb:70

  Scenario: multiple correct and incorrect guesses # features/guess.feature:25
    Given I start a new game with word "foobar" # features/step_definitions/game_steps.rb:9
    When I make the following guesses: a,z,x,o # features/step_definitions/game_steps.rb:22
    Then the word should read "-oo-a-" # features/step_definitions/game_steps.rb:38
    And the wrong guesses should include: z,x # features/step_definitions/game_steps.rb:42

4 scenarios (4 passed)
13 steps (13 passed)
0m0.236s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saas-wordguesser/coverage. 58 / 73 LOC (79.45%) covered.
```

Ahora vamos a la tercera prueba, vamos a escribir el comando `cucumber features/invalid_guess.feature`. La salida es la siguiente:

```
Scenario: guess a noncharacter guess # features/invalid_guess.feature:15
  Given I start a new game with word "snake" # features/step_definitions/game_steps.rb:9
  When I guess "@" # features/step_definitions/game_steps.rb:16
  Then the server should not respond with status 500 # features/step_definitions/game_steps.rb:30
  And the server should respond with status 200 # features/step_definitions/game_steps.rb:30
  And I should see "Invalid guess." # features/step_definitions/game_steps.rb:70
    expected to find text "Invalid guess." in "Play wordguesser!\n\r\n\r\n\r\n\r Adivinanza incor
recta :(\r\n\r \r\n\r\n\r\n\r Wrong Guesses:\r \r\n\r\n\r\n\r \r\n\r\n\r Word so far:\r ----\r\n\r \r\n\r\n\r\n\r
Guess a letter:" (RSpec::Expectations::ExpectationNotMetError)
    ./features/step_definitions/game_steps.rb:71:in `/\^(?:|I )should see "([\^"]*)"(:? within "(["
^\"]*)"?$/`
    features/invalid_guess.feature:20:in `And I should see "Invalid guess."'

Failing Scenarios:
cucumber features/invalid_guess.feature:8 # Scenario: guess an empty guess
cucumber features/invalid_guess.feature:15 # Scenario: guess a noncharacter guess

2 scenarios (2 failed)
10 steps (2 failed, 8 passed)
0m0.094s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 46 / 65 LOC (70.77%) covered.
```

La razón por estas fallas es de que no encuentran las etiquetas `Invalid guess`

Esto lo podemos solucionar cambiando el mensaje que le damos al usuario, de la siguiente manera:

```
In [ ]: post '/guess' do
  letra = params[:guess].to_s[0]

  if letra.nil? || !letra.match(/[a-zA-Z]/)
    flash[:message] = "Invalid guess."
  elsif !@game.guess(letra)
    flash[:message] = "You have already used that letter."
  end

  redirect '/show'
end
```

Verificamos:

```
Feature: guess invalid inputs
  As a player playing wordguesser
  So that I can try and break the app
  I want to see when my guess is invalid

Scenario: guess an empty guess # features/invalid_guess.feature:8
  Given I start a new game with word "snake" # features/step_definitions/game_steps.rb:9
  When I guess "" # features/step_definitions/game_steps.rb:16
  Then the server should not respond with status 500 # features/step_definitions/game_steps.rb:30
  And the server should respond with status 200 # features/step_definitions/game_steps.rb:30
  And I should see "Invalid guess." # features/step_definitions/game_steps.rb:70

Scenario: guess a noncharacter guess # features/invalid_guess.feature:15
  Given I start a new game with word "snake" # features/step_definitions/game_steps.rb:9
  When I guess "@" # features/step_definitions/game_steps.rb:16
  Then the server should not respond with status 500 # features/step_definitions/game_steps.rb:30
  And the server should respond with status 200 # features/step_definitions/game_steps.rb:30
  And I should see "Invalid guess." # features/step_definitions/game_steps.rb:70

2 scenarios (2 passed)
10 steps (10 passed)
0m0.085s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 46 / 65 LOC (70.77%) covered.
```

Ahora vamos a la tercera prueba, vamos a escribir el comando `cucumber features/repeated_guess.feature`. La salida es la siguiente:

```
    Given I start a new game with word "giraffe" # features/step_definitions/game_steps
.rb:9
    When I guess "z" # features/step_definitions/game_steps
.rb:16
    And I guess "z" again # features/step_definitions/game_steps
.rb:16
    Then the word should read "-----" # features/step_definitions/game_steps
.rb:38
    And I should see "You have already used that letter" # features/step_definitions/game_steps
.rb:70

