

Etec Dr. Geraldo José Rodrigues Alckmin

Creative Custom

Daniel Moreira Gonçalves da Silva
Gabriel Henrique Bessa da Silva
Samuel Sanches Guimaraes



CEETEPS

**Centro Estadual de Educação
Tecnológica “Paula Souza”**

**Etec Dr. Geraldo José Rodrigues
Alckmin**

Trabalho de Conclusão de Curso

**Habilitação Técnica de Nível Médio
de Técnico em Desenvolvimento de
Sistemas**

Orientador Prof. Gilberto Abud Junior

2024

DEDICATÓRIA

Dedicamos esse TCC(Trabalho de conclusão de curso) principalmente a Deus, aos nossos pais e aos nossos professores pela dedicação, amor incondicional e incentivo constante ao longo da nossa vida. Graças ao apoio de todos, pudemos chegar até aqui e realizar esse sonho.

AGRADECIMENTO

A jornada de conclusão deste trabalho de conclusão de curso (TCC) foi repleta de aprendizados e desafios, e nada disso seria possível sem o apoio de pessoas e instituições que me auxiliaram ao longo do caminho. Primeiramente, agradecemos o nosso orientador, Gilberto Abud Junior, por sua dedicação, paciência e valiosas orientações. Sua expertise e incentivo foram fundamentais para o desenvolvimento e aprimoramento deste projeto. Somos gratos aos nossos familiares e amigos pelo apoio incondicional, compreensão e palavras de incentivo nos momentos mais desafiadores. Vocês nos motivaram a seguir em frente e me proporcionaram o suporte necessário para superar as dificuldades. Agradecemos também à instituição Centro Estadual de Educação Tecnológica “Paula Souza” por nos proporcionar a oportunidade de realizar este estudo e por todo o conhecimento adquirido ao longo da graduação.

Agradeço também a todos os colegas que compartilharam momentos de aprendizado, dúvidas e alegrias durante a jornada acadêmica. Finalmente, agradecemos a todos que, de alguma forma, direta ou indiretamente, contribuíram para a realização deste TCC. Agradecemos a todos que acreditaram em nos e no nosso potencial.

EPIGRAFE

**“A única coisa que sei é que nada sei,
e isso me incomoda profundamente”.**

Sócrates

LISTA DE TABELAS E ILUSTRAÇÕES

- **Design Sem Limites** (EditOutlined)
- **Variedade de Produtos** (ShoppingOutlined)
- **Entrega Rápida** (RocketOutlined)
- **Qualidade Garantida** (HeartOutlined)
- **Suporte Dedicado** (TeamOutlined)
- **Preços Competitivos** (DollarOutlined)

LISTA DE ABREVIATURAS E SIGLAS

- **API:** Interface de Programação de Aplicações
- **tRPC:** TypeScript Remote Procedure Call
- **URL:** Localizador Uniforme de Recursos

SUMÁRIO

Dedicatória.....	03
Agradecimento.....	04
Epígrafe.....	05
Lista de tabelas e ilustrações.....	06
Lista de Abreviaturas e Siglas.....	07
Introdução.....	10

INTRODUÇÃO

Em um mundo cada vez mais conectado e individualista, a Creative Custom surge como a plataforma inovadora que redefine a personalização de produtos. Mais do que uma solução, a Creative Custom é uma experiência única que permite aos clientes expressarem sua individualidade através de produtos personalizados que refletem seu estilo e personalidade. Com uma interface intuitiva e amigável, a plataforma oferece uma ampla gama de opções para que cada cliente crie peças únicas e autênticas, democratizando o acesso à personalização com preços competitivos. A Creative Custom se destaca por sua tecnologia de ponta, foco em soluções sob medida e cultura centrada no cliente, buscando a excelência no atendimento e a satisfação total dos consumidores. A missão da Creative Custom é clara: tornar-se líder no mercado de personalização de produtos, ser reconhecida pela sua inovação, qualidade e excelência no atendimento ao cliente, e contribuir para o crescimento do mercado como um todo. Através de sua proposta inovadora e acessível, a empresa impulsiona o futuro da personalização, abrindo novas oportunidades para empresas e consumidores. Este trabalho de conclusão de curso apresenta em detalhes a proposta inovadora da Creative Custom, explorando o mercado de personalização de produtos, as funcionalidades da plataforma, o plano de negócios abrangente da empresa e os resultados esperados. Ao abordar a Creative Custom em sua totalidade, este estudo demonstra o potencial transformador da empresa nesse segmento promissor, posicionando-a como referência em personalização sob medida. Com a Creative Custom, a personalização se torna mais do que uma tendência, é uma forma de expressão individual e autêntica, onde cada cliente é o criador de sua própria história.

CRONOGRAMA DE ATIVIDADES

- 1. Pesquisa e Desenvolvimento de Tecnologia (Mês 1 e 2)**
- 2. Estratégia de Marketing e Branding (Mês 3 e 4)**
- 3. Lançamento da Plataforma e Suporte ao Cliente (Mês 5 e 6)**
- 4. Expansão e Aperfeiçoamento (Mês 7 e 8)**
- 5. Avaliação e Melhoria Contínua (A partir do Mês 9,10 e 11)**

OBJETIVO

O objetivo da Creative Custom é oferecer uma plataforma acessível e intuitiva de personalização e impressão sob demanda que permita aos usuários transformarem suas ideias em produtos únicos, personalizados com alta qualidade. A empresa busca democratizar o acesso à criação personalizada, capacitando qualquer pessoa a criar e vender seus próprios produtos, desde roupas até itens de decoração, com uma experiência de design sem limitações e suporte dedicado.

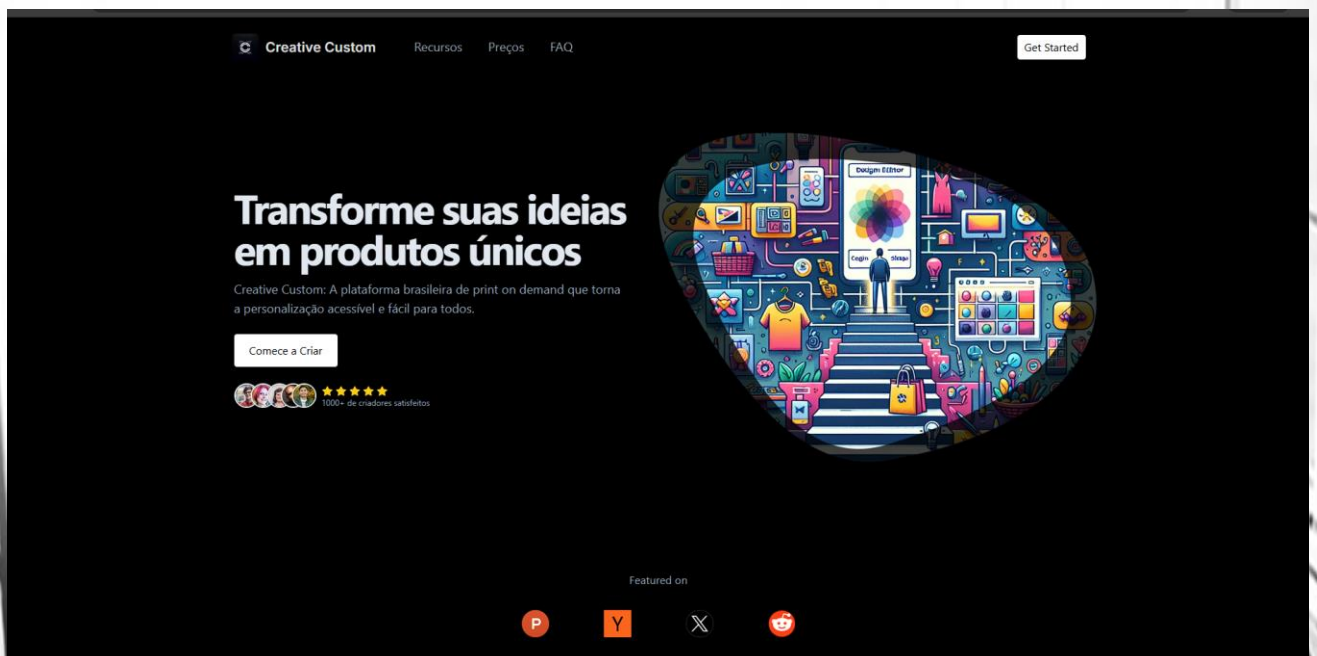
JUSTIFICATIVA

Percebemos uma oportunidade única: Uma loja brasileira de print on demand. O mercado brasileiro está pronto para uma nova experiência em personalização. Cansados da concorrência com empresas internacionais e da falta de opções customizadas para o nosso público, estamos lançando uma loja de print on demand 100% brasileira. Nos últimos anos, o termo “Print on demand” disparou em pesquisas, evidenciando uma oportunidade de negócio promissora no Brasil, com poucas empresas atuando nesse ramo. Quando olhamos para o futuro, as projeções do mercado de Print on Demand são extremamente animadoras. Segundo o relatório da Grand View Research, podemos esperar um crescimento anual composto (CAGR) de 26,1% até 2030.

LANDING PAGE:

```
<!-->  
    <head></head>  
    <body>  
      <script src="/_next/static/chunks/webpack-7a7e3a629db6bd1c.js" async></script>  
      <script>(self.__next_f=self.__next_f||[]).push([0]);self.__next_f.push([2,null])</script>  
      <script>  
        self.__next_f.push([1,"!HL[/\\_next/static/media/a34f9d1fa5f3315-s.p.woff2],\\font\\,{\"crossOrigin\":\"\", \"type\": \"font/woff2\"}]{n2:HL[/\\_next/static/css/ac5ceb84b52826b5.css\\,\"style\\\"]\\n"}]  
      </script>  
    </body>  
  </script>  
  <script>  
    self.__next_f.push([1,\"en\":[{\"$\",\"$L7\",null},{\"parallelRouterKey\":\"children\"},{\"segmentPath\":  
[\"children\"],[\"error\":{\"undefined\"},\"errorStyles\":{\"undefined\"},\"errorScripts\":{\"undefined\"},\"template\":{\"$\",\"$L8\",null,  
[]}],\"templateStyles\":{\"undefined\"},\"templateScripts\":{\"undefined\"},\"notFound\":{\"$\",\"$La\",null,{}},\"notFoundStyles\":  
[],\"styles\":{\"null\"},\"params\":{\"$b\"}],null,null},{\"couldBeIntercepted\":false,\"initialHead\":{\"null,\"$Lc\"},\"globalErrorComponent\":{\"$d\",\"missingSlots\":{\"$We\"}}]}(\\n")]  
  </script>  
  <script>  
    self.__next_f.push([1,\"c:[[{\"$\",\"meta\", \"0\"},{\"name\":\"viewport\", \"content\":{\"width=device-width, initial-scale=1}}],[\"$\",\"meta\", \"1\", {\"charset\":\"utf-8\"}], [\"$\",\"title\", \"2\",  
[\"children\":{\"welcome\"}], [\"$\",\"meta\", \"3\", {\"name\":\"description\", \"content\":{\"Creative Custom\"}], [\"$\",\"meta\", \"4\", {\"name\":\"next-size-adjust\"}]]\\n4:null\\n"]]  
  </script>  
  <next-route-announcer style="position: absolute;"></next-route-announcer>  
  <main class="dark">  
    <div class="bg-white text-black dark:bg-black dark:text-slate-200">  
      <div class="max-w-xl pt-4 mx-auto inset-x-0 z-50 w-[95%] lg:w-full">  
        <section class>  
          <div class="py-8 lg:py-44 px-5 max-w-7xl mx-auto grid lg:grid-cols-2 place-items-center relative">  
            <div class="relative z-10 p-4 md:p-0">  
              <div class="lg:absolute right-0 top-0 w-4/5 lg:w-1/2 h-full">  
                  
              </div>  
            </div>  
          </section>  
          <section class="py-16">  
            <div class="max-w-7xl mx-auto">  
              <h2 class="text-center text-slate-600 dark:text-slate-400">Featured on</h2>  
              <div class="flex gap-8 md:gap-20 items-center justify-center mt-8 flex-wrap">  
            </div>  
            </section>  
            <section class="py-16 px-5">  
              <div class="max-w-5xl mx-auto text-center">  
            </div>  
            </section>  
            <section class="py-16 px-5">  
              <div class="max-w-7xl mx-auto text-center">  
            </div>  
            </section>  
            <section class="py-16 px-5 id="features">  
              <div class="max-w-7xl mx-auto">  
                <h2 class="text-4xl lg:text-5xl font-bold lg:tracking-tight">Desbloqueie seu potencial criativo com nossas ferramentas poderosas</h2>  
                <p class="text-lg mt-4 text-slate-600 dark:text-slate-400">Descubra como o Creative Custom pode transformar suas ideias em realidade</p>  
              <div class="grid sm:grid-cols-2 md:grid-cols-3 mt-16 gap-16">  
                <div class="flex gap-4 items-start">
```

