

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO
CAMPUS SÃO PAULO
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Alkindar José Ferraz Rodrigues
Carolina de Moraes Josephik
Fabio Mendes Torres
Gabriely de Jesus Santos Bicigo
Leonardo Naoki Narita
Mariana da Silva Zangrossi

Lixt

Desesenho da aplicação

São Paulo

2021

Alkindar José Ferraz Rodrigues
Carolina de Moraes Josephik
Fabio Mendes Torres
Gabriely de Jesus Santos Bicigo
Leonardo Naoki Narita
Mariana da Silva Zangrossi

Lixt

Desesenho da aplicação

Desenho de aplicação para desenvolvimento
na disciplina de Projeto Integrado I no 1º
semestre de 2021.

Prof. Ivan Francolin Martinez

Prof. José Braz de Araujo

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Campus São Paulo

Tecnologia em Análise e Desenvolvimento de Sistemas

São Paulo

2021

Lista de abreviaturas e siglas

API	Application Programming Interface — Interface de programação de Aplicação. Citado em 2.1
HTTPS	Hypertext Transfer Protocol — Protocolo seguro de transferência de hipertexto. Citado em 2.1
REST	Representational State Transfer — Transferência de Representação de Estado: modelo de transferência de dados no qual o estado de um objeto é serializado e transferido entre aplicações. Citado em 2.1

1 Introdução

2 Desenvolvimento da Aplicação

Neste capítulo descrevemos os conceitos, análises e ferramentas utilizadas pela equipe TGT para o desenvolvimento do produto Lixt, incluindo os requisitos do projeto, as tecnologias utilizadas, e a arquitetura e a modelagem do produto. Isto é apresentado para que se possa estabelecer parâmetros e métricas que guiarão o desenvolvimento e a entrega final do projeto.

2.1 Arquitetura

Com base na análise do projeto, e nos requisitos que foram levantados como necessários, a arquitetura cliente-servidor é plausível como modelo para o produto que pretendemos entregar. Esta arquitetura é composta por duas aplicações distintas:

- Uma aplicação *front-end*, focada na interação com o usuário e apresentação de dados de uma forma agradável e intuitiva. Esta aplicação será implementada em JavaScript, com o framework React Native, e disponibilizada para as plataformas iOS e Android.
- Uma aplicação *back-end*, que será responsável por tratar os dados coletados no *front-end* e disponibilizar as informações que serão mostradas aos usuários. Como esta aplicação requer uma lógica de servidor, estabilidade e ampla disponibilidade, esta aplicação será implementada em Java, com uso do framework Spring Boot, que abstrai a criação de um servidor.

Podemos ver na [Figura 1](#) uma representação desta arquitetura.

A comunicação entre estes serviços será feita com o uso do protocolo [HTTPS](#), que permite a aplicação cliente realizar chamadas ao servidor através de urls, seja para buscar informações para apresentar ao usuário ou postar informações coletadas dele. O framework Spring, além de abstrair a implementação da lógica de um servidor, implementa *listeners* para estas urls, auxiliando a criação de pontos na aplicação do servidor focados na comunicação com a aplicação cliente.

O uso do protocolo HTTPS oferece algumas vantagens a aplicação *front-end*, que não precisa esperar uma solicitação ao *back-end* ser finalizada antes de realizar outras solicitações, aumentando a responsividade da aplicação cliente. Para além disso, quando combinada ao modelo [REST](#) na construção da [API](#), o protocolo HTTP oferece meios eficientes para que as aplicações se comuniquem.

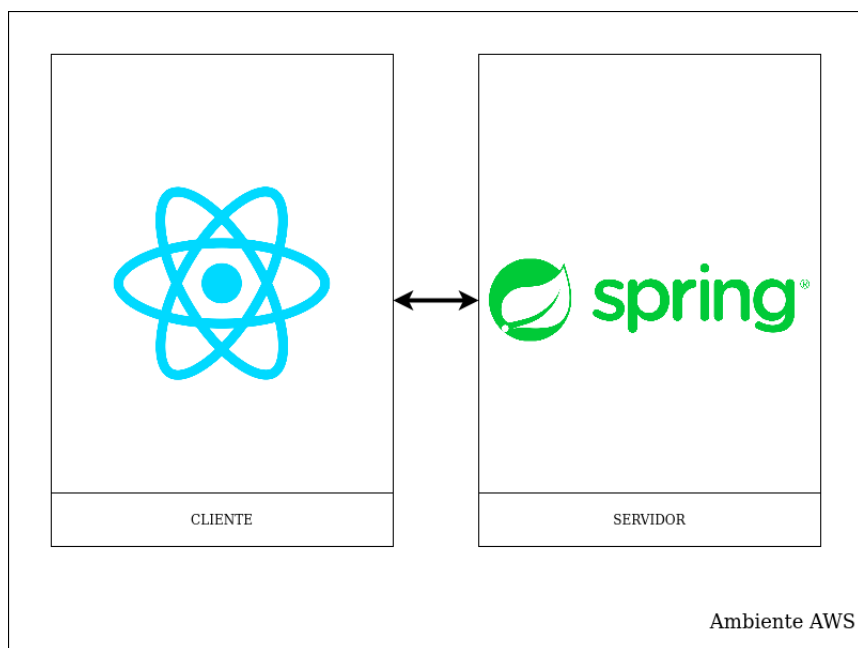


Figura 1 – Arquitetura Lixt

Como plataforma de servidor, o serviço AWS será utilizado, uma vez que é oferecida de maneira gratuita para a realização deste projeto, e nele serão armazenadas algumas instâncias da aplicação *back-end*. Esta redundância é necessária como forma de garantir a estabilidade do sistema, de forma que sempre haja alguma disponível para o processamento de novas requisições, e, em caso de falha numa delas, o serviço não seja interrompido aos usuários.

