

PROTÓTIPO DE APLICAÇÃO COM FRAMEWORK FLUTTER PARA O TERCEIRO SETOR

Fernando Henrique Rovaris¹

Luciano Antunes²

Camila Serafim Daminelli³

Resumo: Criar uma aplicação própria com o objetivo de alcançar a maior população possível torna-se inviável economicamente, principalmente quando se trata de instituições do terceiro setor que realizam ações sociais. Esta pesquisa busca explorar as características e funcionalidades do *framework* Flutter no desenvolvimento de uma aplicação mobile para terceiro setor, buscando aproximar projetos sociais e possíveis voluntários, que compartilham os mesmos ideais, através de campanhas.

Palavras-chave: Flutter. Multiplataforma. Terceiro Setor. Projetos Sociais.

ABSTRACT: Creating your own mobile application with the aim of reaching the largest possible population becomes economically unfeasible, especially when it comes to third sector institutions that carry out social actions. This research seeks to explore the characteristics and functionalities of the Flutter *framework* in the development of a mobile application for the third sector, seeking to bring together social projects and potential volunteers, who share the same ideals, through campaigns.

Keywords: Flutter. Multiplatform. Third sector. Social projects.

1 INTRODUÇÃO

Segundo o Centro Regional de Informação para a Europa Ocidental (UNRIC, 2019) mais de 780 milhões de pessoas vivem atualmente abaixo do Limiar Internacional da Pobreza (com menos de 1,90 dólar por dia). Mais de 11% da população mundial vive na pobreza extrema e luta para satisfazer as necessidades mais básicas na esfera da saúde, da educação, do acesso à água e ao saneamento.

No Brasil a revisão nos números de beneficiados e no valor dos benefícios, pelos programas federais, a partir de 2017, e sobretudo, o desemprego e a inflação provocados pela Pandemia de Covid-19 trouxeram novamente o problema da fome para muitos lares brasileiros (COSTA, 2020). Isto demonstra, dentre outras coisas, que a segurança alimentar no país é um fenômeno instável entre as famílias mais

¹ Fernando_rovaris@unesco.net.

² luciano@unesco.net.

³ csd@unesco.net.

pobres, e que a erradicação da fome requer medidas mais amplas, integradas e permanentes.

Aliados à fome e à precarização das condições de vida das famílias, os índices de violência demonstram que ambos os fenômenos estão intimamente relacionados, em que os efeitos e as consequências de um composto aumentam os efeitos do outro. Seguindo essa lógica pode-se afirmar que a fome provoca um encadeamento de problemas sociais que afetam, sobretudo, as crianças e os adolescentes: o abandono escolar, o subemprego, a cooptação pelo tráfico de drogas, etc... (ANDRADES et al., 2019).

Compreende-se como Terceiro Setor aquele formado por organismos e entidades civis, comunitárias ou religiosas sem fins lucrativos, que empreendem ações de cunho social no intuito de promovam acesso a bens, bem-estar e que tragam qualidade de vida aos públicos atendidos (MENDONÇA et al., 2017). O levantamento de fundos constitui um dos maiores desafios para as organizações, pois trata da busca, arrecadação e captação de recursos financeiros de diferentes fontes ou potenciais doadores como pessoas ou organizações, que compartilham da mesma visão e estão dispostas a contribuir na edificação do projeto.

A busca dos recursos é realizada por meio de convênios estaduais, parcerias com empresas privadas ou por campanhas de doação (SILVA et al., 1997). Visando auxiliar os projetos sociais na arrecadação e captação de recursos, com a construção de uma aplicação multiplataforma direcionada ao compartilhamento das missões, visões e valores, para que as pessoas ou organizações que compartilharem dos mesmos, possam contribuir de forma sustentável nas doações de recursos que são abundantes ou até mesmo que poderiam ser descartados por não uso como brinquedos, roupas e utilitários.

Aplicações móveis são utilizadas para facilitar e auxiliar no cotidiano das pessoas em diferentes contextos, como por exemplo, em realização de compras online, compartilhamento de lembranças em redes sociais, aproximação de amigos e familiares geograficamente distantes, entre outros (LIMA, 2019).

A elaboração de aplicativos pode ter, contudo, um alto custo. Se considerarmos a criação de um aplicativo nativo para cada plataforma que se deseja alcançar, certamente o valor ficará inviável para a maioria das instituições que destinam o pouco

recurso que conseguem nos projetos que executam para pessoas e famílias mais necessitadas. Considerando, ainda, que cada plataforma utiliza uma linguagem de programação diferente para construção de seus aplicativos nativos, existe a necessidade de o desenvolvedor conhecer e dominar cada uma de forma individual, o que levaria anos de estudos e trabalho. No entanto, existem *frameworks* de desenvolvimento híbrido criados para diminuir o custo e a manutenção das aplicações, sendo necessária apenas a criação de um código fonte para diferentes plataformas (LIMA, 2019).

Flutter é o atual *framework* de código aberto da empresa Google. Lançada em 2017, utiliza a linguagem de programação Dart. Seu uso tem crescido muito entre desenvolvedores e empresas (LIMA, 2019). O Flutter vem se tornando um dos principais *frameworks* utilizados no mercado atualmente. Podemos citar algumas de suas principais características, como a alta escalabilidade e utilização de componentes chamados *widgets*, dispõe de interfaces personalizadas e únicas. Flutter evita os problemas de desempenhos encontrados em *frameworks* que utilizam pontes de compilação da linguagem de programação utilizada para a nativa. Isto ocorre pela linguagem Dart dar suporte à compilação antecipadamente, para código nativo chamado de *Ahead-Of-Time* (AOT) (LIMA, 2019 apud GAŽO, 2018).

Quando se trata de aplicações voltadas para o terceiro setor se tem alguns exemplos já desenvolvidas, como PARANÁ SOLIDÁRIO e UMA AJUDA BOA são iniciativas brasileiras que conecta doadores a instituições para o terceiro setor, ferramentas que criam pontes entre empresas e pessoas físicas que queiram doar para instituições de caridade produtos como alimentos, roupas, móveis, eletrodomésticos, ração para animais, entre outros (PARANÁ SOLIDÁRIO, 2020; ARAÚJO, 2021). