

Transcrição F1

Função: Líder de Desenvolvimento do Observatório

Tempo de experiência : 8 meses

Seção 1_

(início de entrevista)

Daniel: Qual função você exercia na FAPESB?

F1: Eu era, no caso do projeto do Observatório, eu era líder técnico do projeto.

D: Então você fazia o que o líder da equipe fazia? Ou diferente?

F1: Na verdade ele era mais o cara que cuidava da equipe [de desenvolvimento] como um todo. Eu era o cara que cuidava desse projeto especificamente, tecnicamente.

A: Ele era mais gerencial da equipe de TI e você era o líder do projeto.

F1: Isso, ele cuidava mais da gerência da equipe como um todo, eu cuidava desse projeto especificamente. (Entrevistado lista outros projetos em que ele não estava encarregado). Eu cuidava apenas desse projeto, eu fui lá pra fazer esse projeto.

A: Então você era como um desenvolvedor líder do [projeto].

F1: Isso! Mas não só desenvolvedor. Eu desenvolvía, mas eu também era quem fazia toda a parte de engenharia do que ia acontecer.

A: Levantamento de requisito, essas coisas?

F1: Isso, o que é que tinha que ser feito, a validação dos dados, como é que ia ser feito e tal. Isso quem decidia era eu.

D: Você trabalhou nos dois projetos, no Observatório e no Sistema de Bolsas.

F1: Não, eu só trabalhei no Observatório. No Sistema de Bolsas eu dei uma ajuda aqui e ali, mas eu não trabalhei.

D: Entendi. Você já tinha desenvolvido em algum sistema que tinha que mexer com essa parte de descoberta de dados, analytics?

F1: Não.

D: Foi a primeira vez?!

A: E você ficou quanto tempo lá na FAPESB com esse projeto?

F1: Bom, eu fiquei do início ao fim. Eu entrei quando começou e eu saí quando ele terminou. Agora exatamente quanto tempo, eu não sei dizer. (Entrevistado faz as contas)

A: 9 meses.

F1: É, acho que mais ou menos. Bota 8 que eu acho que é mais seguro.

D: A gente tem uma perguntinha básica, que a gente pergunta pra todo mundo, que é pra pessoa falar sucintamente como é o sistema do observatório?

(Entrevistado confirma a pergunta e o entrevistador corrige)

A: É só fazer um resumo do que é o sistema.

F1: Você quer que eu diga o que é o sistema do Observatório?!

A e D: Isso!

F1: O observatório da FAPESB é um sistema feito para poder mostrar, adquirir a transparência dos dados das bolsas e dos fundos aplicados na pesquisa pela FAPESB. Então ele é um sistema, o objetivo maior do sistema é justamente mostrar, é abrir as contas, não diria abrir as contas. Seria minerar o banco da FAPESB, legado, e mostrar tudo que tem sido investido na FAPESB ao longo dos anos. Criar esse panorama e criar indicadores pra poder estimar os próximos passos das FAPESB. Ou seja, se eu tô vendo, sei lá, um dado... Um exemplo, que era uma das coisas que o presidente queria, se eu tô vendo aqui que aplicar tantos milhões, em bolsas de determinada área, está rendendo tantos doutores, é melhor que eu aplique mais nesta área do que aplicar na área X, por exemplo. Então esse é o maior objetivo da FAPESB. Era mostrar esses dados que tinham, o que já tinha sido investido e também estimar os próximos passos de como é que o dinheiro da FAPESB ia ser aplicado.

D: Beleza. E pra você fazer esse observatório, como era o processo de engenharia de software? Por exemplo, de requisitos até você fazer a parte de teste e falar “O software tá legal agora. Agora por ir [ao ar], pode subir o sistema [para produção]”. Como foi esse processo?