HOME PAGE



SOFTWARES E OUTROS RECURSOS DE APOIO

- **Linguagem JavaScript;**
- **CSS (Cascading Style Sheets);**
- **Visual Studio Code**
- **Node.js**
- **Typescript**
- **Prompt de Comando**
- **PRISMA**
- **DOCKER Desktop**
- **REACT**
- **TRPC**

RESULTADOS

ESPERADOS: A Creative Custom busca alcançar resultados como a criação de parcerias estratégicas com fornecedores globais confiáveis, garantindo qualidade premium nos produtos personalizados, e o desenvolvimento de uma plataforma intuitiva com domínio .com até 2026, oferecendo uma experiência de design fácil e acessível. Outro objetivo é construir um marketplace robusto, permitindo que artistas monetizem suas criações e pequenos empreendedores lancem suas próprias linhas de produtos, além de escalar operações para alcançar vendas expressivas e fidelizar uma base sólida de clientes, tornando-se referência no Brasil em personalização e print-on-demand.

ALCANÇADOS: a Creative Custom já definiu um plano estratégico detalhado, focado na estruturação da plataforma e na seleção de fornecedores, além de identificar parceiros internacionais para produção e entrega de produtos personalizados. Também concebeu um modelo de negócios inovador, que combina inteligência artificial, personalização criativa e um sistema de comissões para artistas no marketplace. O projeto conta ainda com um plano de ação escalável, metas claras para consolidação até 2026 e engajamento inicial com potenciais investidores e parceiros estratégicos para garantir o desenvolvimento e a expansão da iniciativa.

REFERÊNCIAS

- SALIBI NETO, José; MAGALDI, Sandro. A Era da Personalização: O Novo Mercado das Coisas. São Paulo: Editora Gente, 2022.
- DE BOTTON, Alain. O Futuro das Coisas. São Paulo: Companhia das Letras, 2020.
- SOUZA, Pedro A. de. A Empresa Futura: Uma Nova Abordagem para a Inovação e a Sustentabilidade. Rio de Janeiro: Editora Elsevier, 2021.
- SEMLER, Ricardo. O Novo Chic: O Futuro da Personalização e da Moda. São Paulo: Editora Campus, 2019.
- MELO, Claudio G. de. Inovação em Serviços: O Caso da Personalização. São Paulo: Editora Atlas, 2023.

ANEXOS

LOGIN PAGE:



Creative Custom
Bem-vindol

E-mail

Senha

[Esqueceu a senha?](#)

Entrar

Não tem uma conta? [Cadastre-se](#)

