



UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE CIENCIAS

ESCUELA PROFECIONAL DE CIENCIAS DE LA COMPUTACIÓN

PC1 respuestas - Creación de aplicaciones SaaS

Pool Lennin Salvatierra Arevalo[20190282G]

Hector Ruben Bobbio Hermoza [20171071D]

Arturo Medardo Hinostroza Olivera [20191548K]

DESARROLLO DE SOFTWARE | CC3S2

I. OBJETIVO

Comprender los pasos necesarios para crear, versionar e implementar una aplicación SaaS, incluido el seguimiento de las librerías de las que depende para que sus entornos de producción y desarrollo sean lo más similares posible.

II. CREACIÓN Y VERSIONADO DE UNA APLICACIÓN SAAS SENCILLA

Para versionar el proyecto, creamos un directorio llamado 'SaaS' y ejecutamos 'git init' dentro de dicho directorio, como se muestra en la figura 1.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

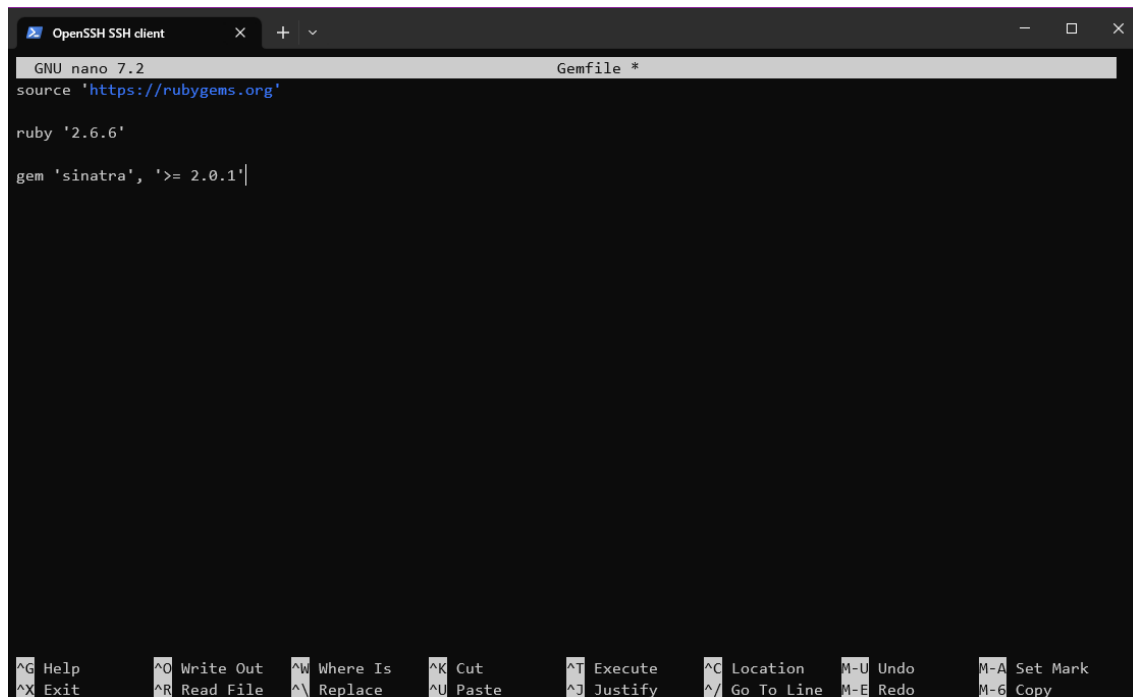
PS C:\Users\lennin254> ssh lenin254@192.168.18.17
lenin254@192.168.18.17's password:
linux debian 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
last login: Sun Sep 24 22:49:52 2023 from 192.168.18.5
lenin254@debian:~$ ls
cittireview  cv_debug.log  Documents  fonts          kitty  Pictures  RubyMine-2023.2.1  RubyMineProjects  venv  zoom_and64.deb
cookies.txt  Desktop       Downloads  julia-1.9.3-linux-x86_64.tar.gz  Music  Public    RubyMine-2023.2.1.tar.gz  Templates          Videos
lenin254@debian:~$ cd 0
lenin254@debian:~/0$ cd Documents/
lenin254@debian:~/Documents$ cd Github/P
lenin254@debian:~/Documents/Github/P$ cd Github/PC1/
lenin254@debian:~/Documents/Github/PC1$ mkdir SaaS
mkdir: created directory 'SaaS'
lenin254@debian:~/Documents/Github/PC1$ cd SaaS/
lenin254@debian:~/Documents/Github/PC1/SaaS$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/lenin254/Documents/Github/PC1/SaaS/.git/
lenin254@debian:~/Documents/Github/PC1/SaaS$
```

Figura 1: git init

Generamos un nuevo archivo llamado 'Gemfile' y copiamos el contenido mostrado en la Figura 2.



```
OpenSSH SSH client
GNU nano 7.2 Gemfile *
source 'https://rubygems.org'

ruby '2.6.6'

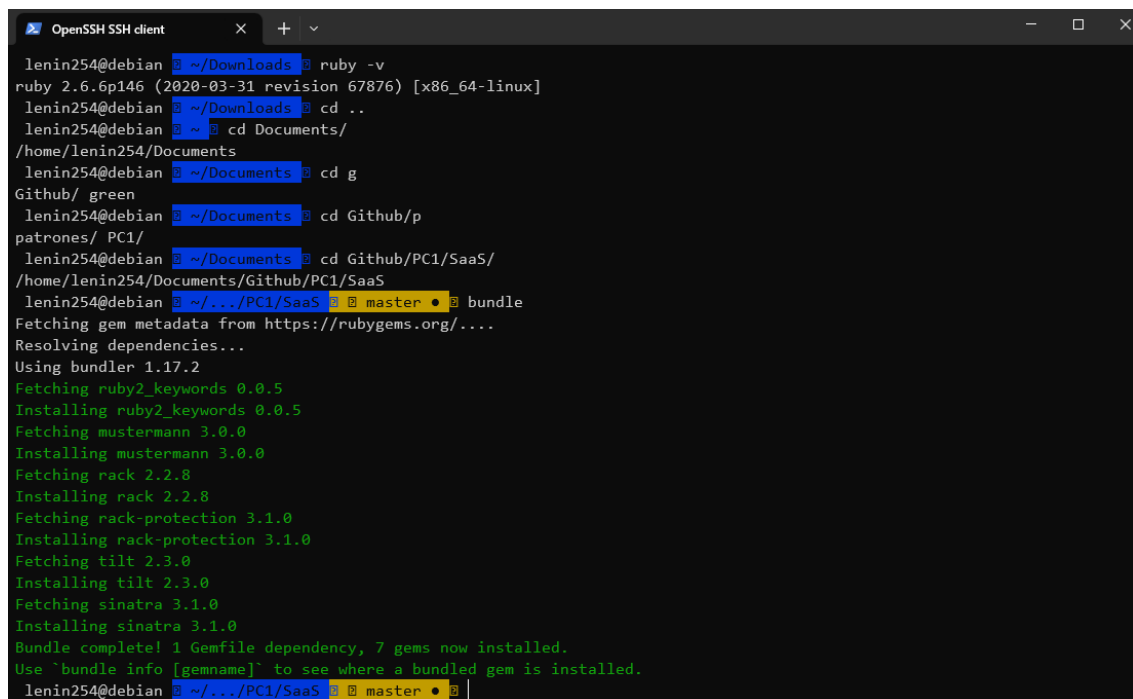
gem 'sinatra', '>= 2.0.1'

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo      M-6 Copy
```

Figura 2: Gemfile

I. Correr el Bundler

Ahora ejecutamos el comando `'bundle'`, esto se muestra en la Figura 3.



