

Testing Manual

Students:

Salvador Orozco Villalever	-	A07104218
Juan Andrés Reynoso Mazoy	-	A01328249
Ricardo Rodiles Legaspi	-	A01325081
Jesús Alejandro Coles Moguel	-	A01171039
Raúl Barranco Cortés	-	A01090517

Date: October 18th, 2018

Subject: Business Solution Development Capstone Project

Professor: Dr. Juan Manuel González Calleros



Testing Manual

Testing overview

The project's testing consists of the following modules:

- Unit testing for the controllers in the backend server
- Coding conventions and standards testing with enforced automation through ESLint with the Airbnb JavaScript style guide for the backend server
- Unit testing for forms, buttons and methods in the frontend
- Coding conventions and standards testing with enforced automation through Angular Lint

Tests run in the following two environments:

1. Locally, in a development machine
2. In Travis CI, our continuous integration solution, using an Amazon Web Services (AWS) virtual machine.

Commands for test execution

Backend:

- **npm test:** this command runs a background backend test server and executes the unit tests for the controllers. It was conceived for Travis CI.
- **test-local:** this command starts the execution of the unit tests for the controllers without running a background server. It is used with a development server.
- **npm run lint:** this command runs the coding conventions and standards check for the server.

Frontend:

- **ng test:** this command executes the Jasmin Angular tests
- **ng lint:** this command runs the coding conventions and standards check for the frontend Angular project.

Testing rules and conventions

1. Each pull request containing a behavior addition or modification, be it in the back or frontend, must include the corresponding tests that prevent regressions in those components.
2. The owner of the pull request whose changes modified the behavior of a component is responsible for modifying the component's tests to align them with the new expected behavior.
3. No pull request that is failing the Travis CI checks (builds, unit tests and coding standards tests) can be merged into the development branch.
4. The backend testing job in Travis CI must always be done with a new database, to ensure that all tests are run in the same conditions. For this reason, a script for recreating, remigrating and reseeding the database must be executed before the tests are run.