

# CARLABS FULLSTACK TECHNICAL TEST

---

You are going to create a SPA of a restaurant app that allows user to place some orders and review them. The backend is also all with you.

## WHAT ARE WE GOING TO EVALUATE?

---

- Technological decisions and documentation;
- Object Oriented Design;
- Respect SOLID principles;
- Readability;
- Maintainability;
- Tests quality and coverage;
- Capability to learn and adapt.

## GENERAL REQUIREMENTS

---

- You can use an external tool to document your project (like swag). But normally a well written README is enough;
- Keep code clear, only comment what is really necessary. Your code and tests should talk by themselves;
- Keep it simple, be careful to not over engineer. Write the necessary to have a good and clean code;
- Good practices counts;
- Usage of dependency injection and inversion of control (remember of SOLID) is highly appreciated;
- Push your solution in a Github repository. You can keep it public and use as portfolio, so give your best;
- In Carlabs, software engineers uses different O.S. (Windows, Linux and Mac), so your solution cannot be tied to a specific O.S. and should run in all machines.

## FRONTEND REQUIREMENTS

---

- Don't reinvent the wheel, use a SPA framework (Angular, React, Vue, etc);
- Must be decoupled from backend, so avoid use erb, razor, or any other template engine tied to backend;
- The main page must contain, at least, 4 components:
- input textbox (editable, single line)
- output textbox (readonly)
- button (send order)
- some data grid (read only)
- The grid must keep the history of inputs and outputs requested by the user. This data **must not** be persisted on a database;
- Frontend can have unit and integration tests (nice to have).

# BACKEND REQUIREMENTS:

---

- You can use any backend technology to develop this, but keep in mind that our backend stack is mainly C# and Javascript (nodejs);
- Feel free to use your preferred technology and tell us why this was chosen;
- If you does not have experience with C# or JS (nodejs) and you want to try it anyway, we are going to take in consideration that you have a learn curve, and this is a differential;
- Must have unit tests.
- Unit tests are live documentation of your code, so be carefull with their text.

---

TYPESET WITH [MARKDOWNER \(HTTP://MARKDOWNER.GITHUB.IO\)](http://MARKDOWNER.GITHUB.IO)