Scenario: guessing an incorrect letter does not count towards guesses # features/repeated_guess.f
eature:23
    Given I start a new game with word "snake" # features/step_definitions
/game_steps.rb:9
    When I guess "z" 30 times in a row # features/step_definitions
/game_steps.rb:49
    Then I should be on the show page # features/step_definitions
/game_steps.rb:65
3 scenarios (3 passed)
13 steps (13 passed)
0m0.166s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 55 / 65 LOC (84.62%) covered.
```

Como no tenemos errores aquí pasamos a la siguiente prueba. Entonces escribimos el comando `cucumber features/cheating.feature`. La salida es:

```
Scenario: navigate to win page # features/cheating.feature:13
  Given I start a new game with word "snake" # features/step_definitions/game_steps.rb:9
  When I try to go to the URL "/win" # features/step_definitions/game_steps.rb:57
  Then I should be on the show page # features/step_definitions/game_steps.rb:65

    expected: "/show"
    got: "/win"

    (compared using eql?)
    (RSpec::Expectations::ExpectationNotMetError)
    ./features/step_definitions/game_steps.rb:67:in `^/(?|I )should be on (.)$/'
    features/cheating.feature:16:in `Then I should be on the show page'
    And the word should read "-----" # features/step_definitions/game_steps.rb:38

Failing Scenarios:
cucumber features/cheating.feature:7 # Scenario: navigate to lose page
cucumber features/cheating.feature:13 # Scenario: navigate to win page
2 scenarios (2 failed)
8 steps (2 failed, 2 skipped, 4 passed)
0m0.096s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 44 / 65 LOC (67.69%) covered.
```

La falla de estas pruebas indican que no se está manejando bien los intentos por parte del usuario al navegar directamente al resultado de la página de victoria o derrota, debemos asegurarnos que el usuario no debe de hacer trampa, para ello vamos a implementar lo siguiente:

```
In [ ]: get '/win' do
  if @game.check_win_or_lose != :win
    redirect '/show'
  else
    erb :win
  end
end

get '/lose' do
  if @game.check_win_or_lose != :lose
    redirect '/show'
  else
    erb :lose
  end
end
```

1. Bloque /win

- Este bloque se ejecutara cuando alguien intente navegar directamente hacia /win
- Luego verifica si el jugador gana, si todavia no gana entonces se redirige a la ruta /show
- Si el jugador ha ganado, entonces se redirige a la vista asociada a win

2. Bloque /lose

- Este bloque se ejecutara cuando alguien intente navegar directamente hacia /lose
- Luego verifica si el jugador perdio, si todavia no pierde entonces se redirige a la ruta /show
- Si el jugador ha perdido, entonces se redirige a la vista asociada a lose

Validamos:

```
C:/Ruby26-x64/lib/ruby/2.6.0
C:/Ruby26-x64/lib/ruby/2.6.0/x64-mingw32. Reverting to Ruby lexer.
Feature: gain access to the word without guessing it

  As a adversary playing wordguesser
  So that I can see the word without ever entering a letter
  I want to navigate to a ending page and see the word

Scenario: navigate to lose page # features/cheating.feature:7
  Given I start a new game with word "snake" # features/step_definitions/game_steps.rb:9
  When I try to go to the URL "/lose" # features/step_definitions/game_steps.rb:57
  Then I should be on the show page # features/step_definitions/game_steps.rb:65
  And the word should read "-----" # features/step_definitions/game_steps.rb:38

Scenario: navigate to win page # features/cheating.feature:13
  Given I start a new game with word "snake" # features/step_definitions/game_steps.rb:9
  When I try to go to the URL "/win" # features/step_definitions/game_steps.rb:57
  Then I should be on the show page # features/step_definitions/game_steps.rb:65
  And the word should read "-----" # features/step_definitions/game_steps.rb:38

2 scenarios (2 passed)
8 steps (8 passed)
0m0.072s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 49 / 69 LOC (71.01%) covered.
```

Bien, ahora vamos a probar la ultima prueba, entonces ejecutamos el comando `cucumber features/game_over.feature`, la salida es la siguiente:


```
14 scenarios (14 passed)
56 steps (56 passed)
0m0.292s
Coverage report generated for Cucumber Features to C:/Users/user/Desktop/PrimeraPCDS/hw-sinatra-saa
s-wordguesser/coverage. 71 / 72 LOC (98.61%) covered.
```

En el siguiente enlace puede ver el proyecto completado:

<https://safe-beach-32833-ab16c02a534f.herokuapp.com/new> (<https://safe-beach-32833-ab16c02a534f.herokuapp.com/new>)

In []: