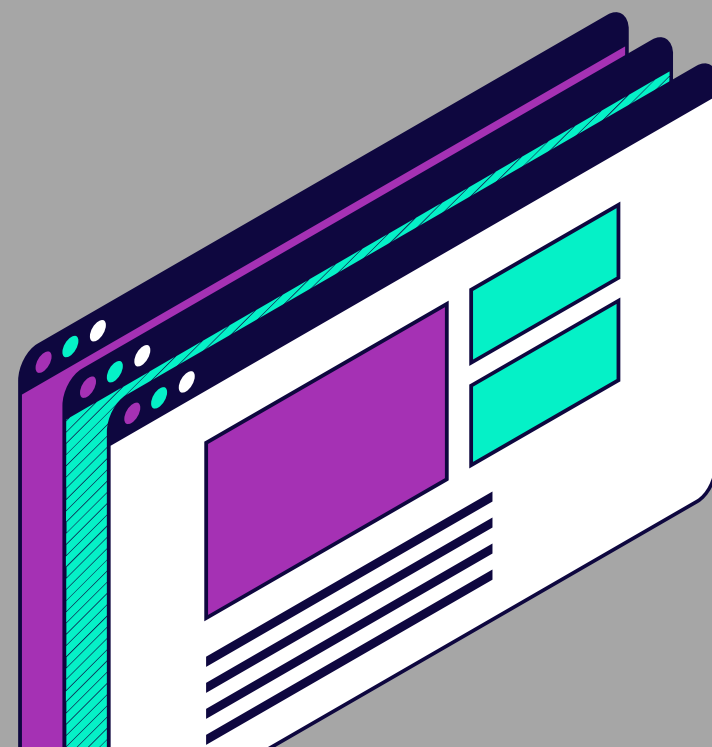




{ Java
GIRLS }

C CODEBANK



Integrantes



Andréa Letícia de S. Tavares.



Lilia Oliveira



Malu Caires Souza



Marina Duarte Utsch



Nadine Lau



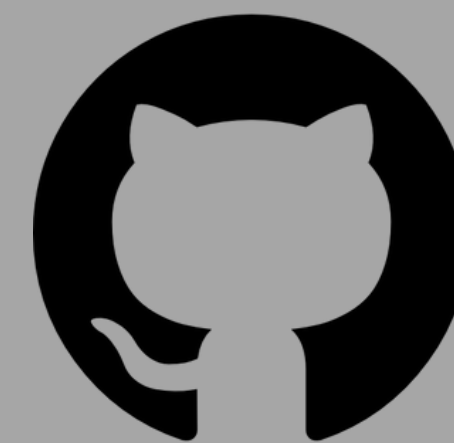
Tecnologias utilizadas na construção do projeto.



Slack



Jira



GitHub



IntelliJ



Postman

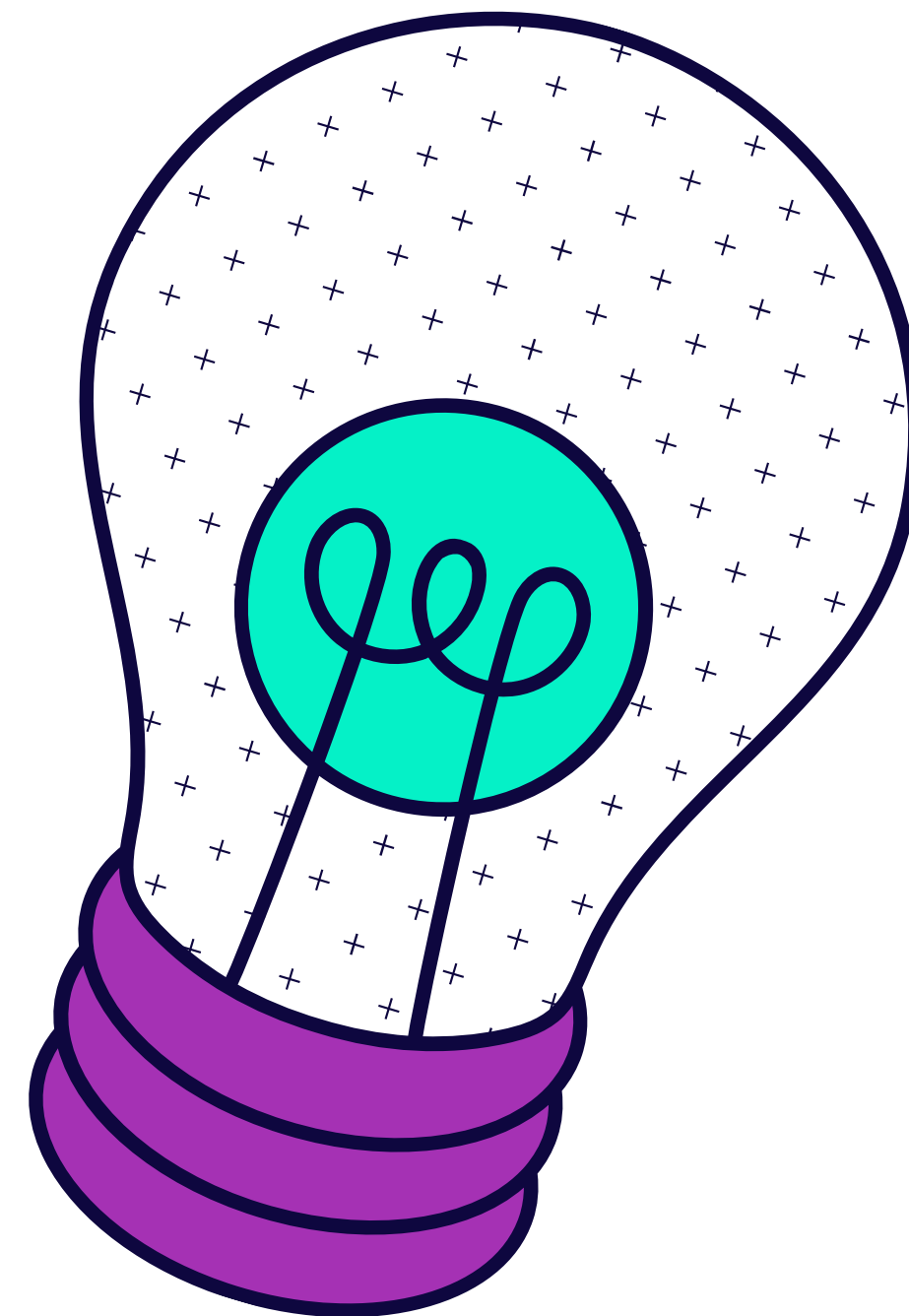


