**Plano de *onboarding* – Desenvolvimento**

Este plano tem por objetivo estabelecer a trilha de desenvolvimento inicial para a pessoa ingressante no time de desenvolvimento da Cappta. Ele deve ser abrangente o suficiente para cobrir as *squads* do transacional-físico e pagamentos, níveis de senioridade e especificidade pessoal de desenvolvimento (*backend/frontend*).

O *onboarding* das pessoas desenvolvedoras consiste em um período de duas semanas, ocorrido imediatamente após o *onboarding* Stone/Cappta. Este período pode ser estendido ou contraído em até uma semana, caso haja necessidade. Nele, é esperado que a pessoa se dedique à um intenso estudo teórico sobre os tópicos abaixo apontados, que se dividem essencialmente quanto à senioridade da pessoa.

Ao fim desse período, a pessoa deve produzir uma apresentação para a *squad* em que está inserida sobre os temas estudados. Idealmente, ela deve ocorrer na sexta-feira da segunda semana de estudos, entretanto podem haver pequenas variações.

A ideia da estrutura de tópicos da apresentação é dar liberdade para que as pessoas ingressantes, sobretudo em perfis júnior, possam alinhar este período de estudo com seus objetivos de desenvolvimento pessoal.

1. **Objetivos específicos do *onboarding***

* Integrar novos desenvolvedores(as) às *squads* do transacional-físico e pagamentos:

Existe uma forte Cultura orientada ao time dentro da Tecnologia da Cappta. Neste sentido, apoiamos a inserção mais rápida possível dos profissionais ingressantes em suas *squads*. Assim, dentro do tempo de estudo, é recomendado que os ingressantes permaneçam nas salas dos times, participando como ouvintes das discussões inerente ao negócio.

* Fazer com que a pessoa ingressante compreenda apropriadamente as expectativas da liderança para ela:

Nossa Cultura também é fortemente orientada à transparência. Neste contexto, é interessante que cada pessoa conheça as expectativas da liderança da empresa e também seja transparente quanto à suas expectativas de desenvolvimento pessoal. Assim, este momento de *onboarding* também pode e deve ser utilizado para este alinhamento de expectativas.

* (Júnior) Garantir o aprendizado e fixação de conceitos fundamentais de programação e boas práticas de desenvolvimento

Outro fortíssimo aspecto da nossa Cultura é apoiar o desenvolvimento pessoal e profissional de nossos Devs. Para tal, acreditamos que uma base sólida é essencial para o desenvolvimento futuro de profissionais em TI e desenvolvimento. Com isso, sobretudo para os perfis Júnior (embora não restrito a eles), é encorajado o estudo com forte abordagem teórica, prévia às atividades de desenvolvimento.

1. **Conteúdo da apresentação e *RoadMaps***

O conteúdo de apresentação é variado, sobretudo segundo o nível de senioridade profissional. Ele consiste em tópicos estruturais (obrigatórios) e adicionais (opcionais).

Os tópicos estruturais devem ser abordados em todas as apresentações, enquanto os opcionais devem estar alinhados com as expectativas de desenvolvimento pessoal da pessoa ingressante. Entretanto, minimamente um dos tópicos adicionais é necessário na apresentação.

* 1. Dev. Júnior

Devs. Júniors são encorajados a estudar os tópicos abaixo com foco em uma abordagem mais teórica. Isso significa que o uso de código ou pseudocódigo durante a apresentação não é obrigatório, muito embora o uso de código ajuda a esclarecer alguns pontos.

* + - Tópicos estruturais:
      * Programação Orientada à Objetos (Conceitos)
      * Conceitos SOLID
      * Protocolo HTTP (Métodos)
      * Conceitos HTTP e REST
    - Tópicos adicionais sugeridos:
      * Programação Funcional
      * *Clean Code*
      * Protocolo TCP/IP
      * DNS
      * APIs
      * Algoritmos
      * Funcionamento do Sistema Operacional
      * Protocolo HTTP (Respostas)

É importante notar que, embora estejamos sugerindo alguns tópicos estruturais, a lista não necessariamente precisa ficar limitada a eles. Uma vez que a pessoa ingressante julgue apropriado, ela pode adicionar temas de tecnologia não listados à sua apresentação.

Além disso, é importante notar que, embora alguns conceitos não sejam necessários na apresentação, eles são pré-requisitos para o entendimento de outros, conforme evidenciado no diagrama da Fig.1.