```
OpenSSH SSH client
lenin254@debian: ~/Downloads ruby -v
ruby 2.6.6p146 (2020-03-31 revision 67876) [x86_64-linux]
lenin254@debian: ~/Downloads cd ..
lenin254@debian: ~ cd Documents/
/home/lenin254/Documents
lenin254@debian: ~ cd g
Github/ green
lenin254@debian: ~/Documents cd Github/p
patrones/ PC1/
lenin254@debian: ~/Documents cd Github/PC1/SaaS/
/home/lenin254/Documents/Github/PC1/SaaS
lenin254@debian: ~/.../PC1/SaaS master bundle
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.
lenin254@debian: ~/.../PC1/SaaS master
```

Figura 3: bundle

Guardamos esta version con git y comentario `'Configurar el Gemfile'`

Preguntas

¿Cuál es la diferencia entre el propósito y el contenido de Gemfile y Gemfile.lock? ¿Qué archivo se necesita para reproducir completamente las gemas del entorno de desarrollo en el entorno de producción?

| | Gemfile | Gemfile.lock |
|------------------|------------------------------------------------------|-----------------------------------------------------|
| Propósito | Especifica las gemas y sus restricciones de versión. | Versiones exactas de gemas y dependencias. |
| Contenido | Lista de gemas y restricciones de versión. | Versiones específicas y resolución de dependencias. |

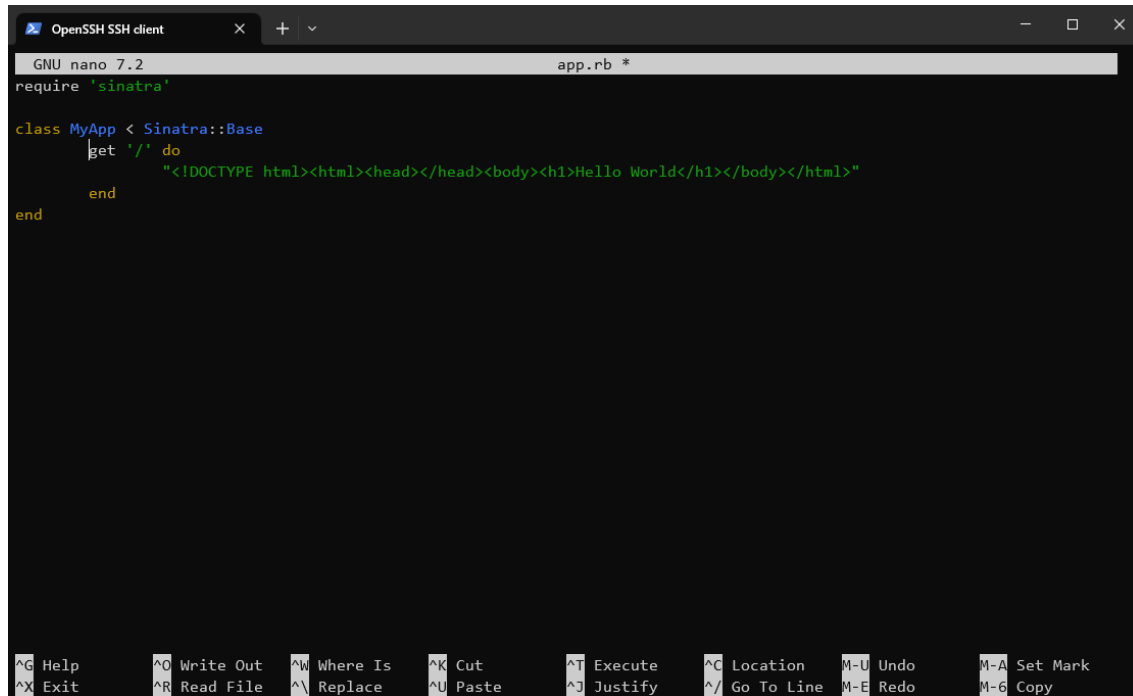
Para garantizar la consistencia de versiones de gemas en producción necesitamos ambos archivos:

- **Gemfile:** Especifica las dependencias y versiones deseadas.
- **Gemfile.lock:** Contiene las versiones exactas de las gemas y sus dependencias que fueron instaladas en el entorno de desarrollo. Al utilizar este archivo en producción, garantizas la consistencia de versiones.

Después de ejecutar el bundle, ¿por qué aparecen gemas en Gemfile.lock que no estaban en Gemfile?
Después de ejecutar 'bundle', gemas adicionales pueden aparecer en 'Gemfile.lock' que no estaban en 'Gemfile' debido a dependencias de las gemas instaladas. Estas dependencias pueden haber sido resueltas automáticamente por 'bundle' para asegurar la compatibilidad y funcionalidad correcta.

II. Crea una aplicación SaaS sencilla con Sinatra

Creemos el archivo 'app.rb' y completamos como se muestra en la figura 4.



```
OpenSSH SSH client
GNU nano 7.2 app.rb *
require 'sinatra'

class MyApp < Sinatra::Base
  get '/' do
    "<!DOCTYPE html><html><head></head><body><h1>Hello World</h1></body></html>"
  end
end
```

Figura 4: app.rb

Tambien creamos el archivo 'config.ru' y completamos como se muestra en la figura 5.

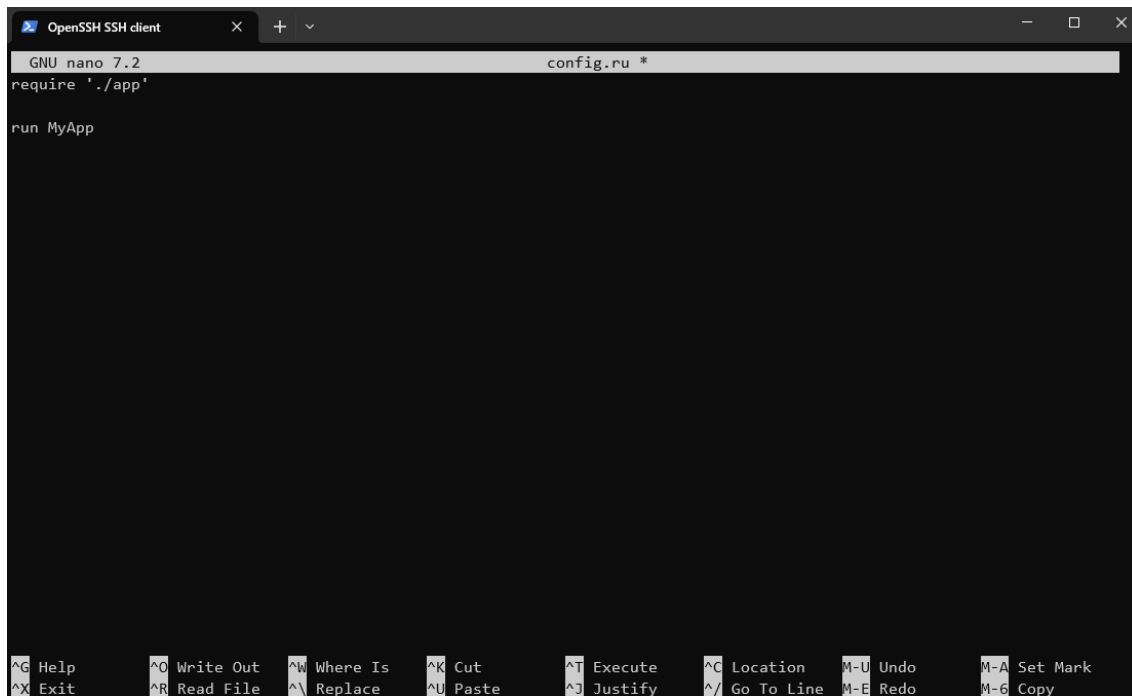


Figura 5: config.ru

Lanzamos el comando `bundle exec rackup -port 3000` como se ve en la Figura 6 y tambien podemos ver la pagina en la Figura 7.

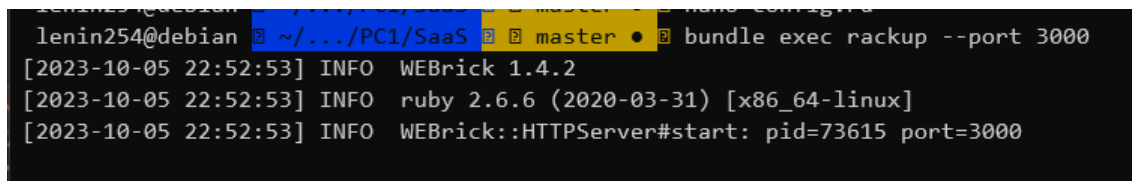


Figura 6: rackup 3000

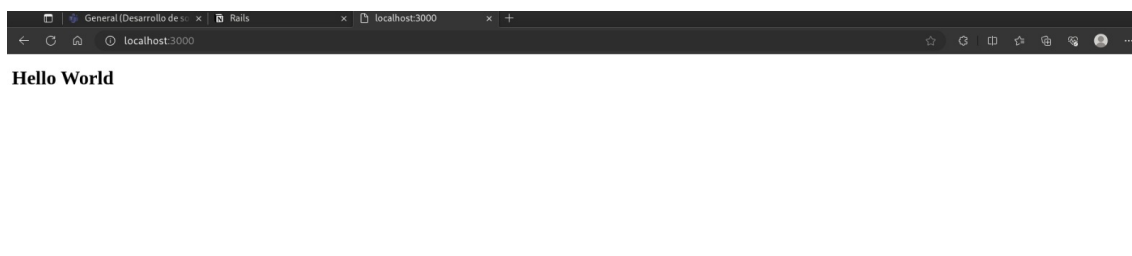


Figura 7: Hello Word

Preguntas

¿Qué sucede si intentas visitar una URL no raíz cómo <https://localhost:3000/hello> y por qué? Si intento visitar esa url tengo de salida el mensaje "Sinatra doesn't know this ditty", significa que Sinatra no ha definido una ruta para esa URL específica. Porque en la Figura 4 de `app.rb` espesifica `get '/'` do, para que ese url muestre algo, deberia cambiarse por `get '/hello'` do.

III. Modifica la aplicación

Modificamos `app.rb` para que muestre **Goodbye world** y que muestre en el navegador como la Figura 8.

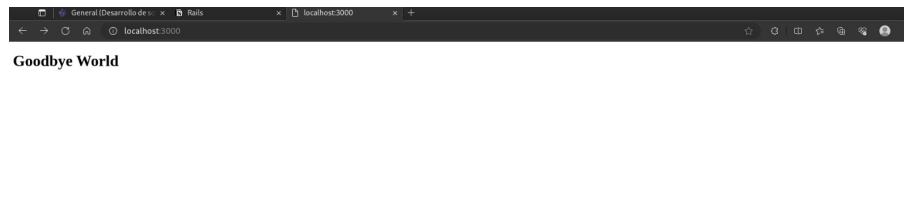


Figura 8: Goodbye Word

Agregamos al `Gemfile` lo de la Figura 9.

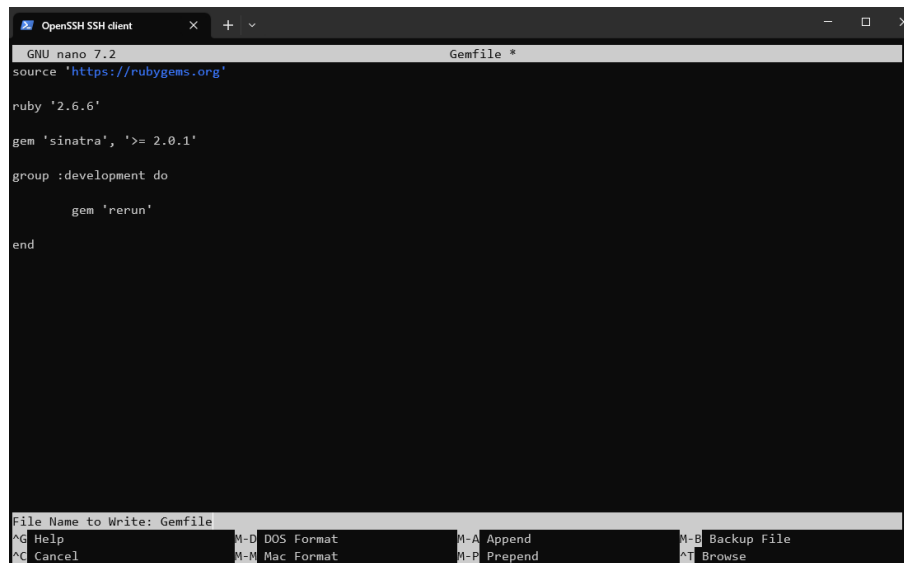


Figura 9: Gemfile

Guardamos y ejecutamos `bundle install` y corremos el comando `bundle exec rerun - rackup -port 3000` como se muestra en la Figura 10.

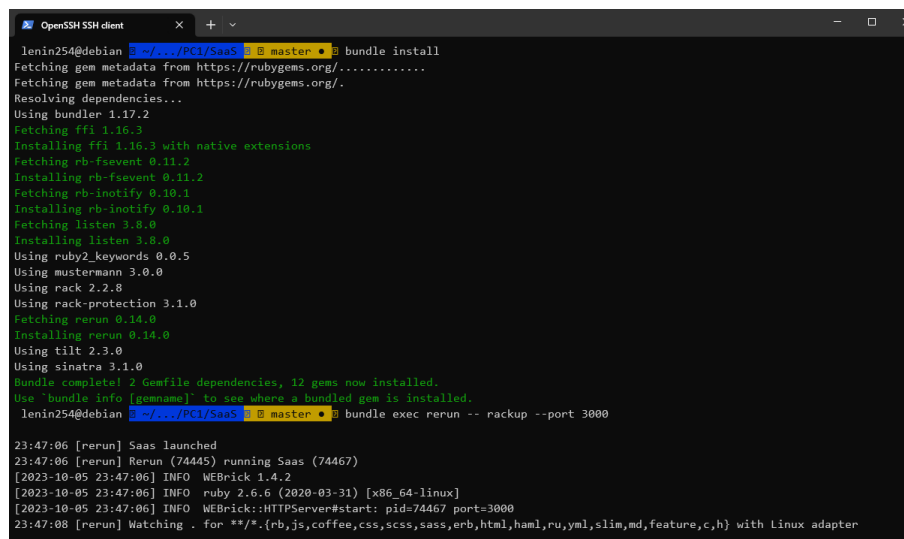


Figura 10: Output

y podemos verificar que cuando actualizamos algo para la pagina, se puede notar como en la figura 11 y que el comando sigue ejecutándose como se muestra en la Figura 12.

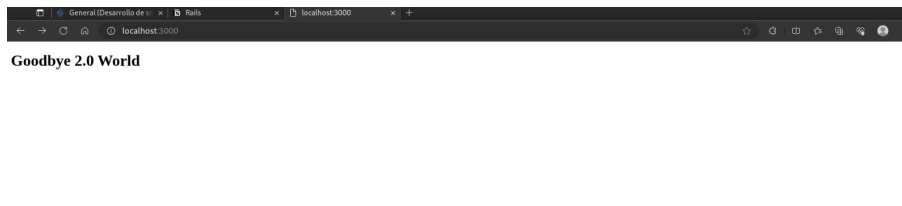


Figura 11: Pagina actualizada

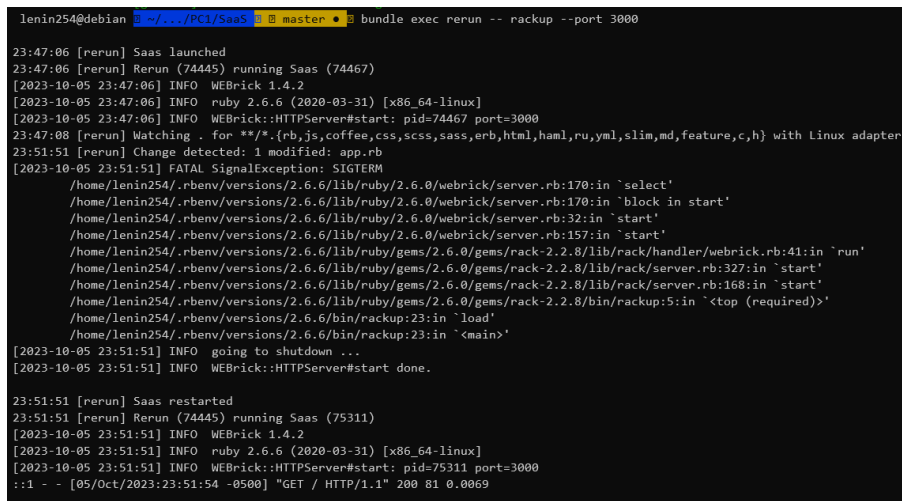


Figura 12: Ejecución de la terminal

IV. Implementar en Heroku

Despues de crearnos una cuenta en Heroku y instalar heroku, nos logeamos como en la figura 13.



Figura 13: login

y creamos un proyecto en heroku como en la figura 14.

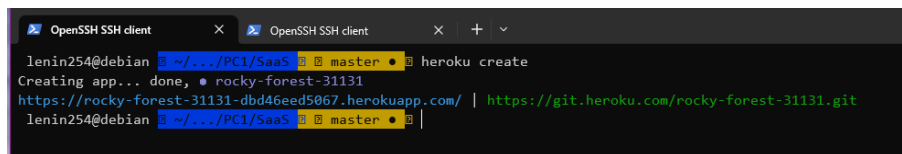


Figura 14: heroku create

Creamos el archivo llamado **procfile** y rellenamos como en la Figura 15.

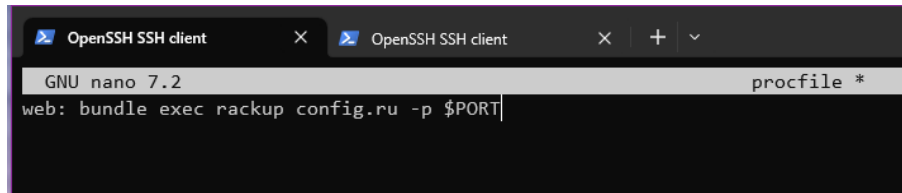


Figura 15: procfile

Ejecutamos 'git push heroku master' con heroku 20, como se ve en la Figura 16.

```
lenin254@debian ~/.../PC1/SaaS [master] heroku stack:set heroku-20
Setting stack to heroku-20... done
You will need to redeploy • rocky-forest-31131 for the change to take effect.
Run git push heroku main to trigger a new build on • rocky-forest-31131.
lenin254@debian ~/.../PC1/SaaS [master] git push heroku master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 501 bytes | 501.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Updated 6 paths from 6694660
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Using buildpack: heroku/ruby
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: -----> Punging Cache. Changing stack from heroku-22 to heroku-20
remote: -----> Installing dependencies using bundler 1.17.3
remote: Running: BUNDLE_WITHOUT='development:test' BUNDLE_PATH=vendor/bundle BUNDLE_
remote: Fetching gem metadata from https://rubygems.org/....
remote: Using bundler 1.17.3
remote: Fetching rack 2.2.8
remote: Fetching tilt 2.3.0
remote: Fetching ruby2_keywords 0.0.5
remote: Installing ruby2_keywords 0.0.5
remote: Installing tilt 2.3.0
remote: Installing rack 2.2.8
remote: Fetching mustermann 3.0.0
remote: Installing mustermann 3.0.0
remote: Fetching rack-protection 3.1.0
remote: Installing rack-protection 3.1.0
remote: Fetching sinatra 3.1.0
remote: Installing sinatra 3.1.0
remote: Bundle complete! 2 Gemfile dependencies, 7 gems now installed.
remote: Gems in the groups development and test were not installed.
remote: Bundled gems are installed into `./vendor/bundle`
remote: Bundle completed (1.01s)
remote: Cleaning up the bundler cache.
remote: -----> Detecting rake tasks
remote: ##### WARNING:
```

Figura 16: push heroku

III. PARTE 1: WORDGUESSE

Ahora clonaremos un repositorio remoto para crear un juego de adivinanza de palabras. Para esto, hemos incluido los archivos de este repositorio (que se puede encontrar en <https://github.com/saasbook/hw-sinatra-saas-wordguesser>) dentro del repositorio que ya hemos creado.

I. Desarrollo con TDD y Autotest

En esta sección haremos uso de la gema Autotest, que nos permitirá automatizar el proceso de testeo. Esta gema detectará el framework de pruebas que estemos utilizando, que en nuestro caso será RSpec, y se encargará de detectar algún cambio en los archivos de prueba en `spec/` y en los archivos de clase correspondientes en `lib/`. Para empezar a usarla, usamos el comando `bundle exec autotest`.