H2



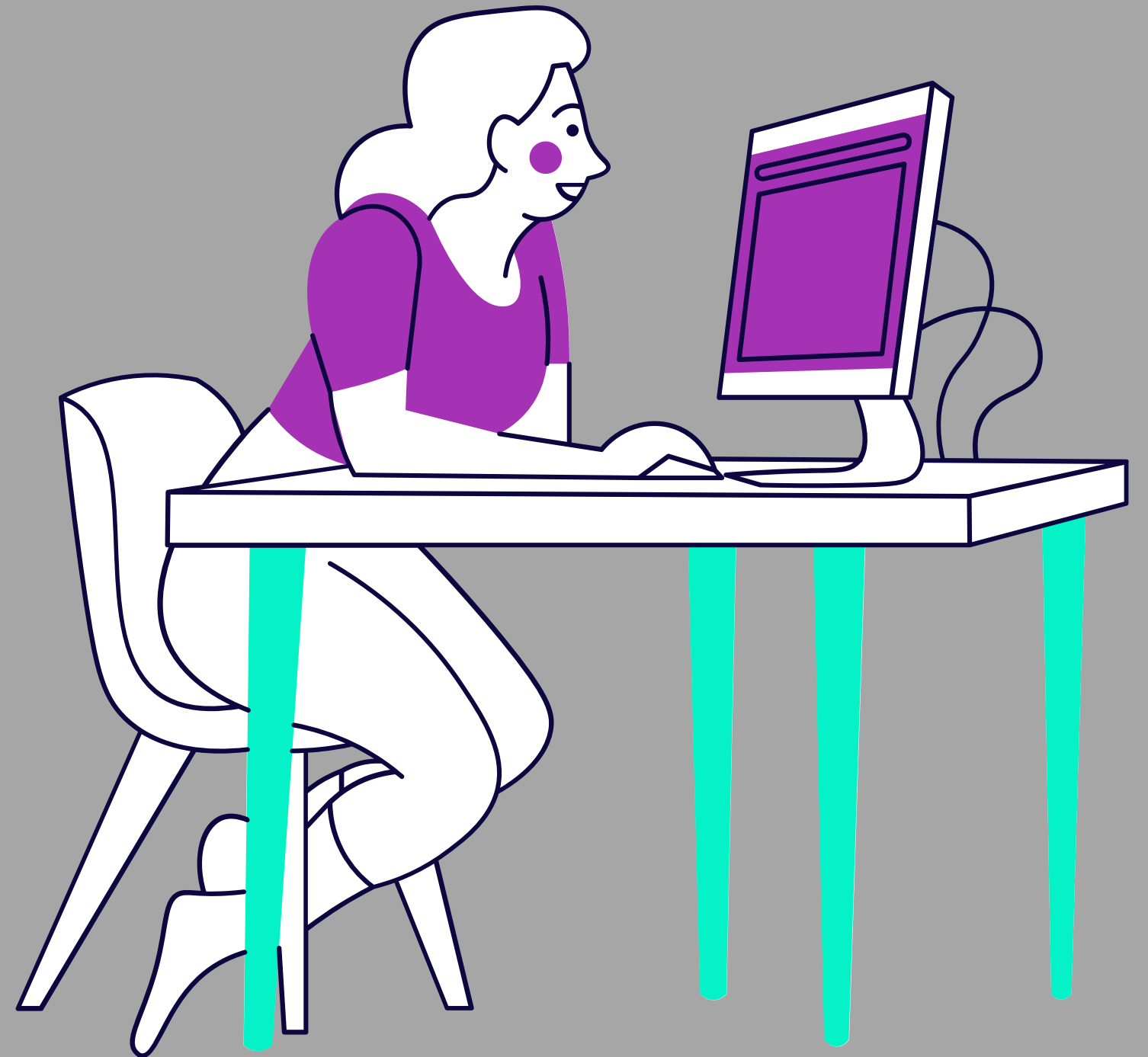
Pergunta:

O que é CodeBank?



O que é CodeBank?

O CodeBank é um sistema de gerenciamento de clientes e transações, que permite o cadastramento de novos clientes, com dados pessoais e dados para contato, e transferência de recursos entre contas.



{Requisitos Obrigatórios}

- Cadastro de clientes
- Busca dos clientes cadastrados
- Transferência de saldo entre as contas
- Busca de histórico de transações entre contas

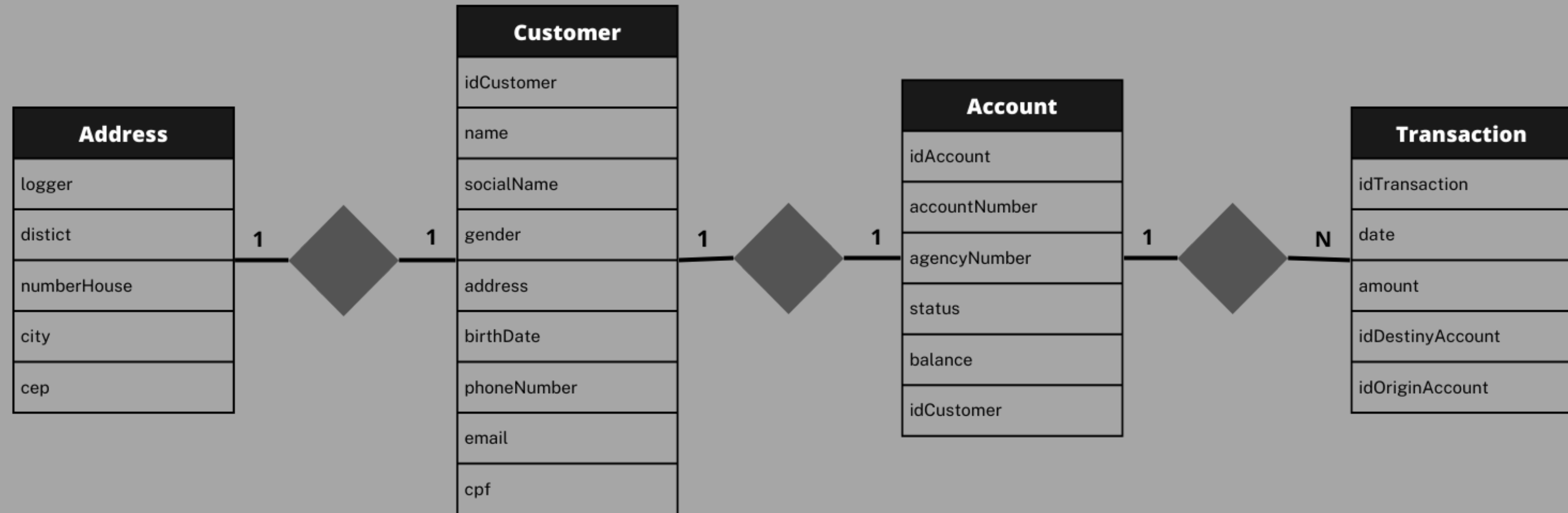


{Diferenciais}

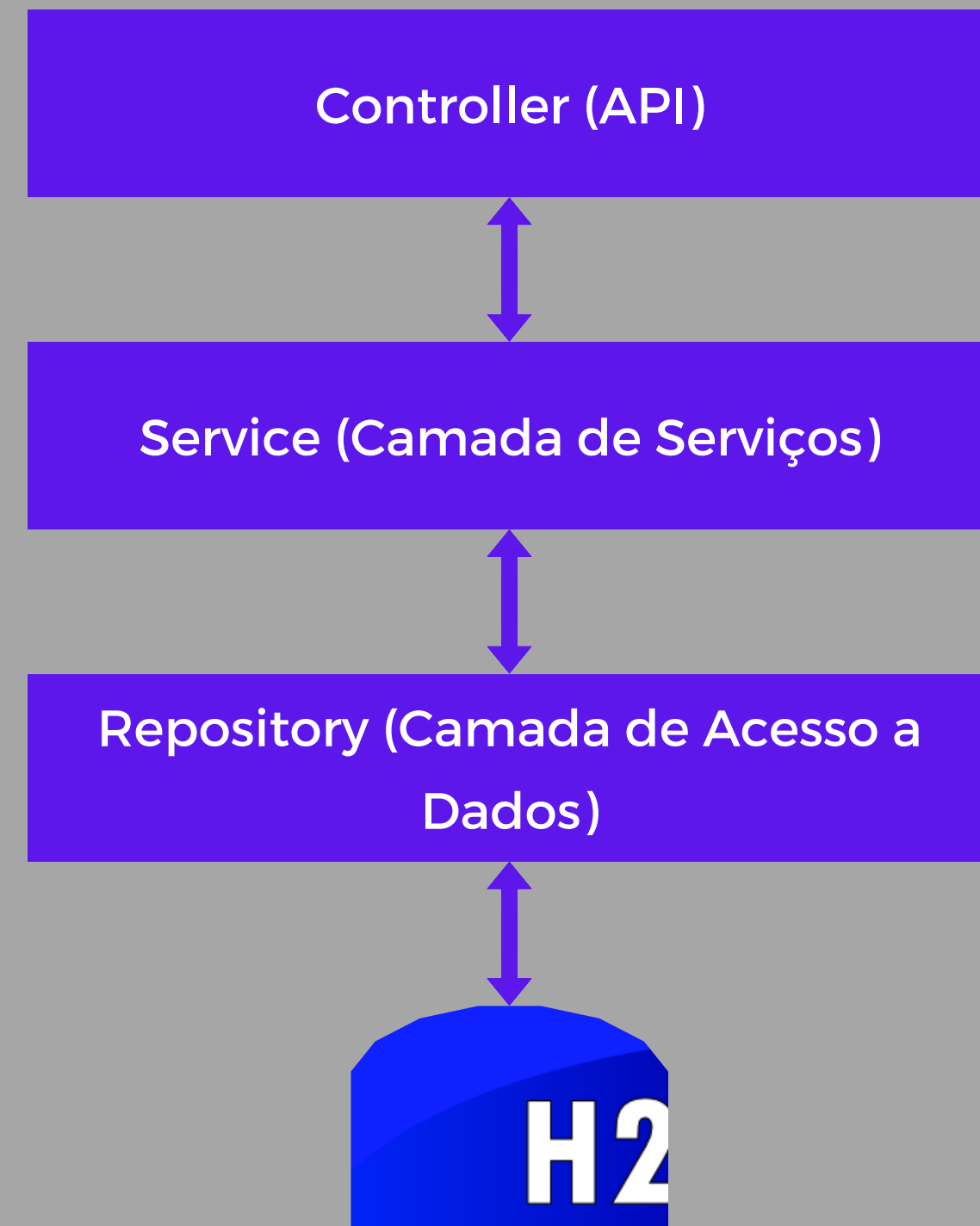
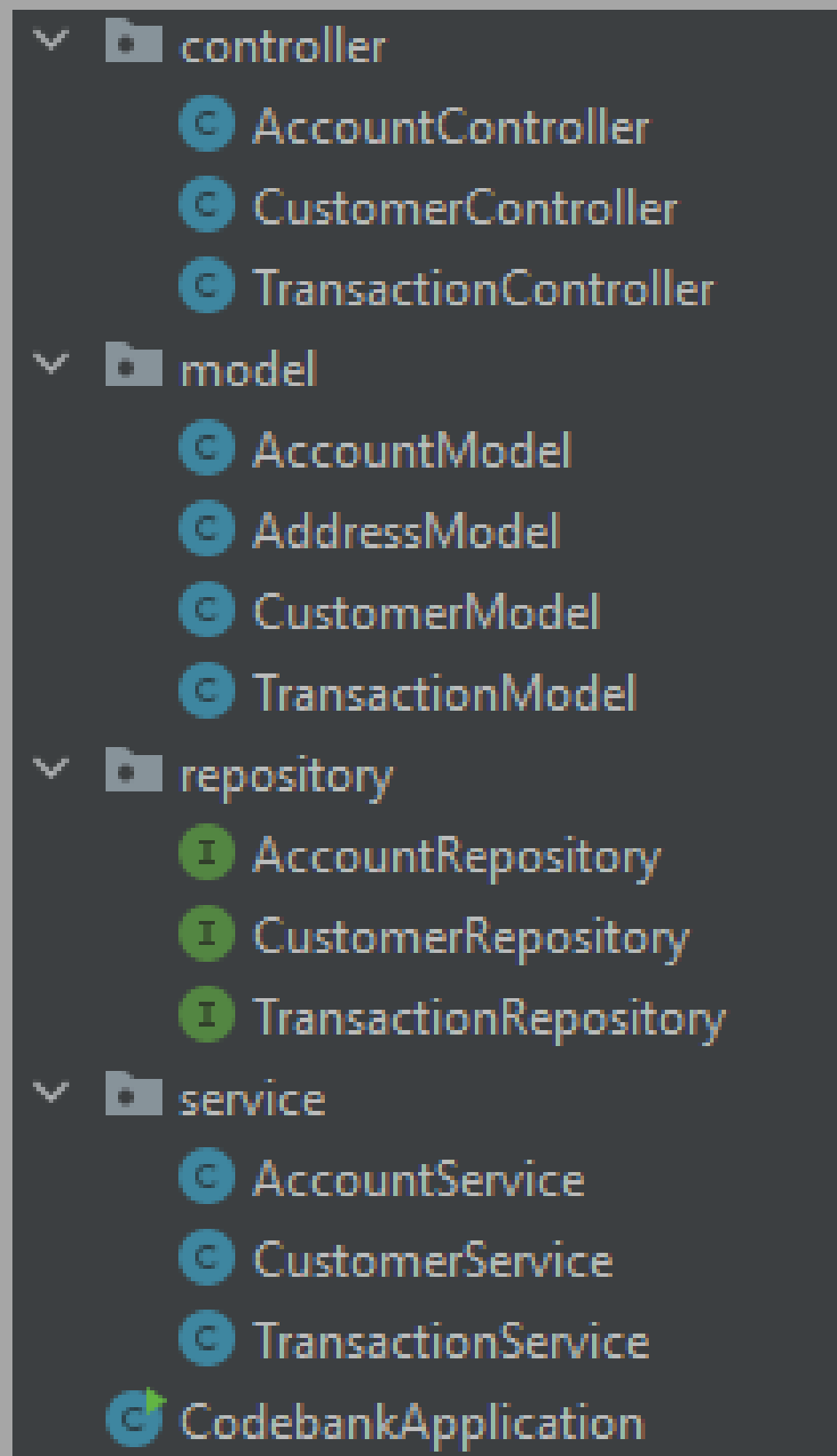
- Busca de clientes por CPF;
- Busca de contas pelo Status;
- Alteração de Status da Conta;
- Página de erro ao apresentar parametros inexistentes na busca;



{Diagrama Entidade Relacionamento}



{Arquitetura}



{CRUD}



/customer

GET

POST

PUT

DELETE

/account

GET

POST

PUT

DELETE

/actuator/health

GET

/actuator/metrics

GET

/transaction

GET

POST

DELETE



{Códigos}

```
@Embeddable
@Getter
@Setter
@NoArgsConstructor
@Entity
public class AddressModel {

    private String logger;
    private String district;
    private String city;
    private int num;
    private String cep;

}
```

```
@Getter
@Setter
@NoArgsConstructor
@Entity
public class CustomerModel {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", nullable = false)
    private Long idCustomer;
    private String name;
    private String socialName;
    private String gender;

    @Embedded
    private AddressModel address;
    private LocalDate birthDate;
    @Column(unique = true, updatable = false)
    private String cpf;
    @Column(unique = true)
    private String phoneNumber;

    @OneToOne
    @JoinColumn(name = "account_id")
    private AccountModel account;

}
```



{Códigos}

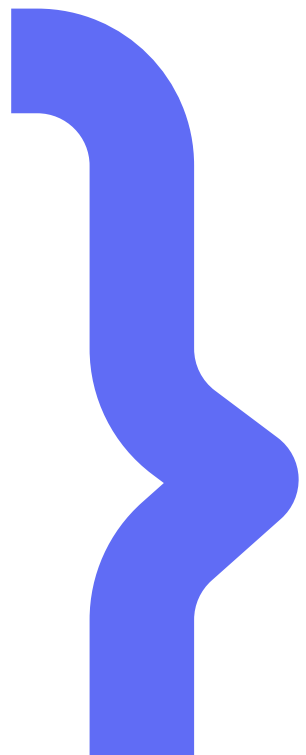
```
@Getter
@Setter
@NoArgsConstructor
@Entity
public class CustomerModel {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", nullable = false)
    private Long idCustomer;
    private String name;
    private String socialName;
    private String gender;
    1 usage
    @Embedded
    private AddressModel address;
    private LocalDate birthDate;
    @Column(unique = true, updatable = false)
    private String cpf;
    @Column(unique = true)
    private String phoneNumber;
    @OneToOne
    @JoinColumn(name = "account_id")
    private AccountModel account;
```

```
@Getter
@Setter
@NoArgsConstructor
@Entity
@SequenceGenerator(name = "ACC_SEQ", sequenceName = "ACCOUNT_SEQ", initialValue = 1000, allocationSize = 1)
public class AccountModel {
    1 usage
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", nullable = false)
    private Long idAccount;
    1 usage
    @Column(unique = true, insertable = true, updatable = false)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "ACC_SEQ")
    private int accountNumber;
    1 usage
    @Column(insertable = true, updatable = false)
    private int agencyNumber;
    2 usages
    private Boolean status;
    2 usages
    private float balance;
    1 usage
    @OneToOne
    @JoinColumn(name = "customer_id_customer")
    private CustomerModel customer;
    1 usage
    @OneToMany
    List<TransactionModel> transaction = new ArrayList<>();
```

```
@Getter
@Setter
@NoArgsConstructor
@Entity
public class TransactionModel {
    1 usage
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", nullable = false)
    private Long idTransaction;
    1 usage
    private Date date;
    1 usage
    private float amount;
    1 usage
    @ManyToOne
    @JoinColumn(name = "id_destiny_account_id")
    private AccountModel idDestinyAccount;
    1 usage
    @ManyToOne
    @JoinColumn(name = "id_origin_account_id")
    private AccountModel idOriginAccount;
```



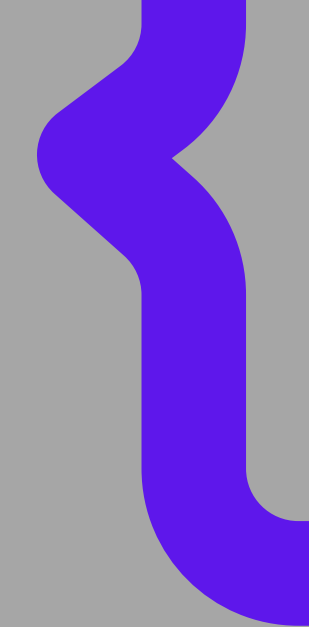
Projeto final



{Vídeo}

- <https://www.loom.com/share/ebdc6cb11837482bac8d2a877aacf6b3>





Obrigada!
Você chegou ao
final do curso.

E estamos muito orgulhosas de você!

