UEAT

Technical Test
Full-Stack Developer

Name: Luis Goyburo R.

The goal of this test is to learn more about who you are as a developer and your skills. There is no trick or puzzle. We look for people that can write robust and clean code, know the best practices and can work efficiently with their colleagues. We do not expect you to be an expert in all areas. We are looking for one strong area as well as a minimal understanding of other areas. Answer the questions with as many details as you can and if anything is unknown to you or unclear, do not hesitate to mention it in the answer.

1. **What new Web technologies have you learned recently? Can you talk a little bit about them?**

In the frontend stack, I have been working with **Tailwindcss.** Its an utility css framework that really improve the accurace and development of user interfaces. Its very fast to construct a site identity directly in markup. It is very intuitive naming their classes, and when their classes is not good enough, you can extend them through tailwind.config.js file.
It is very tiny in production because automatically remove unused classes when building. And if you are repeating groups of classes in your components, it is easy to create custom classes compuond of tailwind utility classes.

With respect to backend I was working in prototypes with **Nestjs**, a nodejs framework to build efficient an scalable server side applications. It is constructed with http server frameworks, Express (default) or Fastify (incredibly faster, up to 20000 request per second).
Once installed, you could work with any package manager, npm, yarn or pnpm. Automatically creates controllers, modules and services. By default works with typescript. And is relatively easy to implement REST APIs, Graphql with apollo, Microservices, Websockets and Stand alone applications.

2. **For you, what are the three most important qualities for a developer?**

Technical experience, as a result of a close relationship with the client. This experience must include the knowledge of a variety of languages and frameworks, so to have the authority to decide wich path to choose for a scecific task.This level of expertise will give productivity and value for the end client.

Fast Learning: In these times where technology varies every six months on average, it is radically important to have the ability to learn quickly and to be able to decide if a new technology will be productive or just a trending topic.

Soft skills: Today, where applications are increasingly complex and interconnected, it is essential that developers have soft skills to be able to work as a team. We refer to communication, leadership and empathy.

3. Can you talk about a few design patterns you appreciate? In what situation have you used them?

Design patterns do not depend on languages or platforms, are like blueprints that help developers to accomplish very different tasks.

Observer Design Pattern has a subject that contains a list of observers who are notified in the event of some changes in the state of the subject. In react this pattern is used to pass states from a parent component to a child component. In react this information is called “props”. Whenever a state changes, the component is re-rendered. When this state is passed to a child component as a prop, this child component will be re-rendered too. Here the observers are the child components, and the parent component is the subject.

Singleton Design Pattern. It is one of the most well-known patterns. In its core, it restricts a class to have just one instance and ensures that it is globally accessible. It might come in handy when you need to manage something from across your whole application. (loggers, global state).

Strategic Design Pattern. It is used when we have multiple algorithms for a specific task, and the user decides which implementation to be used ad runtime. Every checkout process has this implementation because there are multiple options to pay for a product or service, could be a credit card payment, cash payment, Paypal, etc.

Decorator Design Pattern attachs additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality. Frameworks like Nestjs, use intensively to give more funcionalities to basic code dinamically, wrapping them with decorators.

4. If you had a slow API call, what would be the methods to optimize it?

Assuming that the API is well design, implemented and operated, and the problem is the scalability or new business cases for the API, we have this options to improve the API call:

Cache usage is typical way of performance improvement. The purpose of the cache is to serve responses without recalculating them again. It is important to check if the data to serve is real time data, in that case, cache is not a good option. Another consideration is the knowledge of users behavior to get a high hit ratio. The cache can be implemented through http cache headers, application cache or distributed memory service like Redis and Memcached.

Proxy: Proxying the API lets scale better the original API. With the introduction of a new endpoint (the proxy) the proxy service can have multiple original APIs, if the database or the network it is not an issue.

Data denormalization / NoSql: When the database is the bottleneck, it is an alternative to use a NoSql database. This could reduce the number of queries and to provide rich responses. The document data store could have in a single point all the information of a business entity.

Graphql: If the API is a REST API and the data is complex, so it is necessary to have multiple calls to get all the data, Graphql could be a better alternative because it has a query capability, so in just one request could serve all the data.

5. **You develop a Web page with pretty interactive animations. Unfortunately, those animations are not supported with Internet Explorer 9 and the page does not load. If supporting IE9 was a requirement, how would you solve that?**

If IE9 is a requirement, the animation could be done with javascript. Using setInterval and clearInterval and modifying the style settings of the element in time. The element must be in a container with position relative, and the element must have position absolute.
For more elaborated animations, we could use Paul Irish's polyfill, https://gist.github.com/paulirish/1579671. This polyfill implement requestAnimationFrame and cancelAnimationFrame in javascript for IE9.