```
hector@hector-OMEN-by-HP-Laptop-16-b0xxx:~/Desktop/PC1_CC3S2/hw-sinatra-saas-wordguesser-master$ bundle exec autotest
(Not running features. To run features in autotest, set AUTOFEATURE=true.)
loading autotest/rspec
"/home/hector/.rbenv/versions/2.6.6/bin/ruby" -rrubygems -S "/home/hector/.rbenv/versions/2.6.6/lib/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "/home/hector/Desktop/PC1_CC3S2/hw-sinatra-saas-wordguesser-master/spec/wordguesser_game_spec.rb"
Run options: exclude {:pending=>true}

All examples were filtered out

Finished in 0.00023 seconds (files took 0.20726 seconds to load)
0 examples, 0 failures
```

Figura 17: Ejecutamos autotest

Observamos que detecta cero ejemplos y cero fallos. Esto se debe a que tenemos `Run option: exclude :pending =>true`, lo cual hace que autotest ignore los casos de prueba que tengan el parámetro `:pending =>true`. Este es el caso de todos los casos de prueba contenidos en `wordguesser_game_spec.rb`. Podemos observar que si removemos uno de estos parámetros de uno de los casos de prueba y guardamos el archivo, `autotest` detecta este cambio y volverá a ejecutar las pruebas.

```
Finished in 0.00023 seconds (files took 0.20726 seconds to load)
0 examples, 0 failures

"/home/hector/.rbenv/versions/2.6.6/bin/ruby" -rrubygems -S "/home/hector/.rbenv/versions/2.6.6/lib/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "/home/hector/Desktop/PC1_CC3S2/hw-sinatra-saas-wordguesser-master/spec/wordguesser_game_spec.rb"
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 WordGuesserGame object
     Failure/Error: expect(@game.word).to eq('glorp')
     NoMethodError:
       undefined method `word' for #<WordGuesserGame:0x0000563773328510 @word="glorp">
     # ./spec/wordguesser_game_spec.rb:16:in `block (3 levels) in <top (required)>'

Finished in 0.01125 seconds (files took 0.2019 seconds to load)
1 example, 1 failure

Failed examples:

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

Figura 18: Autotest vuelve a ejecutar las pruebas


```

Finished in 0.20188 seconds (files took 1.75 seconds to load)
18 examples, 18 failures

Failed examples:

rspec ./spec/wordguesser_game_spec.rb:13 # WordGuesserGame new takes a parameter and returns a WordGuesserGame object
rspec ./spec/wordguesser_game_spec.rb:28 # WordGuesserGame guessing correctly changes correct guess list
rspec ./spec/wordguesser_game_spec.rb:32 # WordGuesserGame guessing correctly returns true
rspec ./spec/wordguesser_game_spec.rb:41 # WordGuesserGame guessing incorrectly changes wrong guess list
rspec ./spec/wordguesser_game_spec.rb:45 # WordGuesserGame guessing incorrectly returns true
rspec ./spec/wordguesser_game_spec.rb:54 # WordGuesserGame guessing same letter repeatedly does not change correct guess list
rspec ./spec/wordguesser_game_spec.rb:58 # WordGuesserGame guessing same letter repeatedly does not change wrong guess list
rspec ./spec/wordguesser_game_spec.rb:62 # WordGuesserGame guessing same letter repeatedly returns false
rspec ./spec/wordguesser_game_spec.rb:66 # WordGuesserGame guessing same letter repeatedly is case insensitive
rspec ./spec/wordguesser_game_spec.rb:77 # WordGuesserGame guessing invalid throws an error when empty
rspec ./spec/wordguesser_game_spec.rb:80 # WordGuesserGame guessing invalid throws an error when not a letter
rspec ./spec/wordguesser_game_spec.rb:83 # WordGuesserGame guessing invalid throws an error when nil
rspec ./spec/wordguesser_game_spec.rb[1:3:1] # WordGuesserGame displayed word with guesses should be 'b-n-n-' when guesses are 'bn'
rspec ./spec/wordguesser_game_spec.rb[1:3:3] # WordGuesserGame displayed word with guesses should be '-----' when guesses are 'def'
rspec ./spec/wordguesser_game_spec.rb[1:3:3] # WordGuesserGame displayed word with guesses should be 'banana' when guesses are 'ban'
rspec ./spec/wordguesser_game_spec.rb:111 # WordGuesserGame game status should be win when all letters guessed
rspec ./spec/wordguesser_game_spec.rb:115 # WordGuesserGame game status should be lose after 7 incorrect guesses
rspec ./spec/wordguesser_game_spec.rb:119 # WordGuesserGame game status should continue play if neither win nor lose

```

Figura 19: Quitamos todos los pending=>true y se observan los 18 errores

Preguntas

Según los casos de prueba, ¿cuántos argumentos espera el constructor de la clase de juegos (identifica la clase) y, por lo tanto, cómo será la primera línea de la definición del método que debes agregar a `wordguesser_game.rb`?

Podemos observar por el caso de prueba para el constructor de la clase `WordGuesserGame`, que este constructor espera un argumento: `word`, pero un objeto de esta clase tendrá 3 variables de instancia: `word`, `guesses` y `wrong_guesses`. Conociendo esto, podemos modificar el método `initialize` de la clase `WordGuesserGame`. Para que esta pruebas puedan verificar que el objeto se está instanciando como debe, la prueba debe ser capaz de acceder a las variables de instancia del objeto, para esto usaremos `attr_accessor`. Podremos verificar que con estas especificaciones se pasa la prueba.

```

describe WordGuesserGame do
  # helper function: make several guesses
  def guess_several_letters(game, letters)
    letters.chars do |letter|
      game.guess(letter)
    end
  end

  describe 'new', :pending => true 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
end

```

Figura 20: Caso de prueba para `WordGuesserGame.new`

```

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

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

  attr_accessor :word, :guesses, :wrong_guesses

```

Figura 21: Constructor de la clase WordGuesserGame

```

(Not running features. To run features in autotest, set AUTOFEATURE=true.)
loading autotest/rspec
"/home/hector/.rbenv/versions/2.6.6/bin/ruby" -rrubygems -S "/home/hector/.rbenv/versions/2.6.6/lib
/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --tty "/home/hector/Desktop/PC1_CC3S2/hw-sinatra-
saas-wordguesser-master/spec/wordguesser_game_spec.rb"
Run options: exclude {:pending=>true}

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object

Finished in 0.00122 seconds (files took 0.25035 seconds to load)
1 example, 0 failures

```

Figura 22: Se pasa la prueba correctamente

A continuación se van a mostrar que las 18 pruebas pasan correctamente y la explicación de la estructuración del código se encuentra en [class WordGuesserGame](#).

```

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
  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 0.13683 seconds (files took 1.36 seconds to load)
18 examples, 0 failures

```

Figura 23: Se pasa la prueba correctamente

IV. PARTE 2: RESTFUL PARA WORDGUESSER

Pregunta

Enumera el estado mínimo del juego que se debe mantener durante una partida de Wordguesser.

