**About me:**

My name is Mauricio, I study computer systems engineering at ITESO located in Guadalajara. I find a passion in technology ever since I was very young. I started getting into computer science when I was 12 and started to lookup ways to manipulate computer games to my benefits. I used to play Age of Empires 2 and learned about cheat codes, then in Facebook games such as pet society and Farmville I started to use cheat-engine where you “program” the game to modify certain attributes such as resources and things related.

Something that I really enjoy is pushing myself with hard projects or assignments that will require a lot of effort and from that I’ve learned quite a bit.

Questions for my interviewer?

Vtex

Multitenant architecture

Elastic cloud

How do you manage B2B

How long have you been working at VTEX?

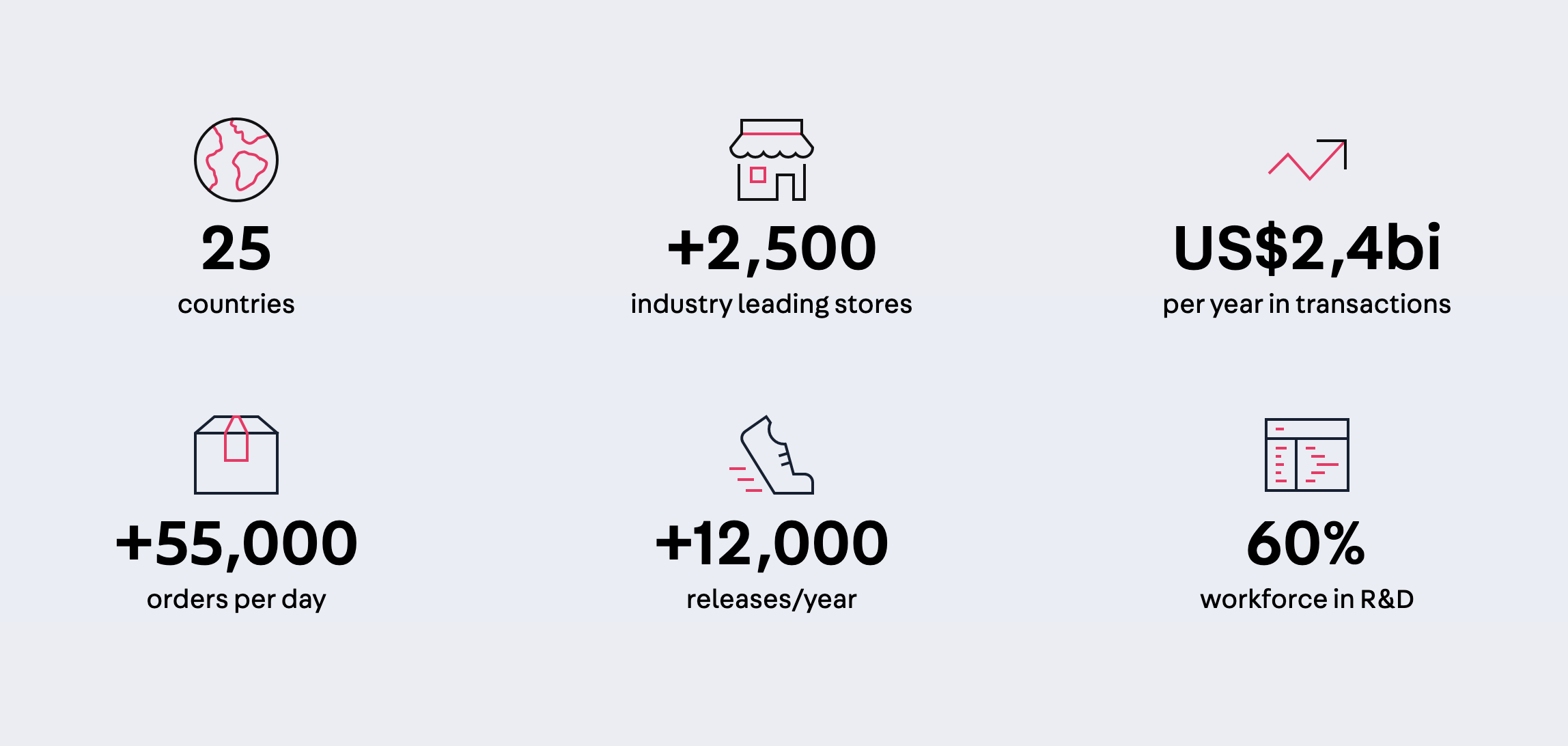
What has been your biggest challenge?

How is the work environment over there?

What tips would you give in order to thrive in VTEX?

How do you test and handle all the updates done to the VTEX tech?

There are new technologies coming out every day, which is wonderful and at the same time crazy. How do you adapt to the ever-changing technological environment?



We experiment with new ways to work, new team compositions and workflows all the time. We believe in an organizational structure that eliminates the need for managers, bosses or any kind of formal hierarchy.

Why I like Microsoft?

