

Carta de boas-vindas,

### **Contextualização:**

A FoodDelivery é uma empresa de tecnologia sediada em Salvador - BA desde 2025. Nós passamos por algumas transformações e, como você pode ver, estamos em um novo momento. Nós até mudamos de sede! Mas o nosso desafio continua o mesmo: conectar restaurantes, bares, projetos gastronômicos, e agora mercados e farmácias, aos consumidores em qualquer lugar com apenas um clique na palma de sua mão. Estamos desenvolvendo novas funcionalidades e trazendo aos nossos clientes: empreendedores e consumidores finais, o melhor conforto e agilidade possível. E nada como automações nos serviços para dar aquela experiência premium, não é?

Neste ano, a foodDelivery volta com o seu programa de trainee mais forte e robusto que nunca. Parabéns por terem sido aprovados! Quais de vocês serão as próximas sandies ou juniors das nossas equipes? Nós estamos ansiosos para desenvolver os seus talentos. Vocês vão ser integrados aos nossos times da área de Dados, Desenvolvimento Mobile Nativo e Desenvolvimento Web e farão parte do nosso projeto mais ambicioso: o Lu delivery. Vamos nessa?

Como trainees na área de Desenvolvimento Mobile Nativo, vocês enfrentarão quatro desafios crescentes ao longo do semestre. Acreditamos que o desenvolvimento de software é uma atividade social, por isso, vocês trabalharão em pequenos grupos, com metas e entregas claras, e serão avaliados pela qualidade do que entregam, pelo esforço nas tarefas, pela organização e clareza.

Vocês desenvolverão um protótipo de aplicativo que fará uso de um backend desenvolvido por outros times. As etapas incluem: 1) Implementação de interfaces de usuário; 2) Manipulação de dados locais e remotos; 3) Integração com APIs; e 4) Otimização de desempenho e gerenciamento de estados. Os desafios não são simples, mas serão desenvolvidos em grupo, e vocês verão que a colaboração e as competências sociais são fundamentais para o sucesso. Esperamos que se sintam inspirados e motivados neste processo formativo.

## Manual de Conduta:

Nos últimos anos, o uso de IA generativas por parte dos nossos candidatos nos trouxe uma reflexão importante: estamos preocupados apenas em fazer entregas de código ou estamos preocupados em fazer entrega de valor? Nós da foodDelivery não temos problema algum com o uso de IAs Generativas, inclusive achamos essas ferramentas interessantes no apoio do processo formativo de vocês, assim como um contato positivo com novas culturas empresariais. Entretanto, isso exige mais de vocês, que não podem ser apenas escritores de código, pois isso, como vocês já repararam, as IAs generativas sabem fazer tendo um prompt bem escrito. O nosso objetivo nesse processo é saber se os nossos novos talentos sabem resolver problemas. Se eles compreendem o que estão fazendo e o porquê estão fazendo. Afinal, **se você não sabe o problema que quer resolver, nenhum prompt de IA vai resolvê-lo para você.** Desta forma, qual o seu valor como profissional? É isso que vamos explorar neste semestre.

Nesta primeira etapa, temos alguns objetivos claros para serem desenvolvidos: uma revisão de estatística e uma introdução à exploração de dados. Além disso, queremos acostumar a vocês a usarem repositórios compartilhados, utilizando, inicialmente, o modelo de desenvolvimento centralizado (ou [Centralized Git Workflow](#)). Falando em Git, vocês já estão em um momento da sua formação em que ferramentas de versionamento de código devem ser consideradas competências básicas. Contudo, se você ainda não se sente confortável com o uso da ferramenta Git, você pode estudar esse curso: [Curso em Vídeo: Git](#) ou esse outro curso [Minicurso de Git: Código Fonte](#). Nós recomendamos fortemente que vocês comecem a aprender a ferramenta o quanto antes e estude ambos, se possível for. **Nós vamos considerar que a aprendizagem dessa ferramenta está sendo consolidada nessa primeira entrega, mas nas demais, vamos considerar que essas pequenas coisas já foram superadas.**

Além da ferramenta de versionamento, vamos avaliar em vocês a sua proficiência na linguagem de programação **Kotlin**. Queremos avaliar o seu conhecimento em estrutura de dados e a aplicação delas na resolução de problemas. Se você ainda não se sente confortável com a linguagem Kotlin, recomendamos esse curso: [Curso de Kotlin: Rapadura Dev](#). Você também terá suporte do seu padrinho ao longo do semestre, mas estudar e, principalmente, **PRATICAR** é fundamental. Não adianta apenas conhecer, é preciso **PRATICAR**.

**Atenção:** Não pegue atalhos nesse caminho! Por favor, não faça as IAs generativas executar uma tarefa que é sua. Por um segundo, pense: Se ela faz o que você deveria fazer, por que alguém te contrataria e não pagaria uma assinatura dela? Se ela não fica doente, não atrasa, não tem problemas pessoais e é infinitamente mais barata do que você, por que alguém te contrataria para você escrever um prompt para ela? Você, talvez, ainda não tenha a experiência de carreira suficiente para ter o seu valor provado. Não se

**iluda! Não conte mentiras para você! Seja honesto nesse processo! Aprenda, pois para quem está entrando no mercado, o seu valor vai ser fazer o que uma IA generativa não faz.**

Então, nesse momento, você já percebeu que temos como base três coisas: (1) saber como fazer é tão importante quanto fazer; (2) saber usar ferramenta de versionamento de código é peça fundamental no desenvolvimento das atividades desse semestre e (3) aprender Kotlin não é uma opção, é obrigatório! Mas como você será avaliado sobre esse pontos? Vamos ver nas próximas seções.

## Produto

Quando pensamos em trainees na área de desenvolvimento mobile nativo na FoodDelivery, nós pensamos em duas características claras: uma pessoa boa em lógica de programação e em Kotlin. Como a nossa parceria com a Unex não é recente, nós sabemos que vocês já passaram pelas aventuras de Algoritmos, Estrutura de Dados e talvez até mesmo um pouco de Desenvolvimento de API. Mas você sabe por que esses conteúdos são tão importantes? Porque com eles, vocês serão capazes de explorar uma série de soluções computacionais para construir aplicativos robustos e eficientes em Android.

Nas próximas etapas do seu trabalho como trainee, você vai trabalhar com a implementação de interfaces de usuário, manipulação de dados locais e remotos, e integração com APIs. Deste modo, é importante que estejamos seguros que vocês sabem o que é um ciclo de vida de uma Activity/Fragment, como gerenciar estados de tela, como trabalhar com threads e concorrência para operações assíncronas, e como otimizar o desempenho do aplicativo. Aí você fala: eu vim ser trainee da área mobile ou de engenharia de software? Eu lhe respondo: você vai aprender que os dois estão bem correlacionados. Não tem como construir um aplicativo sem conhecer sobre arquitetura de componentes Android (Activities, Fragments, Services). Como vamos falar de persistência de dados sem falarmos de bancos de dados locais (Room) ou consumo de APIs REST/GraphQL? Como vamos falar de usabilidade sem falarmos de padrões de design de interface e experiência do usuário? Todo esse conhecimento de desenvolvimento mobile nativo é fundamental.

Como falamos anteriormente, os projetos deste programa serão implementados em grupos. As equipes devem ter **obrigatoriamente** 3 membros. Não são 2 membros, nem tão pouco 4 membros, mas 3. Exceções devem ser comunicadas ao tutor e registradas e autorizadas por ele. Cada time vai ter um nome conforme planilha: [Nome dos Times e Seus Membros](#). Os nomes dos times foram sorteados no nosso primeiro encontro presencial.

Neste primeiro desafio, vamos implementar um sistema de pedidos de comida para um restaurante que tem como finalidade gerenciar pedidos e itens de um restaurante, operando exclusivamente via console (sem interface gráfica) e utilizando a linguagem Kotlin. É crucial que a implementação não faça uso de funções ou métodos externos ao fluxo principal do programa, seguindo uma lógica linear. Os dados deverão ser mantidos apenas em memória, ou seja, não haverá persistência em banco de dados.

## 1. Estrutura do Menu Principal:

O sistema deve apresentar um menu interativo com as seguintes opções para o usuário:

- **1. Cadastrar Item:** Permite adicionar novos produtos ao menu do restaurante.
- **2. Atualizar Item:** Possibilita modificar informações de itens já cadastrados.
- **3. Criar Pedido:** Inicia o processo de montagem de um novo pedido.
- **4. Atualizar Pedido:** Permite alterar o status de um pedido existente.
- **5. Consultar Pedidos:** Exibe informações sobre os pedidos, com a opção de filtrar por status.

## 2. Gerenciamento de Itens:

Cada item cadastrado no menu do restaurante deve possuir as seguintes características:

- **Nome:** Identificação do produto (Ex: "Hambúrguer Clássico", "Pizza Calabresa").
- **Descrição:** Detalhes sobre o item (Ex: "Pão, carne, queijo, alface, tomate", "Molho de tomate, mussarela, calabresa").
- **Preço:** Valor monetário do item.
- **Quantidade em Estoque:** Número de unidades disponíveis do item.
- **Código:** Um identificador único para o item, gerado automaticamente pelo sistema no momento do cadastro.

## 3. Criação e Gestão de Pedidos:

Ao criar um novo pedido, algumas regras e funcionalidades devem ser observadas:

- **Mínimo de Itens:** Um pedido deve conter, obrigatoriamente, no mínimo um item adicionado.
- **Cupons de Desconto:** Antes da finalização do pedido, o sistema deve oferecer a opção de adicionar um cupom de desconto. A abstração do cálculo do desconto pode ser simplificada (ex: um percentual fixo).
- **Pagamento:** A etapa de efetivação do pagamento será abstraída; ao ser criado, o pedido é automaticamente considerado como pago.  
**Status Inicial:** Todo novo pedido criado terá o status inicial de **ACEITO**.
- **Fluxo de Status:** Um pedido pode transitar pelos seguintes status:
  - **ACEITO**
  - **FAZENDO**
  - **FEITO**
  - **ESPERANDO ENTREGADOR**
  - **SAIU PARA ENTREGA**
  - **ENTREGUE**

#### 4. Consulta de Pedidos:

Ao selecionar a opção de consultar pedidos no menu principal, o sistema deverá apresentar um submenu. Este submenu permitirá ao usuário filtrar a visualização dos pedidos com base em seus diferentes status.

O projeto deve ser versionado e entregue no repositório de cada equipe. O nome do repositório deve seguir o padrão: tia-lu-dev-mobile-<nome\_do\_time>. Atualize a tabela de [Equipes](#) com o link do seu repositório. Não esqueça de adicionar o tutor ao repositório. Você pode enviar o convite para o nickname: lasilva.

Vocês também devem elaborar uma apresentação no [Modelo de Apresentação](#). As apresentações serão realizadas apenas para o tutor. A apresentação também precisa ser versionada no repositório com o seguinte padrão de nome: apresentacao\_<nome\_da\_equipe>.pdf. Além do versionamento, a apresentação deve ser entregue no blackboard no formulário: [Formulário de Submissão de Relatório](#).

## **Avaliação:**

Neste semestre, vamos introduzir algumas nomenclaturas que serão importantes ao longo do processo avaliativo e formativo de vocês. Vocês serão solicitados, a cada período de tempo, aproximadamente entre 3 e 4 semanas, uma entrega de valor. Nós chamamos essas solicitações de **demandas**. Vocês atuarão em equipe para entregar o resultado dessas **demandas** que vem da área de produto. Quem especifica o produto é conhecido, em ambientes que utilizam metodologias ágeis, de **Product Owner**, ou simplesmente, **PO (lê-se Pi-Ou)**. A entrega de valor realizada ao final de cada demanda vai ser chamada de **Entregável**. Em geral, os **entregáveis** são arquivos de **código fonte** que realizam uma atividade. Nós vamos usar siglas para nos referirmos aos entregáveis, por exemplo: **Entregável 1 - E1**, **Entregável 2 - E2**, e assim por diante.

A cada entrega, vocês passarão por uma **avaliação oral**. Nessa avaliação, vocês terão que apresentar o contexto do que foi desenvolvido e serão questionados pelo tutor. A avaliação oral forma um **peso** que vai ser utilizado para ponderar o valor obtido no entregável. Deste modo, o valor final do entregável será:

$$VF\_E = (AO * E) / 3$$

onde:

**VF\_E** é o Valor Final do Entregável

**AO** - Apresentação Oral

**E** - Valor do Entregável

## **Observações importantes:**

1. Apresentações fora do padrão estabelecido serão pontuadas com zero.
2. As apresentações devem ser realizadas no máximo de 7 minutos.
3. Apenas um membro da equipe vai realizar a apresentação. O membro da equipe será escolhido pelo tutor no dia da apresentação.
4. Se a equipe não apresentar, o valor da apresentação será zero.
5. Se um membro da equipe faltar a apresentação no dia ficará com 30% da nota de apresentação da equipe.
6. Apresentações não versionadas serão pontuadas com zero.

**Pontuação:**

<b>Atividade</b>	<b>Peso</b>
<b>Apresentação</b>	3

<b>Atividade</b>	<b>Nota</b>
<b>Entregável 1</b>	3 pontos

**Prazos:**

<b>Atividade</b>	<b>Data</b>
<b>OAT1.1 - Entregável</b>	23/08/2025
<b>Apresentação</b>	até 25/09/2025