La palabra oculta; el registro de letras adivinadas correctamente; el registro de letras adivinadas incorrectamente. De manera práctica, la clase WordGuesserGame, organizada de manera efectiva, guarda esta información utilizando sus variables internas, siguiendo las recomendaciones de un diseño apropiado basado en objetos.

I. El juego como recurso RESTful

Pregunta

Enumera las acciones del jugador que podrían provocar cambios en el estado del juego.

Adivina una letra: posiblemente modifica las listas de adivinanzas correctas o incorrectas; posiblemente resulta en ganar o perder el juego.

Comienza un juego nuevo: elige una nueva palabra y establece las listas de adivinanzas incorrectas y correctas como vacías.

II. Asignación de rutas de recursos a solicitudes HTTP

Pregunta

Para un buen diseño RESTful, ¿cuáles de las operaciones de recursos deberían ser manejadas por HTTP GET y cuáles deberían ser manejadas por HTTP POST?

Las acciones ejecutadas con el método GET no deben tener impacto en el estado del recurso. Por lo tanto, 'show' puede ser manejado con GET, pero 'create' y 'guess' (que alteran el estado del juego) deben utilizar POST. En realidad, en una arquitectura orientada a servicios completa, también podríamos optar por emplear otros verbos HTTP como PUT y DELETE, pero no profundizaremos en ese aspecto en esta tarea.

Pregunta

¿Por qué es apropiado que la acción New utilice GET en lugar de POST?

Porque la acción "new" por sí misma no provoca ningún cambio de estado: simplemente devuelve un formulario que el jugador puede enviar.

Rutas:

Las rutas para cada una de las acciones RESTful del juego, según la descripción de lo que debe hacer la ruta:

| Descripción de la Acción | Ruta |
|---------------------------------------------------------------------------------------------------------------|--------------|
| Mostrar el estado del juego, permitir que el jugador ingrese una adivinanza; puede redirigir a Ganar o Perder | GET /show |
| Mostrar el formulario que puede generar POST /create | GET /new |
| Iniciar un nuevo juego; redirecciona a Mostrar Juego después de cambiar el estado | POST /create |
| Procesar la adivinanza; redirecciona a Mostrar Juego después de cambiar el estado | POST /guess |
| Mostrar página de has ganado con botón para comenzar un nuevo juego | GET /win |
| Mostrar página de has perdido con botón para comenzar un nuevo juego | GET /lose |

V. PARTE 3: CONEXIÓN DE WORDGUESSER A SINATRA

Pregunta

En `app.rb`, `@game` es una variable de instancia de ¿Qué clase?

`@game` sería una variable de instancia de la clase `WordGuesserApp`. Esta variable de instancia se utiliza para mantener el estado del juego en la aplicación Sinatra.

I. La sesión

Debemos recordar que las solicitudes HTTP no tienen estado, para estas no existe una noción de "juego actual". Para objetos muy grandes (Las cookies tienen un límite de 4Kb) normalmente se requeriría una referencia a una base de datos, pero podemos aprovechar el pequeño tamaño de nuestra app para guardar nuestros objetos en la misma solicitud mediante las cookies. Para esto usaremos el hash `session[]` de la librería `session` de Sinatra.

Existe también otro objeto similar a una sesión: el hash `flash[]` que corresponde a la gema `sinatra-flash`. Esta nos permite recordar mensajes cortos a través de dos solicitudes HTTP para ser luego borrada.

Pregunta

¿Por qué esto ahorra trabajo en comparación con simplemente almacenar esos mensajes en el hash de sesión `session[]`?

`flash[]` tiene la ventaja de eliminarse automáticamente después de pasar por una redirección HTTP, mientras que `session[]` tendría que eliminarse manualmente.

II. Ejecutando la app Sinatra

Para ejecutar el programa usamos el comando `bundle exec rackup -port 3000`

Pregunta

Según el resultado de ejecutar este comando, ¿cuál es la URL completa que debes visitar para visitar la página **New Game**?

Tendríamos que visitar `http://localhost:3000/new` la cual es una página web local.

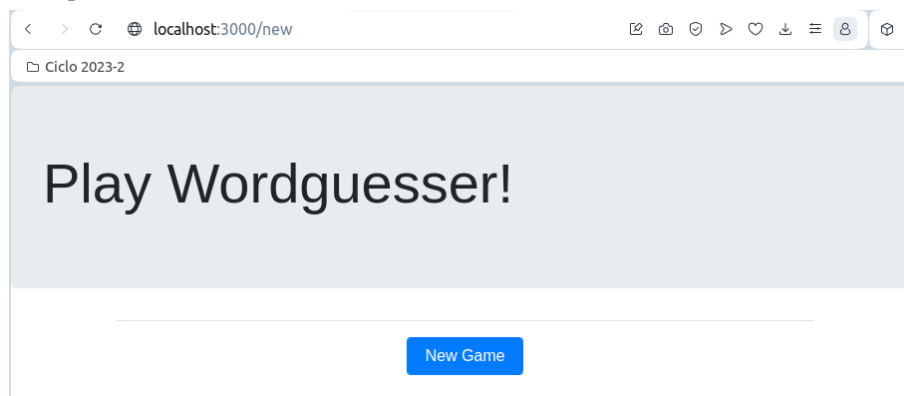


Figura 24: Aplicación Sinatra ejecutándose

Pregunta

¿Dónde está el código HTML de esta página?

El código HTML está en `new.erb` en la carpeta `views`.

Podemos verificar que presionar el botón de **New Game** nos arroja un error. Esto es porque no se ha definido dónde debe mandarse el formulario.