3 Viabilidade Financeira

O projeto de análise de viabilidade financeira consiste em averiguar a garantia de lucro sobre as despesas do projeto. Portanto, nesse projeto será descrito cada processo a fim de fazer essa verificação.

3.1 Gerenciamento de Custos

Nesse tópico, serão abordados temas de investimento inicial e de desenvolvimento do projeto, incluindo tópicos de análise de requisitos, desenvolvimento, manutenções e imprevistos.

3.1.1 Análise de Requisitos e Desenvolvimento

Para iniciar o projeto, é necessário fazer os primeiros planejamentos, elicitação de requisitos, abstrair e concretizar as primeiras ideias e fazer os primeiros planejamentos (diagramas, cronogramas e documentação). Logo após, o projeto chega na fase de desenvolvimento, onde é começado a se tornar real.

Contudo, o projeto não vai possuir nenhum custo de análise e implementação do sistema, devido ao fato de ser um projeto educacional.

3.1.2 Manutenções

Inevitavelmente, manutenções do sistema ocorrerão pós finalização do projeto e estar devidamente funcional em produção. Contudo, os custos de manutenções também não serão cobrados, devido a ser um projeto educacional.

3.2 Custos de deploy e de Ambiente de Produção

Nesse tópico, são apresentados os custos de manter o sistema funcional e disponível para os usuários. Desse modo, será feito uma previsão anual de cada plataforma utilizada:

3.2.1 Backend

Inicialmente gratuito no Amazon EC2, sendo permitido 750h de instâncias por mês durante o período de 12 meses.

A partir do momento que for necessário grande porte, será indicado o plano Sob Demanda do Amazon EC2, que garante viabilidade econômica e estratégica (visto que o preço é calculado a partir do uso).

Nesse plano, o custo de transferência de dados (ou seja, entrada e saída de dados) geram o valor de, na região de São Paulo, no máximo 0,15 USD por GB.

Na [Figura 2](#), seguem os preços dos planos Sob Demanda na região de São Paulo para Linux usando tipo de instância geral com 1 vCPU.



Nome da instância ▲	Taxa horária sob demanda ▼	vCPU ▼	Memória ▼	Armazenamento ▼	Performance das redes ▼
t2.nano	0,0093 USD	1	0.5 GiB	Somente EBS	Baixo
t2.micro	0,0186 USD	1	1 GiB	Somente EBS	Baixo a moderado
t2.small	0,0372 USD	1	2 GiB	Somente EBS	Baixo a moderado
m6g.medium	0,0612 USD	1	4 GiB	Somente EBS	Até 10 gigabits

Figura 2 – Preços do Amazon EC2 - Sob Demanda

3.2.2 Frontend

Tendo em vista que o projeto é *mobile* voltado para dispositivos Android, será publicado na PlayStore, estimando um valor de 25.00 USD anual.

3.2.3 Banco de Dados

Inicialmente gratuito no Amazon RDS, sendo permitido 750h de instâncias durante o período de 12 meses. O Amazon RDS possui suporte a vários Sistemas Gerenciadores de Banco de Dados (SGBD), incluindo o MySQL, que foi o SGBD optado para desenvolver a aplicação Lixt.

A partir do momento que for necessário grande porte, será indicado o plano Sob Demanda do Amazon RDS, que garante viabilidade econômica e estratégica (visto que o preço é calculado a partir do uso).

Na [Figura 3](#), seguem os preços dos planos Sob Demanda na região do Leste dos EUA (única opção disponível em 07/06/2021).

Instâncias de uso geral - Geração atual	vCPU	Memória	Preço por hora
db.mv11.medium	1	4 GiB	0,084 USD
db.mv11.large	2	8 GiB	0,168 USD
db.mv11.xlarge	4	16 GiB	0,336 USD
db.mv11.2xlarge	8	32 GiB	0,672 USD
db.mv11.4xlarge	16	64 GiB	1,344 USD
db.mv11.12xlarge	48	192 GiB	4,032 USD
db.mv11.24xlarge	96	384 GiB	8,064 USD

Figura 3 – Preços do Amazon RDS - Sob Demanda

3.3 Medidas de Obtenção de Retorno Financeiro

Para gerar uma receita positiva a fim de obter lucro, haverá duas formas principais de retorno financeiro:

- Cobrança do aplicativo: O aplicativo estará disponível gratuitamente na PlayStore, não gerando, portanto, retorno financeiro.
- Propaganda/Recomendação: Será utilizado mediador de anuncio AdMob (responsável por conectar aplicações e anunciantes), onde o valor varia por visualizações de anúncios e cliques neles. Contudo, no próprio site do admob, é citado um caso no qual houve 300.000 downloads e recebia, através do AdMob, 100 USD por dia. ((??))