CODIGO LOGIN:

```
<!DOCTYPE html>
<html lang="en" style="background: black; color: white; --100vh: 945px;">
  <head>
    <style data-rc-order="prepend" rc-util-key="@ant-design-icons">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-1000" data-css-hash="1p4ikmq" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-1000" data-css-hash="effect-css-17asnhx-antZoomIn">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="1lujydpv" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="1w9id6l" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="1s54uuf" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="1soky4x" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="18mi75v" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="dsr3pn" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="1fpjme" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-999" data-css-hash="1jp6sup" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-rc-priority="-998" data-css-hash="1huo6e" data-token-hash="11hf6dd">⌘</style>
    <style data-rc-order="prependQueue" data-css-hash="hicwak" data-token-hash="x5ba8r">⌘</style>
    <style data-rc-order="prependQueue" data-css-hash="zsk21b" data-token-hash="11hf6dd">⌘</style>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="preload" href="/_next/static/media/a34f9d1faa5f3315-s.p.woff2" as="font" crossorigin type="font/woff2">
    <link rel="stylesheet" href="/_next/static/css/ac5ceb84b52026b5.css" data-precedence="next">
    <link rel="preload" as="script" fetchpriority="low" href="/_next/static/chunks/webpack-7a7e3a629db6bdc.js">
    <script src="/_next/static/chunks/f1feccf5-2c8c98903ef17d47.js" async></script>
    <script src="/_next/static/chunks/2670-6912c8f6ab49822.js" async></script>
    <script src="/_next/static/chunks/main-app-174f5811a9002514.js" async></script>
    <script src="/_next/static/chunks/8354-839670ebf8b5ac84.js" async></script>
    <script src="/_next/static/chunks/2479-e4b02271e5bf0401.js" async></script>
    <title>Creative Custom</title>
    <meta name="description" content="Creative Custom">
    <meta name="next-size-adjust">
    <script src="/_next/static/chunks/polyfills-78c92fac7aa8fdd8.js" nomodule></script>
    <style id="_goober">⌘</style>
  </head>
  <body>
    <script src="/_next/static/chunks/webpack-7a7e3a629db6bdc.js" async></script>
    <script>(self._next_f=self._next_f||[]).push([0]);self._next_f.push([2,null])</script>
    <script>
      self._next_f.push([1,"1:HL["/_next/static/media/a34f9d1faa5f3315-s.p.woff2"],"font",{"crossOrigin":"","type":"font/woff2"}]]n2:HL["/_next/static/css/ac5ceb84b52026b5.css","style"]n]]
    </script>
    <script>⌘</script>
    <script>⌘</script> == $0
    <script>
      self._next_f.push([1,"c:{{["$","meta"],"0"},{"name":"viewport","content":"width=device-width, initial-scale=1"}},{"$","meta","1"},{"charset":"utf-8"}],{"$","title","2"},
      {"children":["Creative Custom"]},[{"$","meta","3"},{"name":"description","content":"Creative Custom"}],{"$","meta","4"},{"name":"next-size-adjust"}]]n4:nulln]]
    </script>
    <next-route-announcer style="position: absolute;">
    #shadow-root (open)
      <div aria-live="assertive" id="_next-route-announcer_" role="alert" style="position: absolute; border: 0px; height: 1px; margin: -1px; padding: 0px; width: 1px; clip: rect(0px, 0px, 0px, 0px); overflow: hidden; white-space: nowrap; overflow-wrap: normal;"></div>
```

Docker

```
🔥 docker-compose.yml X
C: > Users > aluno > Desktop > zmnNH5-creativecustom-main > 🔥 docker-compose.yml
1  version: '3.8'
2  services:
3    database:
4      container_name: postgresql
5      image: postgres
6      restart: always
7      environment:
8        POSTGRES_USER: root
9        POSTGRES_PASSWORD: root
10       POSTGRES_DB: api
11     ports:
12       - '5442:5432'
13   pgadmin:
14     container_name: pgadmin4
15     image: dpape/pgadmin4
16     restart: always
17     environment:
18       PGADMIN_DEFAULT_EMAIL: admin@admin.com
19       PGADMIN_DEFAULT_PASSWORD: root
20     ports:
21       - '5052:80'
22   mail:
23     container_name: mail
24     image: axllent/mailpit
25     restart: always
26     volumes:
27       - ./data:/data
28     ports:
29       - 8022:8025
30       - 1022:1025
31     environment:
32       MP_MAX_MESSAGES: 5000
33       MP_DATA_FILE: /data/mailpit.db
34       MP_SMTP_AUTH_ACCEPT_ANY: 1
35       MP_SMTP_AUTH_ALLOW_INSECURE: 1
36
```