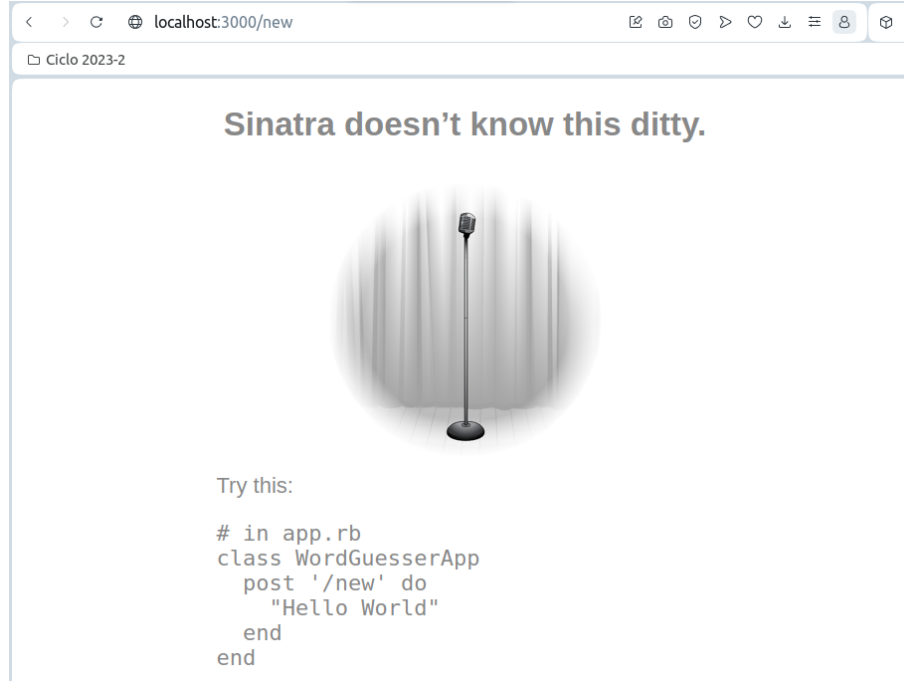


Figura 25: Error encontrado

git push heroku master

```
remote: There is a more recent Ruby version available for you to use:
remote:
remote: 2.6.10
remote:
remote: The latest version will include security and bug fixes. We always recommend
remote: running the latest version of your minor release.
remote:
remote: Please upgrade your Ruby version.
remote:
remote: For all available Ruby versions see:
remote: https://devcenter.heroku.com/articles/ruby-support#supported-runtimes
remote: ##### WARNING:
remote:
remote: EOL Ruby Version
remote:
remote: You are using a Ruby version that has reached its End of Life (EOL)
remote:
remote: We strongly suggest you upgrade to Ruby 3.1.x or later
remote:
remote: Your current Ruby version no longer receives security updates from
remote: Ruby Core and may have serious vulnerabilities. While you will continue
remote: to be able to deploy on Heroku with this Ruby version you must upgrade
remote: to a non-EOL version to be eligible to receive support.
remote:
remote: Upgrade your Ruby version as soon as possible.
remote:
remote: For a list of supported Ruby versions see:
remote: https://devcenter.heroku.com/articles/ruby-support#supported-runtimes
remote:
remote: -----> Discovering process types
remote: Procfile declares types -> web
remote: Default types for buildpack -> console, rake
remote:
remote: -----> Compressing...
remote: Done: 14.8M
remote: -----> Launching...
remote: Released v6
remote: https://rocky-forest-31131-dbd46eed5067.herokuapp.com/ deployed to Heroku
remote:
remote: This app is using the Heroku-20 stack, however a newer stack is available.
remote: To upgrade to Heroku-22, see:
remote: https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/rocky-forest-31131.git
```

Figura 26: output

heroku info

```
lenin254@debian ~/.../PC1/SaaS master heroku info
=== rocky-forest-31131
Auto Cert Mgmt: false
Dynos: web: 1
Git URL: https://git.heroku.com/rocky-forest-31131.git
Owner: pool.salvatierra.a@uni.pe
Region: us
Repo Size: 88 MB
Slug Size: 15 MB
Stack: heroku-20
Web URL: https://rocky-forest-31131-dbd46eed5067.herokuapp.com/
```

Figura 27: output

VI. PARTE 4: CUCUMBER

Cucumber es una herramienta para el desarrollo basado en comportamiento (BDD) que nos permite redactar pruebas de aceptación e integración de alto nivel (es decir, fácilmente entendible).

Cucumber se puede utilizar para probar diversos tipos de software, mientras se pueda simular el sistema y se pueda inspeccionar su comportamiento. Dado que para nuestra aplicación requeriremos solicitudes HTTP, y este comportamiento puede verificarse fácilmente en el navegador, configuraremos Cucumber para usar **Capybara**, un simulador de navegador basado en ruby con un DSL (Lenguaje específico de dominio) para simular acciones del navegador.

Pregunta

Mirando `features/guess.feature`, ¿cuál es la función de las tres líneas que siguen al encabezado **Feature:**?

Esta definiendo la historia de usuario de la funcionalidad (feature) a implementar.

En el mismo archivo, observando el paso del escenario **Given I start a new game with word garply** qué líneas en `game_steps.rb` se invocarán cuando Cucumber intente ejecutar este paso y cuál es el papel de la cadena `garply` en el paso?

Se ejecutarán las siguientes líneas:

```
8
9  When /^I start a new game with word "(.*)"$/ do |word|
10    stub_request(:post, "http://randomword.saasbook.info/RandomWord").
11      to_return(:status => 200, :headers => {}, :body => word)
12    visit '/new'
13    click_button "New Game"
14  end
```

Figura 28

En este ejemplo, `word` capturará la expresión regular definida entre paréntesis, la cual en este caso será **garply**

I. Primer escenario

Pregunta

Cuando el simulador de navegador en Capybara emite la solicitud de `visit '/new'`, Capybara realizará un HTTP GET a la URL parcial `/new` en la aplicación. ¿Por qué crees que `visit` siempre realiza un GET, en lugar de dar la opción de realizar un GET o un POST en un paso determinado?

Porque un usuario final no mandaría un POST a una URL antes de visitarla o interactuar con algún elemento de ella, como un botón o un campo de texto.

Ahora, ejecutaremos el escenario `new game` con el comando `cucumber features/start_new_game.feature`

```

hector@hector-OMEN-by-HP-Laptop-16-b0xxx:~/Desktop/PC1_CC3S2/wordguesser3/hw-sinatra-saas-wordguesser$ cucumber features/start_new_game.feature
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
    expected to find text "Guess a letter" in "Not Found" (RSpec::Expectations::ExpectationNotMetError)
    ./features/step_definitions/game_steps.rb:71:in '/^(?:|I )should see "([^\"]*)"(:? within "([^\"]*)"?)?$/':
    features/start_new_game.feature:11:in 'Then I should see "Guess a letter"'
  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

Failing Scenarios:
cucumber features/start_new_game.feature:7 # Scenario: I start a new game

1 scenario (1 failed)
5 steps (1 failed, 2 skipped, 2 passed)
0m0.046s

```

Figura 29: Primer escenario con Cucumber

Este primer escenario falla, pues falta completar el archivo `new.erb`. Realizando una pequeña modificación (agregarle un parámetro `action` al `form`) obtenemos lo siguiente:

```

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