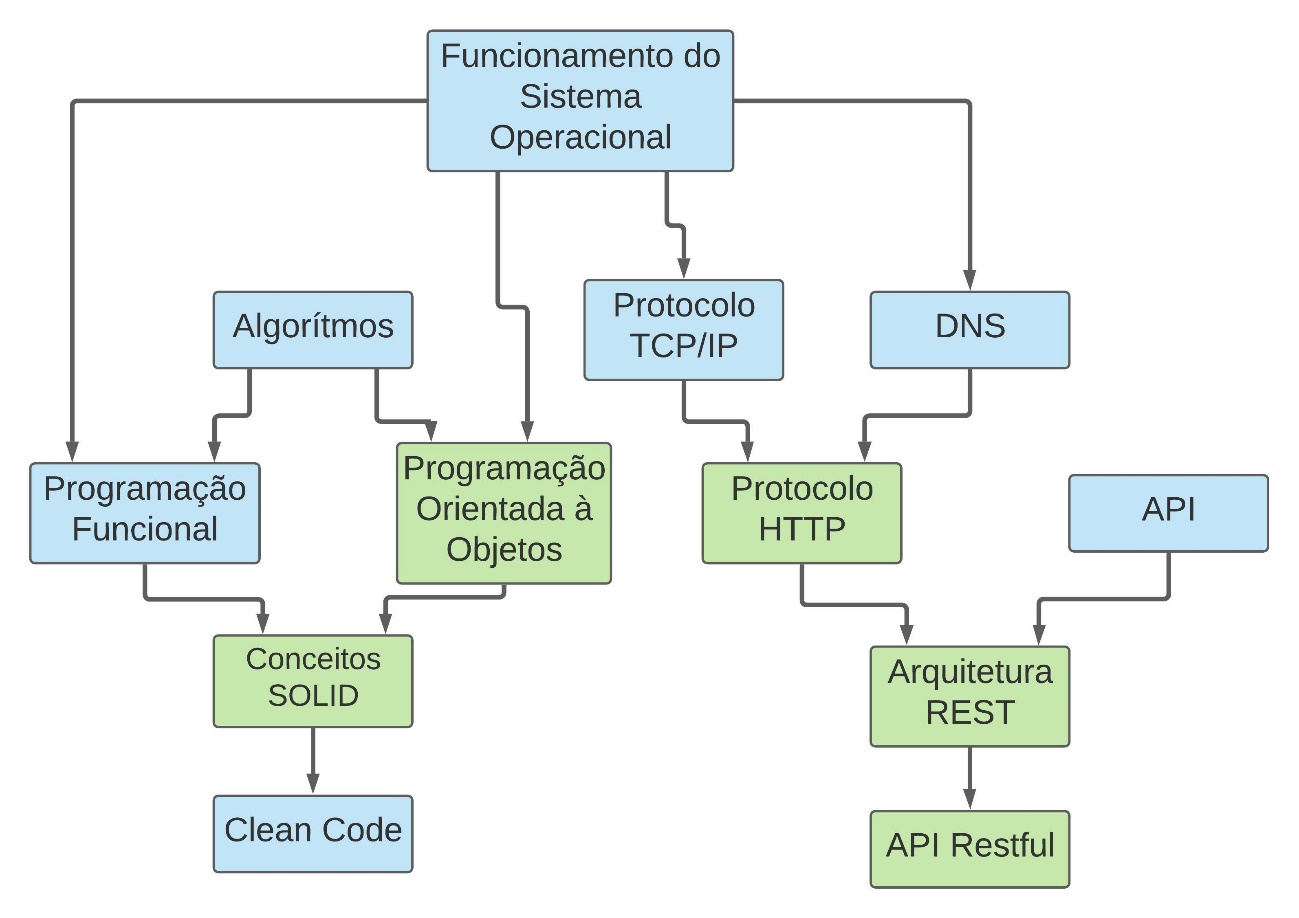


Figura 1 Diagrama de desenvolvimento de apresentação para ingressante Júnior.

* 1. Dev. Pleno

Diferentemente dos devs. Júniors, os devs. Plenos são encorajados a estudar os tópicos abaixo com foco em uma abordagem mais prática, usando, sempre que possível, trechos de código (em uma linguagem de sua escolha) para elucidar os pontos da sua apresentação.

* + - Tópicos estruturais:
      * Programação Orientada à Objetos (Focado em uma linguagem)
      * Protocolo HTTP (Respostas)
      * API *Restful*
      * Integração entre APIs
    - Tópicos adicionais sugeridos:
      * Programação Funcional
      * Validação
      * Boas práticas de programação
      * Funcionamento do Sistema Operacional
      * Criptografia e HTTPS

O *RoadMap* de conceitos para desenvolvedor pleno é apresentado na Fig. 2.