6. **Most of the users for an app are in an area where the network is unstable. With an app mainly used on mobile, how can you make sure it keeps working when the network connection is lost? What strategy would you use?**

Firstly, we need to know which features of the app will be functioning off line. Prioritizing the features that are essential to the business.
Once we choose the features that will be alive off line, we need to decide how we will collect and store data for offline mode in web and mobile app: service workers, IndexedDB, shared preferences, NSUserDefaults, Local storage, SQLite Database. Then, how we will sync the data worked offline with the server when the network is available in web and mobile.

Another more structural alternative is to convert the app to a Progressive Web Application.

7. **Can you describe what is a deadlock? How can you avoid making one?**

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.
Deadlock can arise if the following four conditions hold simultaneously.
Mutual Exclusion: One or more than one resource are non-shareable (Only one process can use at a time)
Hold and Wait: A process is holding at least one resource and waiting for resources.
No Preemption: A resource cannot be taken from a process unless the process releases the resource.
Circular Wait: A set of processes are waiting for each other in circular form.

In case of a deadlock happen for a lock resource, to avoid the deadlock, the processes must acquire a lock in a fixed order. To illustrate this: If process 1 gets the lock on resource R1 and then R2, at the same time, process 2 also tries to get the lock on resources in the same order as process 1, i.e. On resource R1 and then R2 instead of R2 and then R1.

8. **When is it relevant to add data to a cache? What would be the consequences on the rest of the system?**

Cache is a data store which purpose is to serve resources at a relatively fast speed. In software cache, generally cache are blocks of memory that store data, so it is faster than accessing IO. Managing in this way, cache will increase the throughput and decrease the latency.

However, cache does not increase throughput in all kinds of systems. If the system needs to update or write data frequently, data in cache would need to be constantly changing too, so the application must to mark cache as invalid, and write the updated data to its original data store. This unnecessary process become an overhead. Cache is beneficial to applications that are read-heavy but not write-heavy, where it has a negative impact on the system.

9. What would be the best way to save a user password and making sure it is safe from hackers?

The best way to save sensible data is to use cryptography. We could use Hashes and Salts to store the representation of the password in the database. This way is practically impossible to the hackers to get the password. Of course this must to be done trough https connection.
A hash function is a one-way function that will always return the same value for a given input. Before calling the hash function we need to add a random data (salt) to the password. This way, is very difficult to reverse the hash function.

10. What are the benefits of Cloud Hosting (AWS, Google Cloud, Azure) vs Self Hosting?

If you mean self-hosting as On Premise, this are the advantages to use Cloud Hosting:

- **Capital expense vs variable expense.** You do not have to invest in infrastructure to have your services running.
- **Economies of scale.** It is very difficult to have good hardware prices and low overhead with just one data center. With cloud, you take advantage of economies of scale.
- **Planning a capacity is no longer an issue.** When you make a capacity decision before to deploying a system, you often end up either buying expensive idle resources or dealing with limited capacity.
- **Fast resource deployment.** In cloud computing environment, new resources are just waiting for setup, this result in dramatic increase in agility for the organization. The response time is very short.
- **No more maintenance and operating data centers.** The infrastructure is no longer an issue, just focus on business.
- **Global with minimal costs.** Deploy your systems in multiple regions around the world with just minimal setting.

11. How can you make sure the deployment of a new version does not create other bugs?

Through Canary Deployments:
“Canary deployments are a best practice for teams who have adopted a continuous delivery process. In a canary deployment, a new feature is first made available to a small subset of users. The new feature is monitored for several minutes to several hours, depending on the traffic volume, or just long enough to collect meaningful data. If the team identifies an issue, the new feature is quickly pulled. If no problems are found, the feature is made available to the entire user base.”

12. Describe how you solved a major problem in a past project :

When I develop a project, generally I do not have “major problems”. I used to have minor problems, and they are corrected on the fly, but when I have encounter a Big Problem, it is because the platform or the stack is not fitted to the project, so I change the stack.
I was developing a website for a NGO and with the additions of plugins the site turn to a slow performance, so I change from an apache-wordpress stack to an nginx-node-js-reactjs-nextjs website. The indicators improve from 45 % of speed to 85% of speed.

13. Write in your language of choice a function that converts a roman number *string* to an *integer* . Make sure to build it in a way that will be future proof to new specifications without adding unnecessary complexity. Also make sure it is clean and easy to understand for another developer.

The following symbols represent the values.

I = 1
V = 5
X = 10
L = 50
C = 100
D = 500
M = 1000

Each character adds a number. However, those are the exceptions :

I can be before V and X to be 4 and 9
X can be before L and C to be 40 and 90
C can be before D and M to be 400 and 900

Example 1: XXI is 21
Example 2: XL is 40
Example 3: LVIII is 58

The code can be written within an editor and sent by email in a ZIP.

14. Write unit tests that make sure the function from the previous question is working properly. Make sure to have 100% test coverage and include all possible test cases. Include it in the same ZIP.