```

Figura 30: Modificación al archivo `new.erb`

```

hector@hector-OMEN-by-HP-Laptop-16-b0xxx:~/Desktop/PC1_CC3S2$ cucumber features/start_new_game.feature
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)
0m0.043s

```

Figura 31: Escenario con Cucumber exitoso

Pregunta

¿Cuál es el significado de usar **Given** versus **When** versus **Then** en el archivo de características?
 ¿Qué pasa si los cambias? **Given**, **When** y **Then** son el mismo método, pero con diferentes "nombres" para darle mejor legibilidad a las historias de usuario. Podemos observar que intercambiar estas palabras no modifica el éxito del escenario.


```

hector@hector-OMEN-by-HP-Laptop-16-b0xxx: ~/Desktop/PC1_CC3S2$ cucumber features/start_new_game.feature
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
    Given 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
    Given I should see "Guess a letter" # features/step_definitions/game_steps.rb:70

1 scenario (1 passed)
5 steps (5 passed)
0m0.050s

```

Figura 32: Caption

Pregunta

En `game_steps.rb`, mira el código del paso `I start a new game...` y, en particular, el comando `stub_request`. Dada la pista de que ese comando lo proporciona una gema (biblioteca) llamada `webmock`, ¿qué sucede con esa línea y por qué es necesaria?

`Stub Request` es un comando que "sustituye" la respuesta de un recurso externo. Esto nos permite hacer pruebas con una palabra en específico, en lugar de escoger una palabra aleatoria cada vez que probamos algo.

git push heroku master

```

remote:      2.6.10
remote:
remote:      The latest version will include security and bug fixes. We always recommend
remote:      running the latest version of your minor release.
remote:
remote:      Please upgrade your Ruby version.
remote:
remote:      For all available Ruby versions see:
remote:      https://devcenter.heroku.com/articles/ruby-support#supported-runtimes
remote:
remote: ##### WARNING:
remote:
remote:      EOL Ruby Version
remote:
remote:      You are using a Ruby version that has reached its End of Life (EOL)
remote:
remote:      We strongly suggest you upgrade to Ruby 3.1.x or later
remote:
remote:      Your current Ruby version no longer receives security updates from
remote:      Ruby Core and may have serious vulnerabilities. While you will continue
remote:      to be able to deploy on Heroku with this Ruby version you must upgrade
remote:      to a non-EOL version to be eligible to receive support.
remote:
remote:      Upgrade your Ruby version as soon as possible.
remote:
remote:      For a list of supported Ruby versions see:
remote:      https://devcenter.heroku.com/articles/ruby-support#supported-runtimes
remote:
remote: ----> Discovering process types
remote:      Procfile declares types => web
remote:      Default types for buildpack => console, rake
remote:
remote: ----> Compressing...
remote:      Done: 14.8M
remote: ----> Launching...
remote:      Released v7
remote:      https://rocky-forest-31131-dbd46eed5067.herokuapp.com/ deployed to Heroku
remote:
remote: This app is using the Heroku-20 stack, however a newer stack is available.
remote: To upgrade to Heroku-22, see:
remote: https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/rocky-forest-31131.git
 d13a6a2..e7542d2 master -> master
lenin254@debian ~/.../PC1/SaaS [master] git push heroku master

```

Figura 33: output

heroku

Dentro de Heroku le damos en Open app

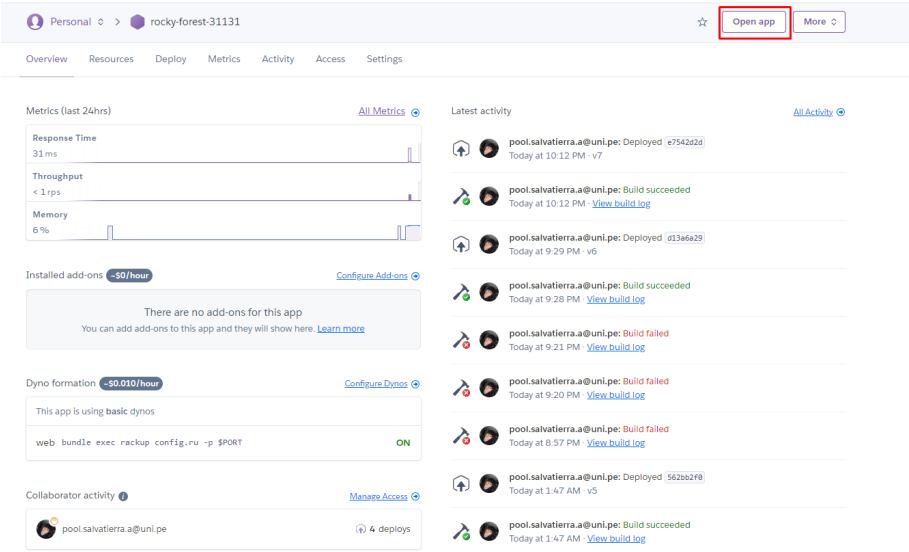


Figura 34: open app

Le damos a new game

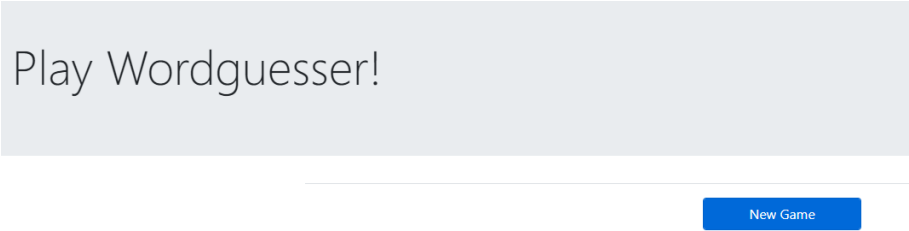


Figura 35: new game

nos muestra lo siguiente

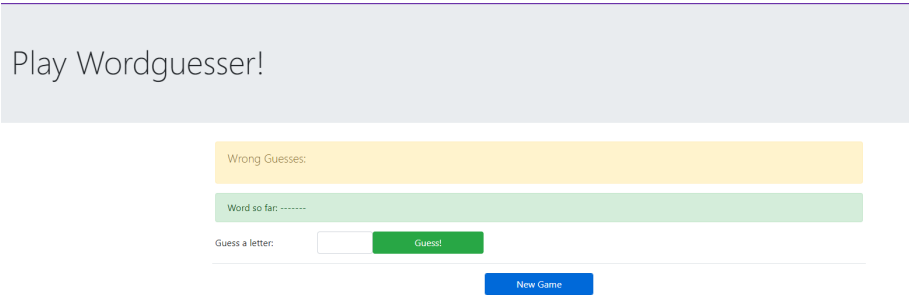


Figura 36: new game

VII. PARTE 5: OTROS CASOS

Play Wordguesser!

You Win!

The word was attract.

New Game

Figura 37: win

pool.salvatierra.a@uni.pe

hbobbioh@uni.pe

arturo.hinostroza.o@uni.pe

DOCUMENTO REALIZADO EN

L^AT_EX