

# RELATÓRIO PIF

à

- a) O tópico da linguagem C que foi mais difícil para mim foi ponteiros. Aprender ponteiros, desde a teoria até a prática, foi um processo bem difícil pra mim. Devido à minha falta de atenção na aula de introdução a esse assunto, eu não tinha entendido para que servem os ponteiros e, com isso, eu acabei ficando para trás na disciplina, sem conseguir acompanhar a maior parte das aulas seguintes. Apesar dessa dificuldade inicial, eu decidi correr atrás dos assuntos perdidos, e vi uma analogia que me ajudou muito no processo de entender o funcionamento dos ponteiros. Essa analogia consiste na comparação de ponteiros com pinos em um mapa. Os pinos encontrados no mapa apontam para uma localização, nesse caso, um endereço, não o local em si, mas o endereço dele; Esses pinos seriam os ponteiros, que estão apontando para uma variável. Já no caso de Haskell, eu não diria que a minha maior dificuldade veio de um tópico específico, mas sim da forma completamente diferente de se programar uma linguagem funcional. Devido ao fato de programar em Haskell ser completamente diferente de em qualquer outra linguagem que eu tinha estudado anteriormente, eu senti muita dificuldade em conseguir fazer até mesmo códigos simples. Apesar disso, após me acostumar mais com o estilo de programação mais "matemático" de Haskell, eu consegui fazer os exercícios da disciplina sem muita dificuldade. O tópico mais fácil de C pra mim, foi, sem dúvidas, listas encadeadas. Devido ao meu esforço para entender os ponteiros, eu acabei criando gosto pelo assunto, e no dia que tivemos a aula sobre listas encadeadas, eu decidi desenhar o funcionamento delas em um papel, o que facilitou bastante minha compreensão do assunto. Na própria cadeira de projetos, eu fiz uma lista encadeada que se comunicava com o nosso site em Django através de um arquivo .txt que seria escrito em C, e lido em Python. O assunto mais fácil de Haskell para mim foram as listas. Devido ao nosso aprofundamento em listas ao longo da cadeira, no momento que nós tivemos que aprendê-las em Haskell, eu já estava bem familiar com a parte teórica do assunto, então só me restou aprender a sintaxe de declaração de listas e de sua divisão em Head (primeiro elemento) e Tail (resto da lista).
- b) Eu diria que o ponto mais forte da disciplina foram os assuntos dados e o quanto eles me fizeram crescer ao longo do período. Tudo que eu aprendi ao longo da disciplina, especialmente sobre C, me tornou um programador mais completo, mais confiante. Programação de baixo nível, nesse caso, em C, é fundamental para a compreensão dos conceitos da computação e acredito que com tudo que me foi ensinado houve uma grande expansão do meu conhecimento sobre a área. A minha principal sugestão para melhorar a cadeira foi dada ao longo do período, que era a de aumentar o número de exercícios feitos em classe. Da metade para o final do período, nós fizemos

bastantes exercícios em classe, mas acredito que seria ainda melhor aumentar mais ainda o número de atividades realizadas.

- c) Meu desempenho nessa cadeira deixou a desejar, especialmente no começo do período. Devido à minha falta de disciplina, eu deixei diversos assuntos acumularem e, com isso, acabei criando uma “bola de neve”, que me atingiu com o resultado das primeiras notas. Apesar disso, essa derrota foi o que me fez melhorar meu desempenho. Mesmo com o meu péssimo primeiro resultado, em momento algum eu cogitei desistir e trancar a cadeira, eu decidi dar o meu melhor e seguir em frente, e foi isso que eu fiz. Comecei a prestar mais atenção nas aulas, tirar minhas dúvidas e correr atrás dos assuntos perdidos. Fiz diversos cursos de programação em C na internet e consegui recuperar os conceitos básicos que tinham me escapado, como ponteiros, e com isso, acabei não só aprendendo bastante, mas também criei um gosto maior por programação em C e, conseqüentemente, programação como um todo. A principal sugestão que eu daria para mim no começo do período seria de não acumular assuntos, não procrastinar, e no caso de futuros novos alunos, a minha principal sugestão é de não desistir, por mais complexos que os assuntos pareçam, com o devido esforço, você vai aprendê-los, e, com isso, se tornar um programador melhor.