F1: Bom, falando em processo de engenharia de software, um processo definido a gente não tinha. A gente usava Scrum como metodologia de desenvolvimento, mas assim, no caso do Observatório (passagem não compreendida), a equipe como inteira usava Scrum. Só que no Observatório eu trabalhava praticamente sozinho. Então eu não usava Scrum, porque eu trabalhava sozinho. Não tinha porque ter Scrum se tinha um grupo só. Então o que acontecia, quando a gente tinha as reuniões de Scrum, de todo mundo, eu participava também e dizia o que

eu tava fazendo. Mas isso não implicava na atividade de ninguém, até porque depois que eu saí muito pouca gente mexeu no código do observatório, que não eu. Então metodologia de engenharia de software a gente não tinha. Até porque a gente não tinha tempo. (Entrevistado explica a questão do prazo para o sistema). Então a gente realmente não teve tempo de fazer uma engenharia de software melhor. No final do projeto, como sobrou um tempo, entre a validação de dados, que é a parte que eu vou falar daqui a pouco, e o lançamento oficial do sistema, eu comecei a fazer engenharia reversa de alguns artefatos. Principalmente aqueles que envolvessem mais diretamente o código e a arquitetura do projeto. Porque na época, antes de eu saber que eu ia sair de vez da FAPESB, eu ia sair de férias, então durante as minhas férias alguém ia ter que resolver algum problema do sistema e o conhecimento do sistema tava todo comigo. Então o que é que eu fiz, eu gerei no final alguns diagramas de sequência das funções mais críticas, tinha a arquitetura do projeto que existia e tinha um diagrama de classe. Quanto à teste, os testes eram apenas funcionais, não foi feito teste unitário, nem nada do tipo. E para validar que o sistema tava correto a gente fazia teste com a ajuda de uma stakeholder que é estatística e cuidava do programa de bolsas, cuidava ou ainda cuida, então ela tinha acesso a esses dados do sistema legado e das planilhas que eles tinham lá anotados e a gente fazia o “de-para” pra ver se os dados estavam corretos.

D: Agora para fazer essas descobertas das informações para depois vocês irem lá validar. Você usava alguma ferramenta específica para a descoberta de informação ou você criou suas próprias ferramentas, você fazia à sua própria maneira?

F1: O que você chama de descoberta de informação?

D: Por exemplo assim, as vezes você tem que cruzar informações, por exemplo, algum conhecimento no sentido de bolsistas, área, tempo... E aí o cliente te pedia, numa situação hipotética, para extrair alguma informação disso. Então você tinha que ir lá no banco de dados que você tem pra puxar e relacionar, fazer a ligação desses dados. Você usava alguma ferramenta específica, por exemplo de inteligência, alguma coisa assim?

(gravação é interrompida)

A: A gente tava falando das ferramentas, se você utilizou alguma ferramenta específica pra poder [extrair as informações].

D: (Entrevistado repete a pergunta). Bom é o seguinte, a gente não tinha ... Bom, como vocês já devem saber, a gente tinha como base um sistema legado que já existia lá na FAPESB, que era uma base lá legada, que tinham algumas pessoas técnicas que conheciam a base, porque cuidavam do sistema legado. Então conheciam a base, sabiam onde estavam esses dados e pronto. O que é que aconteceu, a gente foi recolher os requisitos com os clientes e o que a gente perguntava era o seguinte, “Qual é a informação que vocês querem ter?”. Aí eles diziam: (entrevistado dá um exemplo de pedido dos clientes). [Após essa conversa] a gente vinha para essa base, estudava essa base, e aí eu pedia a ajuda do pessoal lá [da equipe de desenvolvimento], principalmente a outro desenvolvedor, que conhecia bastante a base. À partir daí a gente descobria onde estavam as tabelas, descobria como era essa relação. Não tinha ferramenta, era o conhecimento da gente mesmo. À partir daí eu fiz um script de migração. Então o Observatório FAPESB não usa a mesma base do sistema legado. Tanto é que ele precisa ser atualizado. (Entrevistado explica como irá ocorrer essa atualização). Então não tem ferramenta. A ferramenta que tinha, entre aspas, foi eu que fiz para importar. Que nada mais é que um script lá metido a besta.

D: É um script SQL ou linguagem?

F1: É um script SQL.

D: Beleza. Entendi então.

F1: Ele traz de uma base em um formato, formata e coloca na outra base. A base do Observatório é uma base normalizada. Como ia ser um sistema aberto, todo mundo ia poder acessar, a performance foi um ponto que foi bem chave. Então a gente não podia de hipótese nenhuma usar a base do sistema legado, porque era uma base muito ruim. Na verdade, se você for olhar criticamente, é um Data Warehouse (DW). O que foi feito não foi usado nenhum[a ferramenta]. Eu pensei no início usar o Pentaho, alguma coisa assim, mas eu achei que ia dar muito mais trabalho e que não ia atender o que a gente queria e achei que seria melhor fazer na mão mesmo. Mas a base hoje do Observatório, e se você for olhar criticamente o que é o Observatório hoje, ele é um sistema que exibe dados de uma DW.

_Fim da Seção 1

Seção 2_

Users Before Algorithms

D: Como você falou, sempre essa motivação partia de quando você pegava os requisitos com o cliente.

F1: Isso, a gente não fez nenhuma descoberta, assim, pode até ter feito, mas sem querer, mas propositalmente a gente não foi buscar nenhum dado. Assim, nenhum dado que não se conhecia. A gente foi buscar os dados que o cliente queria ver.

D: Você pegar esses requisitos com o cliente, tava dentro do Scrum ou tava, tipo, dentro da sua possibilidade, ou dentro da possibilidade dele, mas sem contar essa metodologia da equipe?

F1: Não. Perceba o que eu falei, o Scrum rolava para a equipe como todo. Eu não podia ser anti social e dizer que eu não ia participar. Mas o projeto Observatório era eu sozinho praticamente. Então não tinha Scrum no observatório. **Não tem porque ter Scrum se minha equipe sou só**

eu. Eu participava das reuniões de Scrum como todo mundo, eu dizia o que eu estava fazendo. Durante a reunião, muitas vezes, os outros caras pediam ajuda para resolver algum problema. Muitas vezes eu trazia os caras e falava, que precisava de [alguém] para ajudar para fazer alguma coisa. (Entrevistado explica os pontos em que se davam as reuniões de Scrum e de como pedia ajuda aos outros da equipe). Mas o projeto Observatório não tinha Scrum. Deu pra entender (trecho não entendido).

Plan for Scale

Daniel: Agora quando você, por exemplo, implementava o algoritmo que extraia informações lá, que ia fazer cruzamento dos dados, você fez ele de maneira escalável? Você implementou ele de uma forma [que] com poucas adaptações você conseguia descobrir mais informações?

F1: Não. Assim, respondendo a sua pergunta criticamente. O algoritmo não descobre informação sozinho. Ele é 100% engessado pra aquilo que [o cliente] queria ver. Em compensação, o simples fato de trazer os dados para essa outra base, dá a possibilidade de usar essa base para descobrir informação. A base do Observatório hoje, se você quiser fazer uma extração de dados, uma predição, por exemplo, você pode usar a base do Observatório pra fazer uma predição. O algoritmo que importa e que cria a base do Observatório não faz isso. Ele só traz o dado, ele é burro, traz e trata. Ele não faz mais nada. Mas a base se você quiser usar, com certeza você vai extrair coisa ali.

D: Então aí a escalabilidade você pensou na hora de fazer essa outra base de dados, não do sistema legado.

F1: Isso, a ideia era trazer os dados. Tanto é que os gráficos, se você entrar no site do Observatório, aqueles gráficos eles são gerados na hora que você solicita. Por exemplo, cada gráfico ali ele é realmente dinâmico. Eu não trago a informação daquele gráfico num pacote e tá lá fechado. Tem uma base e ele busca da base. Então se você quiser criar uma outra métrica

amanhã, é só você fazer uma consulta na base e mostrar o gráfico. O algoritmo que eu criei que importa, ele só importa. Como você vai explorar a informação que está ali é você quem decide.

Early Feedback

D: Agora você falou que tem a pessoa de estatística que faz a validação dos dados, não é?

F1: Isso.

D: Agora o quão frequente você ia lá pra poder validar uma informação. Tipo você esperava a sua funcionalidade, que você estava desenvolvendo, ficar pronta ou você fazia isso várias vezes, mesmo antes de você ter uma versão utilizável do software?