1. It is a company that I’ve interacted with my whole life
2. I value how Microsoft has the purpose of creating and changing the world in any way possible. I am amazed with how you can see Microsoft’s efforts in every single area, from Cloud Computing with azure to A.I to Videogames.
3. I’ve noticed the Microsoft’s openness to the community, I see how the users are able to contribute to Microsoft and in the same way I see how Microsoft gives back to the community.
4. I perceive Microsoft as a company that is looking to make change instead of just being the number one company and earning a lot of revenue.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mancala | Image Recognition | Poker Game |
| Most Challenging | Having to work with someone else’s code, the A.I. TA’s developed the game engine and some other classes and they were not very well documented and some of the variable names were not very descriptive. | The most challenging part of the project was coming up with a way of doing the required operations of a neural net. After doing some research I know how the algorithm worked but not how to implement it. My partner and I did some brainstorming and I came up with the idea of doing all the operations with matrices. | Developing a naïve PC agent to play against turned out to be quite challenging. The first step was to create a method to evaluate the quality of our hand and then the agent could choose to swap cards given certain conditions. |
| What you learned | I learned that even when you have a good agent domain knowledge can be key to improving performance in this case I modified my initial heuristics and also added non determinism to the simulation part of the project this resulted in a 10% increase on my accuracy. | I learned the workings of Neural Nets and about its blackbox nature. This was one of the projects where I learned how truly important math is in computer science. The implementation of the neural net module requires a lot of linear algebra as well as multivariate calculus for the gradient descent. | I learned about the usefulness of data structures and how they can be used to write more understandable code and make the application more modular. |
| Most interesting | UCB was something that I consider highly useful and interesting as it was a way of balancing exploration and exploitation to learn the best move. | The training process of the neural net. How we need data to train and then after that how a validation dataset is also needed and then we proceed to the actual testing phase to see the accuracy of the model. | The sorting function and the evaluation function, because this project was for an introductory course I used bubble sort algorithm and from the sorted hand I was able to design the evaluation function. |
| Hardest Bug | In the initial phase of development I found that my agent was taking paths that were very unconventional and yielded poor results. I reviewed the code and the printed graphic tree (method implemented by TA’s) and saw that my agent always considered that it was my turn. The solution for this was to negate myMove variable in each simulation move so that the agent alternated between PC’s move and mine. | After we had our neural net module implemented we proceeded to train the model to do image recognition on vowels but I realized that when the training was stopped the model would only predict the letter e and it would do so for any kind of input so I decided to train the model again with different epochs but the result was the same. We had already tested the neural net module with easier versions of letter recognition and that worked.  I looked at the code again and noticed that we were using the validation set as testing set and that made me understand why the net acted like it did. | Had a hard time working with the hands because sometimes they needed to be passed into a function where the program did not have access to them, we solved this by making an index array for the user’s and PC’s cards and we made that array a global variable which solved our problem. However, if I could’ve done things differently, I wouldn’t have made those global variables. |
| Enjoyed most | I enjoyed getting into the basics of the game and finding common strategies to beat the opponent. | I really enjoyed the preliminary part which was studying neural nets, I rented three books and downloaded some slides to get a grasp on how neural nets operate. I focused on the theory for an entire week until I felt confident enough to code what I now knew. | I enjoyed designing the structures used in the game such as the card, the deck which contains 52 cards, and a hand that is a collection of 5 cards for the user and for the pc. There was an attribute called inUse which made sure that no card could be used twice in a game. |
| Conflicts with Teammates | This was a individual project, however I discussed the given problem with my peers to try to come up with different ideas to solve the task. | So there were two parts of the project the first one was building the module for neural nets and then the second part was doing the image recognition and obtaining the datasets for training. My team member and I both were mainly interested in the implementation part. Even though I greatly desired solving every single part of the neural net I understood that doing this would either cause a problem in the team or we would end up neglecting the image recognition part so I made a proposal to my team mate, I told him that we should go over the logic of the neural net together and list all the tasks that needed to be done we divided the tasks so that we could both work on both parts of the project and that we would help each other if we stumbled upon a problem. This experience made me learn two things, first that two minds work better than one and my peers are fully capable of getting the job done.  And Second, that the collection of small easy tasks (in this case the image recognition part) are as important as the big complicated algorithms. | My teammate and I had different views on how we wanted the end result to be. He wanted to make a betting system where the user would start out with some money and he could play indefinitely until the user did not have any money left. I, on the other hand wanted to focus my efforts on making the PC intelligent so it could do good moves. The time constraints made it difficult for both features to be implemented.  I recognized that the project was not all mine so I discussed the possible options with my teammate and we both compromised to get to a project where both of us would be content. |

Greatest Weaknesses.

One of my greatest weaknesses is that I’m too eager to solve problems as fast as possible. This may sound like a trait, but it is not because that can result in me omitting or ignoring special cases or the optimal solution. What I do to overcome this is take a deep breath, relax and make sure that I’m thinking of everything I question myself about possible outcomes and evaluate my thought process to see If what I’m thinking works or if I should explore alternatives.