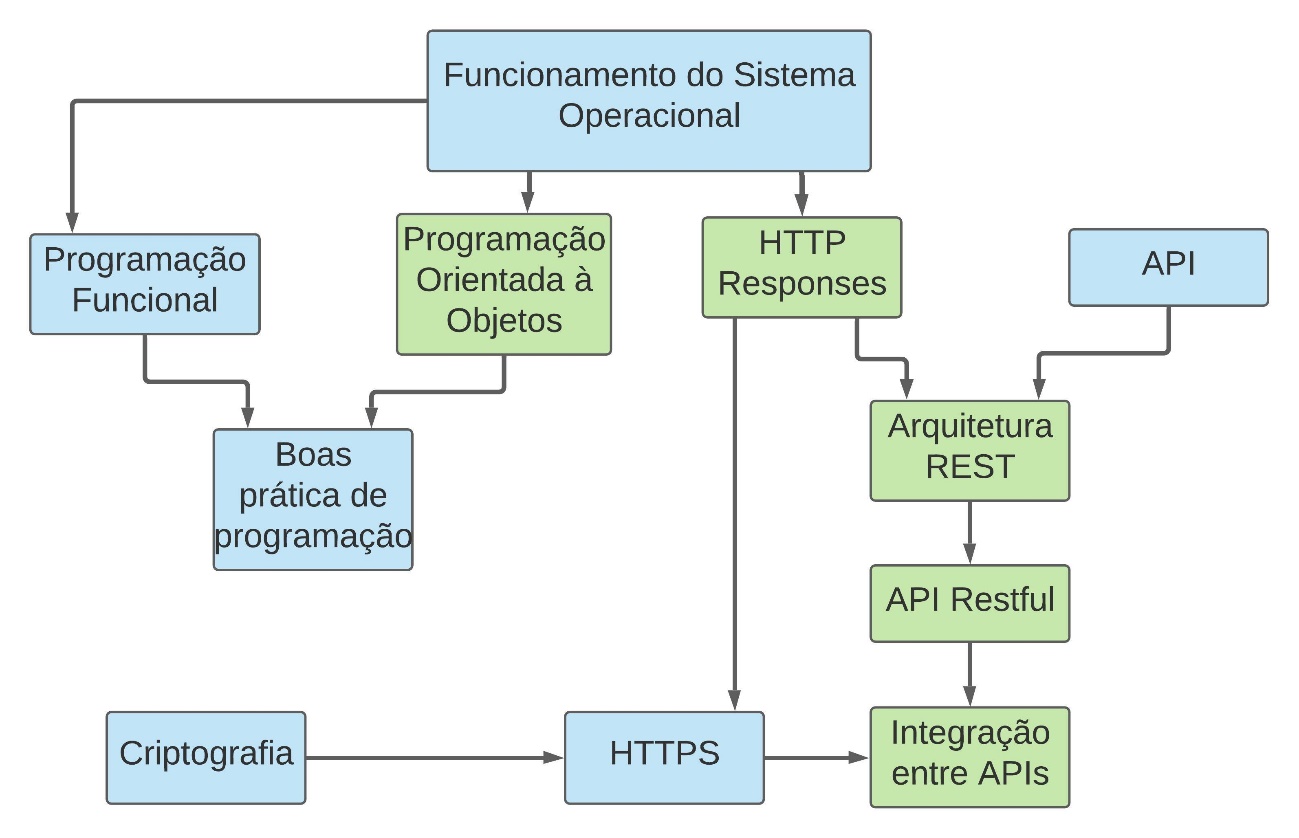


Figura 2 Diagrama de desenvolvimento de apresentação para Pleno.

* 1. Dev. Sênior

Devs. Sêniors devem atuar não apenas no desenvolvimento de software, mas também na gestão do Ecossistema da Cappta como um todo. Pare estes, esperamos uma forte integração entre todas as *squads*, além dos times de Infraestrutura. Além disso, eles precisam de uma rápida inserção nas regras de negócio da Cappta. Deste modo, encorajamos seu estudo em âmbitos mais gerais, como segue:

* + - Tópicos estruturais:
      * Integração entre APIs e ecossistemas
      * Arquitetura de APIs e *Design Patterns*
      * Modelo de Negócio da Cappta
    - Tópicos adicionais sugeridos:
      * Criptografia e HTTPS
      * Programação Orientada à Objetos ou Funcional (Focado em uma linguagem)
      * Aplicação de boas práticas de programação em projetos.
      * Arquitetura de Ecossistemas