F1: Muito antes. Antes do software em si, do portal, do site do Observatório estar pronto, eu já fazia a validação de dados. Eu comecei a fazer a validação dos dados, quando eu só tinha a base. Digo mais, eu comecei a fazer a validação dos dados quando eu tinha só base e a base nem completa tava. Então, por exemplo, quando eu trazia os dados dos bolsistas, eu ia validar bolsistas. Eu ia validar grande área. Tinha que ir validando de pouquinho em pouquinho. Porque era muita coisa e deixar pra validar tudo no final ia ser inviável em questão de tempo. (Entrevistado explica o tempo que levava para executar o script). Então fazer [a validação] no final era inviável. O sistema não estaria no ar, com certeza.

D: E aí você tinha reuniões com essa pessoa, ou era assim mais informal, tinha liberdade [de fazer contato]?

F1: A gente tinha reuniões sempre que o presidente queria. Na verdade a gente tinha reuniões quase todas as sextas, a data marcada para o grupo do Observatório era às sextas feiras. Nem sempre acontecia de ter reunião toda sexta. Além disso, eu tinha total liberdade de ir na sala de

um dos [stakeholders] na hora que eu achasse necessário. Eu ligava para ela, via se podia ir na sala. Então isso nunca foi um entrave não.

A: O acesso de vocês com [os clientes] era praticamente livre?

(Entrevistado confirma a pergunta e explica algum nível de dificuldade com outro cliente. Porém confirma que o acesso ao cliente era frequente.)

Be Open-Minded

D: Agora, por exemplo, quando você pegava algum requisito lá com [os clientes] e ia fazer algum cruzamento de dados, alguma descoberta de dados. Vocês já entravam com predisposição para encontrar outras relações além do que ele tinha pedido?

F1: Não! Não tinha tempo pra isso, infelizmente. (Entrevistado explica razões para não ter tempo). Então a gente não tinha tempo de pensar, a gente tinha que fazer. Eu achava isso tudo muito legal. Quando eu fui pra lá eu tinha várias ideias de fazer várias paradas, de pesquisa, e buscar dado, de fazer inferência, fazer teste estatístico, as paradas lá, pra rodar, descobrir dado. Mas não tinha tempo. Tinha que mostrar o que [o cliente] queria. Então não fiz nada.

D: Por exemplo, você já deparou com situação assim que você tinha pensado em um jeito de cruzar informação, de pegar... e aí a estratégia ficou inviável e aí você teve que repensar tudo de novo, rever outra maneira? Ou isso não acontecia?

F1: Não aconteceu.

Avoid Bad Learning

D: Agora questão da validação dos dados. Essa parte de aplicação estatística... teve aplicação estatística, você sabe dizer?

F1: Fiz.

D: Você sabe dizer algum método, assim, em particular ou só com a pessoa de estatística mesmo?

F1: Quem fez a aplicação estatística fui eu. Ela não foi feita pra poder gerar os dados, nem buscar conhecimento nenhum. A aplicação de estatística foi feita única e exclusivamente pra mostrar o gráfico. (Entrevistado explica um exemplo em um gráfico). Quem fez isso fui eu.

D: Mas eu falo no sentido da parte mais da validação.

F1: Não, não teve isso não. Não teve e vou dizer que nem eu fiz e nem [a stakeholder estatística] não fez.

Live with the data you have

D: Você acabou dizendo que [o cliente] queria as coisas “pra ontem” né, então eram os dados que vocês tinham. Não dava pra esperar um *dataset* mais bonitinho, né?

F1: Pô, o *dataset* era bem bonitinho.

D: Não, eu falo assim o antigo, o legado.

F1: O legado era uma bosta. Muito ruim.

(risos)

(Entrevistado explica alguns problemas da base de dados do sistema legado. Também explica que algumas correções da base tiveram que ser feitas à mão.)

A: Mas vocês tinham que se virar mesmo com aqueles dados que vocês tinham ali, não tinha tempo pra buscar outras coisas...

F1: Tinha que “vestir a roupa de sapo”.

Broad skill set, big toolkit

D: Essa pergunta é só porque a gente tem que fazer mesmo, o que você falou já deu pra entender bem, mas é só porque tá no script mesmo. De tempos em tempos você teve que mudar algoritmo de descoberta, de extração de informação ou não?