Prisma:

```
🔥 schema.prisma X
C: > Users > aluno > Desktop > zmnNH5-creativecustom-main > prisma > 🔥 schema.prisma
1  datasource db {
2    provider = "postgresql"
3    url      = env("SERVER_DATABASE_URL")
4  }
5
6  generator client {
7    provider = "prisma-client-js"
8  }
9
10 enum UserStatus {
11   INVITED
12   VERIFIED
13 }
14
15 /// @@allow('all', true)
16 model Product {
17   id          String    @id() @default(uuid())
18   name        String?
19   description String?
20   price       String?
21   imageUrl    String?
22   cartItems   CartItem[] @relation("product")
23   orderItems  OrderItem[] @relation("product")
24   dateCreated DateTime @default(now())
25   dateUpdated DateTime @default(now()) @updatedAt()
26 }
27
28 /// @@allow('all', true)
29 model Cart {
30   id          String    @id() @default(uuid())
31   userId      String?
32   user        User?     @relation(fields: [userId], references: [id], name: "user")
33   cartItems   CartItem[] @relation("cart")
34   dateCreated DateTime @default(now())
35   dateUpdated DateTime @default(now()) @updatedAt()
36 }
37
38 /// @@allow('all', true)
39 model Order {
40   id          String    @id() @default(uuid())
41   status      String?
42   totalAmount String?
43   shippingAddress String?
44   paymentInfo String?
45   userId      String?
46   user        User?     @relation(fields: [userId], references: [id], name: "user")
47   orderItems  OrderItem[] @relation("order")
48   dateCreated DateTime @default(now())
49 }
```

```

schema.prisma X
C:\Users\aluno > Desktop > zmnNH5-creativecustom-main > prisma > schema.prisma
39 model Order {
47   orderItems OrderItem[] @relation("order")
48   dateCreated DateTime @default(now())
49   dateUpdated DateTime @default(now()) @updatedAt()
50 }
51
52 /// @@allow('all', true)
53 model SupportMessage {
54   id String @id() @default(uuid())
55   name String?
56   email String?
57   message String?
58   userId String?
59   user User? @relation(fields: [userId], references: [id], name: "user")
60   dateCreated DateTime @default(now())
61   dateUpdated DateTime @default(now()) @updatedAt()
62 }
63
64 /// @@allow('all', true)
65 model CartItem {
66   id String @id() @default(uuid())
67   quantity Float?
68   customizationData String?
69   cartId String?
70   cart Cart? @relation(fields: [cartId], references: [id], name: "cart")
71   productId String?
72   product Product? @relation(fields: [productId], references: [id], name: "product")
73   dateCreated DateTime @default(now())
74   dateUpdated DateTime @default(now()) @updatedAt()
75 }
76
77 /// @@allow('all', true)
78 model OrderItem {
79   id String @id() @default(uuid())
80   quantity Float?
81   price String?
82   customizationData String?
83   orderId String?
84   order Order? @relation(fields: [orderId], references: [id], name: "order")
85   productId String?
86   product Product? @relation(fields: [productId], references: [id], name: "product")
87   dateCreated DateTime @default(now())
88   dateUpdated DateTime @default(now()) @updatedAt()
89 }
90
91 /// @auth
92 /// @@allow('all', auth().id == this.id)
93 /// @@allow('all', auth().globalRole == 'ADMIN')

```

```

schema.prisma X
C:\Users\aluno > Desktop > zmnNH5-creativecustom-main > prisma > schema.prisma
93 /// @@allow('all', auth().globalRole == 'ADMIN')
94 /// @@allow('create', globalRole != 'ADMIN')
95 /// @@allow('read', true)
96 model User {
97   id String @id() @default(uuid())
98   name String?
99   email String? @unique()
100  pictureUrl String? @default("https://i.imgur.com/ZdJSK3V.jpeg")
101  tokenInvitation String?
102  emailVerified Boolean?
103  stripeCustomerId String?
104  status UserStatus @default(VERIFIED)
105  globalRole String @default("USER")
106  /// @password
107  /// @omit
108  password String?
109  dateCreated DateTime @default(now())
110  dateUpdated DateTime @default(now()) @updatedAt()
111  accounts Account[]
112  sessions Session[]
113  carts Cart[] @relation("user")
114  orders Order[] @relation("user")
115  supportMessages SupportMessage[] @relation("user")
116 }
117
118 /// @@allow('all', auth().id == user.id)
119 model Account {
120   id String @id() @default(uuid())
121   userId String
122   type String
123   provider String
124   providerAccountId String
125   refresh_token String?
126   access_token String?
127   expires_at Int?
128   token_type String?
129   scope String?
130   id_token String?
131   session_state String?
132   user User @relation(fields: [userId], references: [id], onDelete: Cascade)
133   refresh_token_expires_in Int?
134
135   @@unique([provider, providerAccountId])
136 }
137
138 /// @@allow('all', auth().id == user.id)
139 model Session {
140   id String @id() @default(uuid())

```

```

141 sessionToken String @unique()
142 userId String
143 expires DateTime
144 user User @relation(fields: [userId], references: [id], onDelete: Cascade)
145 }
146

```

Inteligencia Artificial

```

ai.md x
C: > Users > aluno > Desktop > zmnNHS-creativecustom-main > documentation > ai.md > # AI tRPC Api ## Generate picture from prompt
1  # AI tRPC Api
2
3  ## Generate text from prompt
4
5  ```tsx
6  import { Api } from '@core/trpc'
7
8  const { mutateAsync: generateText } = Api.ai.generateText.useMutation()
9
10 const handleGenerateText = async (value: string) => {
11   const { answer } = await generateText({
12     prompt: `My prompt: ${value}`,
13   })
14 }
15 ```
16
17 ## Generate text from prompt and picture
18
19 ```tsx
20 import { Api } from '@core/trpc'
21
22 const { mutateAsync: generateText } = Api.ai.generateText.useMutation()
23
24 const handleGenerateText = async (value: string) => {
25   const { answer } = await generateText({
26     prompt: `My prompt: ${value}`,
27     attachmentUrls: ['https://pictureurl.jpg'],
28   })
29 }
30 ```
31
32 ## Generate picture from prompt
33
34 ```tsx
35 import { Api } from '@core/trpc'
36
37 const { mutateAsync: generateImage } = Api.ai.generateImage.useMutation()
38
39 const handleGeneratePicture = async (value: string) => {
40   const { url } = await generateImage({
41     prompt: `Generate a picture like: ${value}`,
42   })
43 }
44 ```
45

```

Sistema de email

```
email.md X
C > Users > aluno > Desktop > zmnnH5-creativecustom-main > documentation > email.md > Email TRPC Api

1 # Email TRPC Api
2
3 The codebase already includes a complete email library router that you can call from the front-end to send emails to a specific user.
4
5 ## Send an email to a user
6
7 ```tsx
8 import { Api } from '@core/trpc'
9
10 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
11
12 const handleSendEmail = async (value: string) => {
13   await sendEmail({
14     userId: user?.id,
15     subject: 'Email',
16     content: 'New email',
17   })
18 }
19 ...
20
21 ## Send an email to a user using a template
22
23 ```tsx
24 import { Api } from '@core/trpc'
25
26 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
27
28 const handleSendEmail = async (value: string) => {
29   await sendEmail({
30     userId: user?.id,
31     subject: 'Email',
32     templateKey: 'welcome',
33   })
34 }
35 ...
36
37 ## Create an email template
38
39 ### 1. Create a new .ts file in /src/server/libraries/email/templates/
40
41 ```ts
42 export const NewTemplate = `
43 <h2>Hello world</h2>
44 <div>
45   <p>
46     Hello {{name}}
47   </p>
48 </div>
49 `
50
51 ## 2. Add it to the list of templates in /src/server/libraries/email/email.service.ts
52
53 ```ts
54 import { NewTemplate } from './templates/newTemplate'
55
56 public templates = {
57   resetPassword: TemplateResetPassword,
58   verificationCode: TemplateVerificationCode,
59   welcome: TemplateWelcome,
60   newTemplate: NewTemplate
61 }
62 ...
63
64 ## 3. Use it like that
65
66 ```tsx
67 import { Api } from '@core/trpc'
68
69 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
70
71 const handleSendEmail = async (value: string) => {
72   await sendEmail({
73     userId: user?.id,
74     subject: 'Email',
75     templateKey: 'newTemplate',
76   })
77 }
78 ...
79
80
```

```
email.md X
C > Users > aluno > Desktop > zmnnH5-creativecustom-main > documentation > email.md > Email TRPC Api

37 ## Create an email template
38
39 ### 1. Create a new .ts file in /src/server/libraries/email/templates/
40
41 ```ts
42 export const NewTemplate = `
43   Hello {{name}}
44 `
45
46 ## 2. Add it to the list of templates in /src/server/libraries/email/email.service.ts
47
48 ```ts
49 import { NewTemplate } from './templates/newTemplate'
50
51 public templates = {
52   resetPassword: TemplateResetPassword,
53   verificationCode: TemplateVerificationCode,
54   welcome: TemplateWelcome,
55   newTemplate: NewTemplate
56 }
57 ...
58
59 ## 3. Use it like that
60
61 ```tsx
62 import { Api } from '@core/trpc'
63
64 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
65
66 const handleSendEmail = async (value: string) => {
67   await sendEmail({
68     userId: user?.id,
69     subject: 'Email',
70     templateKey: 'newTemplate',
71   })
72 }
73 ...
74
75
```

Sistema de email

```
email.md X
C > Users > aluno > Desktop > zmnnH5-creativecustom-main > documentation > email.md > Email TRPC Api

1 # Email TRPC Api
2
3 The codebase already includes a complete email library router that you can call from the front-end to send emails to a specific user.
4
5 ## Send an email to a user
6
7 ```tsx
8 import { Api } from '@core/trpc'
9
10 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
11
12 const handleSendEmail = async (value: string) => {
13   await sendEmail({
14     userId: user?.id,
15     subject: 'Email',
16     content: 'New email',
17   })
18 }
19 ...
20
21 ## Send an email to a user using a template
22
23 ```tsx
24 import { Api } from '@core/trpc'
25
26 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
27
28 const handleSendEmail = async (value: string) => {
29   await sendEmail({
30     userId: user?.id,
31     subject: 'Email',
32     templateKey: 'welcome',
33   })
34 }
35 ...
36
37 ## Create an email template
38
39 ### 1. Create a new .ts file in /src/server/libraries/email/templates/
40
41 ```ts
42 export const NewTemplate = `
43 <h2>Hello world</h2>
44 <div>
45   <p>
46     Hello {{name}}
47   </p>
48 </div>
49 `
50
51 ## 2. Add it to the list of templates in /src/server/libraries/email/email.service.ts
52
53 ```ts
54 import { NewTemplate } from './templates/newTemplate'
55
56 public templates = {
57   resetPassword: TemplateResetPassword,
58   verificationCode: TemplateVerificationCode,
59   welcome: TemplateWelcome,
60   newTemplate: NewTemplate
61 }
62 ...
63
64 ## 3. Use it like that
65
66 ```tsx
67 import { Api } from '@core/trpc'
68
69 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
70
71 const handleSendEmail = async (value: string) => {
72   await sendEmail({
73     userId: user?.id,
74     subject: 'Email',
75     templateKey: 'newTemplate',
76   })
77 }
78 ...
79
80
```

```
email.md X
C > Users > aluno > Desktop > zmnnH5-creativecustom-main > documentation > email.md > Email TRPC Api

37 ## Create an email template
38
39 ### 1. Create a new .ts file in /src/server/libraries/email/templates/
40
41 ```ts
42 export const NewTemplate = `
43   Hello {{name}}
44 `
45
46 ## 2. Add it to the list of templates in /src/server/libraries/email/email.service.ts
47
48 ```ts
49 import { NewTemplate } from './templates/newTemplate'
50
51 public templates = {
52   resetPassword: TemplateResetPassword,
53   verificationCode: TemplateVerificationCode,
54   welcome: TemplateWelcome,
55   newTemplate: NewTemplate
56 }
57 ...
58
59 ## 3. Use it like that
60
61 ```tsx
62 import { Api } from '@core/trpc'
63
64 const { mutateAsync: sendEmail } = Api.email.send.useMutation()
65
66 const handleSendEmail = async (value: string) => {
67   await sendEmail({
68     userId: user?.id,
69     subject: 'Email',
70     templateKey: 'newTemplate',
71   })
72 }
73 ...
74
75
```