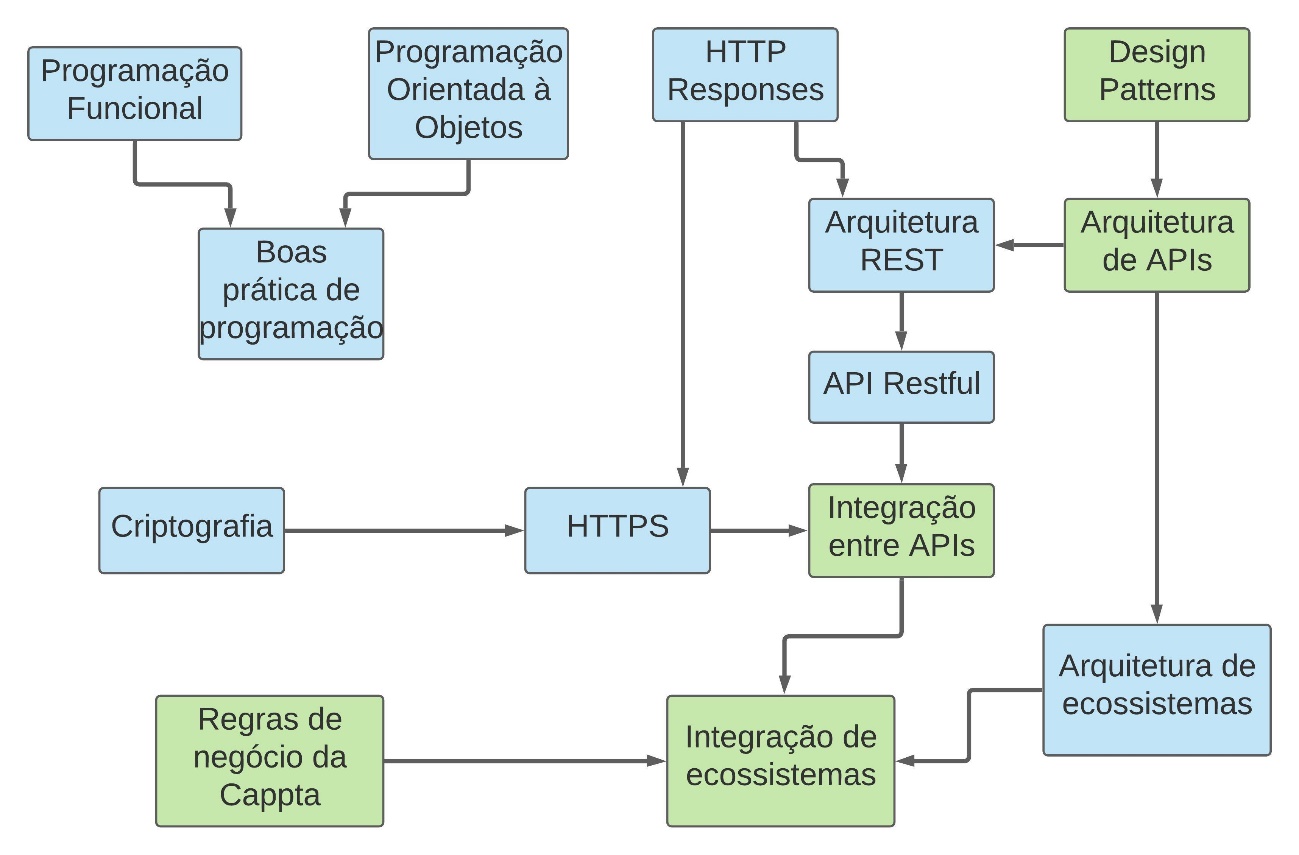


Figura 3 Diagrama de desenvolvimento de apresentação para Sênior.

O *RoadMap* de conceitos para desenvolvedor pleno é apresentado na Fig. 3.

1. **Detalhes da apresentação**

Definidos os pontos a serem apresentados, a pessoa ingressante deverá estruturar sua apresentação para um tempo entre 30 e 40 minutos.

O público da apresentação será a *squad* com a qual a pessoa estará envolvida e, eventualmente, seus tutores e/ou pessoas de outras *squads*. Embora seja recomendado que o público retenha suas perguntas para o fim da apresentação, comentários durante a apresentação são permitidos, desde de que sejam breves e relevantes. O interlocutor nesse caso poderá se estender além do tempo.

A Cappta possui um *template* para apresentação deste gênero, que será disponibilizado no mesmo repositório deste texto. Contudo, embora recomendado, seu uso não é obrigatório.

1. **Tutores**

Tutores são membros do time de Tecnologia da Cappta que estarão à disposição das pessoas ingressantes para esclarecer dúvidas que possam eventualmente surgir durante o período de *onboarding*. Estes se comprometem a oferecer o melhor do seu conhecimento para tirar as eventuais dúvidas dos ingressantes.

É recomendado, entretanto, que as pessoas ingressantes desenvolvam autonomia suficiente para esclarecer os pontos por si só e recorram aos tutores apenas em último caso.

Também é necessário explicitar que os tutores são funcionários da Cappta com suas próprias atribuições e desafios, sendo assim, pode ser que algum tutor específico não esteja disponível em algum período específico.

Neste contexto, fica especificado um limite de tempo para consulta diária do tutor de 10 minutos.

Ainda qualquer membro do time da Cappta pode ser tutor de qualquer ingressantes, desde que estejam satisfeitas duas condições:

* O tutor possui um nível de senioridade maior ou igual ao ingressante
* No caso de mesmo nível de senioridade, o tutor deve possuir ao menos 3 meses de integração com a *squad*.

No caso de indisponibilidade de um tutor específico, outro tutor pode ser consultado, mesmo num assunto no qual não esteja listado na lista abaixo:

**Lista de Tutores**

Programações Orientada à Objetos:

Programação Funcional:

SOLID:

*Clean Code*:

TCP/IP, DNS e HTTP:

API/REST:

Integrações:

Regras de Negócio da Cappta:

**Material de apoio**

Esta é parte do material indicado para estudo (mais adequado para o nível júnior). Muito embora há diversos materiais disponíveis sobre este assunto e a pessoa ingressante pode consultar a vontade.

* Programações Orientada à Objetos:
  + <https://www.youtube.com/watch?v=KlIL63MeyMY&list=PL6qsRzBhn4BlSiDHoGWLj6Op4Ika8zjIC>
  + <https://www.youtube.com/watch?v=QY0Kdg83orY&t=5s>
  + <https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>
  + https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264
* Programação Funcional:
  + <https://www.alura.com.br/artigos/programacao-funcional-o-que-e?gclid=CjwKCAiAgc-ABhA7EiwAjev-jzSPX3V76NGxmlBel73rlE6XdcplNxdMj5HqBC_b7tAusKtQuEMVXRoCfkQQAvD_BwE>
  + <https://www.youtube.com/watch?v=BxbHGPivjdc>
  + https://www.youtube.com/watch?v=BWFpUHPqh1g
* SOLID:
  + <https://www.youtube.com/watch?v=6SfrO3D4dHM>
  + <https://www.youtube.com/watch?v=mkx0CdWiPRA>
  + https://medium.com/desenvolvendo-com-paixao/o-que-%C3%A9-solid-o-guia-completo-para-voc%C3%AA-entender-os-5-princ%C3%ADpios-da-poo-2b937b3fc530
* *Clean Code*:
  + *Livro*: Clean Code – Robert Martin (2008)
  + <https://medium.com/desenvolvendo-com-paixao/1-clean-code-o-que-%C3%A9-porque-usar-1e4f9f4454c6>
  + <https://www.hostgator.com.br/blog/clean-code-o-que-e/>
  + https://www.youtube.com/watch?v=ln6t3uyTveQ
* TCP/IP, DNS e HTTP:
  + <https://www.youtube.com/watch?v=PpsEaqJV_A0>
  + <https://www.youtube.com/watch?v=vkwbHRpq09U>
  + <https://www.tecmundo.com.br/o-que-e/780-o-que-e-tcp-ip-.htm>
  + <https://store.hp.com/us/en/tech-takes/how-does-the-internet-work#:~:text=The%20internet%20is%20a%20worldwide,(TCP)%20%5B5%5D>.
  + <https://www.tecmundo.com.br/o-que-e/829-o-que-e-dns-.htm#:~:text=O%20DNS%2C%20do%20ingl%C3%AAs%20Domain,monte%20de%20n%C3%BAmeros%20e%20pontos>.
  + https://www.youtube.com/watch?v=hwttZtWkXTk
* API/REST:
  + <https://www.youtube.com/watch?v=S7MduKwvVGk>
  + <https://becode.com.br/o-que-e-api-rest-e-restful/>
  + https://dzone.com/refcardz/rest-foundations-restful?chapter=1