(Entrevistado não entende a pergunta e pede para repetir)

(Entrevistador refaz a questão)

F1: Para descobrir novas coisas... Bom, vamos lá, se você pegar que eu já respondi que eu fiz essa migração de forma incremental, eu mudava toda hora. Eu comecei migrando os bolsistas, depois eu fui migrar bolsas, depois fui migrar pagamento de bolsa... Cada vez que eu fazia isso eu mexia no algoritmo. Então a resposta é sim. Agora se você tá perguntando no sentido de descobrir novas coisas, que estavam fora daquele escopo inicial, não. Eu fui fazendo a migração de forma incremental, mas pra descobrir as coisas que já estavam definidas no escopo.

D: E também, mudar o que já estava pronto, nem pensar né?!

F1: Não, se fosse necessário, se achasse erro, por exemplo, a gente mudava. Já teve erro que a gente mudou, mas não mudar pra trazer outras coisa. Mudar para corrigir erro. O escopo foi bem

fechado. Na verdade até aumentou depois, mas não reduziu. Então o que tava feito, foi feito e o que tava pra fazer depois a gente fez. Mas no geral...

(Entrevistador explica que irá fazer perguntas relacionadas aos requisitos não funcionais do sistema)

_Fim da Seção 2

Seção 3_

Performance

D: A questão do desempenho lá. Teve alguma prática que você adotou para que o sistema, pra ajudar que o sistema rodasse num tempo legal, em um tempo aceitável? Teve algo específico que você fez nesse sentido?

F1: Várias coisas. Com eu falei, o Observatório está lidando com uma base de dados gigante. Na verdade é uma base de dados pequena, mas com muitos dados (...) e ele era um sistema aberto, que todo mundo precisava usar. Então eu tinha dois problemas aí: eu tinha que conseguir fazer essas consultas muito rápido, eu tinha que conseguir um sistema que fosse estável e, na verdade teve um outro porém, o sistema tinha que ser (...) tinha que ser facilmente possível você expandir. A ideia era que crescesse muito. Quando eu saí de lá o que tinha, na verdade é o que tem até hoje, só mudou um gráfico, desde quando eu saí. Mas era que ele fosse expansível e que você pudesse colocar muitas outras informações que não estavam aí. Então o que é que eu tive que fazer, a começar da base, eu modeliei a base da melhor forma que eu pude e fui bastante criterioso. Essa base aí tá de forma normal, (entrevistado divaga sobre normalização de banco de dados). Eu tive que ser bastante chato com isso. Então começou pela base, que é uma base muito boa. Aí pronto, seleção de tecnologias. A gente foi selecionar tecnologias que comprovadamente

davam um melhor desempenho. Então ele usou a base de dados que nos mostrou ter um melhor desempenho, que era o Postgre[SQL]. Ele usou um framework que era voltado para desempenho, que foi o Spring. Então ele foi feito todo com Spring para que ele fosse mais rápido. Além disso, pro *front-end* a gente não utilizou nada mais do que Angular. Então tudo que a gente podia passar pra tirar a carga do servidor e passar para o usuário, a gente fez. Então hoje o Observatório praticamente é todo montado no [lado do] usuário, ele vai na base praticamente buscar os dados e mais nada, ele não faz mais nada no servidor. O servidor basicamente faz consulta, porque a gente jogou toda demanda pro Angular. Então isso faz com que o sistema, se vocês olharem o sistema é relativamente rápido. Relativamente não, ele é rápido. Além disso, pra parte de ser expansível, todas as consultas são feitas usando a técnica de reflexão. Pra ser expansível e [de] fácil manutenção. Então pra você fazer uma nova consulta hoje, a gente criou uma notação, você só precisa anotar a classe com o tipo da consulta que a gente criou, coloca a consulta e ele automaticamente gera o gráfico da forma que você quer. Hoje a manutenção e a expansão disso é muito rápido. Você só faz a sua consulta, se você fizer a sua consulta em SQL e jogar no Observatório ele gera o gráfico sozinho.

A: Agora teve algum requisito, sobre performance, o desempenho do sistema, teve algum requisito vindo do cliente, foi um pedido do cliente? Ou isso foi uma preocupação de vocês?

F1: Não, isso tudo foi do cliente. Tinha que ser rápido. O requisito não funcional que tinha era esse. Tem que ser rápido, não pode ser lento. Tem que ser muito rápido. Esse é o requisito que veio.

D: Agora teve alguma arquitetura, padrão de projeto, que você utilizou pra usar nesse sentido?

F1: Eu criei a arquitetura. Assim, ele usa basicamente MVC. A base é MVC, só que assim, se você olhar o sistema são só duas telas. Então eu não tenho muitos *controllers*, eu só tenho duas *views*, eu não tenho muita coisa pra fazer aquele MVC e tal. Essas poucas entidades que existem,

funcionam em MVC, mas eu criei uma arquitetura própria pra poder fazer as consultas de forma performática, que é usando a notação [do Java].

D: Agora teve algum teste específico para aferir desempenho?

F1: Não.

Security

D: Agora em questão da segurança do sistema. O sistema, ele não tem nada assim de logar, dessas coisas assim. Mas agora pra essa base de dados que você criou, como que é a segurança dela?

F1: No que tange à segurança dos dados foi uma preocupação que a gente teve pouco. Por um motivo específico, todos os dados que estão no Observatório são públicos. Todos os dados que estão no Observatório são para serem mostrados. Se por algum acaso “vaze” a gente não se importa tanto. Até porque não foi trazido desses dados nome, nem email, dos usuários. (Entrevistado explica que não tem “perigo” de vazarem informações pessoais).

D: Agora no sentido de alguém poder adulterar os dados, ele entrar no sistema de alguma forma entrar no banco e começar a fazer algumas brincadeiras lá?

F1: Bom, se ele conseguir fazer isso, ele vai fazer via infra-estrutura. E aí a gente não tem como [evitar], não competia à gente defender a infra-estrutura. Porque dentro do sistema ele não vai conseguir fazer. Como você disse o sistema não tem área logada. O sistema não faz insert. Ele não tem nem como fazer um SQL Injection. O sistema não faz insert, não tem área logada, o sistema inclusive, na verdade, a gente até identificou, teve uma hora que eu peguei uma falha de segurança. As consultas elas estavam na tela, como eu falei a gente trazia tudo para a tela. As consultas estavam na tela e iam da tela pro banco, e isso podia fazer [SQL] Injection. (...) Aí eu

troquei as consultas por ids. (Entrevistado explica que no código da tela do sistema tem os ids das consultas). Agora se ele alterar o *id* não vai ter gráfico ou vai trazer o gráfico errado, então a gente não tava preocupado.

A: Mas isso não vai chegar à um nível do banco né?!

F1: Vai fazer uma consulta diferente.

A: Mas eu digo no intuito de você modificar os dados, ou dar um *drop*, alguma coisa assim.

F1: Não. Impossível, nunca, não rola. Não tem como. Ele só pode fazer isso hoje via infra estrutura. (Entrevistado aconselha aos entrevistadores à olhar o código do sistema).

(Divagações)

Availability

D: Agora você falou que o pessoal lá da [departamento de] infra [estrutura] é que fica responsável pela questão da segurança. E disponibilidade também né?

F1: Sim, isso aí é com [o departamento de] infra [estrutura]. Aí a gente não se metia não. O que a gente conseguiu fazer foi, a gente tinha um servidor só pro observatório. (Entrevistado fala das condições de máquinas para o servidor).

D: E você fazia alguma coisa pra ficar [disponível] o maior tempo possível?

(Entrevistado faz uma brincadeira com a pergunta e diz que não tinha o que fazer)

Reliability

D: Alguma coisa que você fez no sistema pra evitar dar *crash*, essas coisas assim?

F1: Bom, eu tentei fazer as coisas direito. Se você disser que eu utilizei alguma técnica pra poder evitar, pra pegar controle de exception? Isso eu não fiz não. Agora a gente não tinha tempo de fazer nada que fosse mais rebuscado. A gente não tinha tempo.

(Divagações)

Usability

D: Agora a questão de usabilidade do sistema, você teve que desenvolver interface?

F1: A gente tinha um cara que fazia isso, um cara muito bom inclusive. (...) Ele que fazia essa parte de interface, a gente não fazia nada. Ele dava um pdf da interface e eu transcrevia isso pra HTML e CSS.

D: Teve tempo de fazer algum teste com usuário pra saber se tava conseguindo mexer?

F1: Não!

Scalability

D: Você falou um pouco da expansão do sistema. Você teve que fazer alguma coisa no... você falou que as buscas você coloca lá um SQL e ele já gera um gráfico pra você. E você falou que você desenvolveu uma arquitetura própria, mas você fez alguma coisa em específico pra você poder escalar o sistema na parte do Spring.

F1: Foi a arquitetura própria.

A: A forma que você arquitetou o sistema...

F1: É, isso tá no Spring. Essa arquitetura que a gente fez tá no Spring. Não é no Angular não. No Angular o que a gente fez foi pra tomar cuidado pra que tudo que pudesse ser processado no cliente, fosse processado no cliente. Daí a única coisa que o Angular faz. A arquitetura foi feita toda no Spring mesmo.

A: Agora essa questão da escalabilidade foi feita basicamente, como você falou, no Spring e no sistema né. Já no conhecimento de novas informações não. Como você falou que sua consulta é bem [engessada].

F1: Não, não é. Bem, assim, pro conhecimento de novas informações ele é escalável no sentido que você pode usar a mesma base e o mesmo sistema pra, se amanhã eu quiser gerar um gráfico diferente, eu faço uma consulta na base que me dê essa informação, eu jogo pro sistema e o sistema vai me gerar o gráfico. O sistema faz isso sozinho? Não, mas se você mostrar, se você fizer a consulta, colocar na anotação, dizer qual é o tipo de gráfico, coloca a coisa na tela, ele mostra. Sem precisar mexer em código. Você só mexe na sua consulta, mais nada.

D: Agora na questão da arquitetura, só em você utilizar MVC você já gera um ganho de escalabilidade bom. Mas teve alguma outra coisa, algum “pulo do gato” que você fez pra que o cara conseguisse crescer o sistema?

F1: De novo, a arquitetura. A arquitetura é o grande “pulo do gato”. Porque ele que consegue fazer isso aí sem fazer quase nada.

D: Sim, mas o que na arquitetura garante isso?

F1: Por exemplo, a gente usa anotação. Então todos os gráficos são feitos com anotações nas classes que lhes convém. (Entrevistado dá um exemplo do sistema)

D: Essa anotação é do próprio Java?

F1: É do próprio Java, é uma Api do Java. Java Annotation.

D: Agora alguém tinha pedido isso pra você ou você fez [por você mesmo]?

F1: Não, decisão minha.

D: E você desenvolvia a arquitetura enquanto programava ou você fez algum gráfico de alto nível pra você entender mesmo :

F1: Não, assim, eu fiz no papel, risquei, mas assim documentado isso não tinha não.

A: Enquanto isso, me diz quais foram as ferramentas que vocês utilizaram lá para o sistema?

(Entrevistado não entende a questão e entrevistador repete)

F1: Java, Angular e SQL. Não tinha nada diferente disso aí não.

(Divagações)

Interoperability

D: Agora o sistema do Observatório tem que se comunicar com outro sistema? Você falou da base legada, que é feita em nível de importação de dados, mas tem algum outro sistema que ele precise conversar?

F1: Não. Hoje, não. A ideia é que vá ter. (Entrevistado dá um exemplo de comunicação com outro sistema).

Adaptability

D: Vê a questão se o sistema está funcionando em vários navegadores, plataforma móvel. Isso daí foi um requisito pra vocês?

A: Vocês se preocuparam com alguma versão de browser? Ou tinha que rodar em celular, em tablet?

F1: A gente se preocupou em rodar em celular.

A: Mas isso foi pedido pelo cliente ou foi uma preocupação de vocês?

F1: Foi pedido.

A: E teve alguma especificação?

F1: Não, longe disso.

D: Alguma ação específica na hora de montar o sistema, pra garantir essa adaptabilidade?

F1: Bootstrap!

A: E foi feito algum tipo de teste sistemático pra poder garantir isso?

F1: Não.

_Fim da Seção 3

(Finalização da entrevista).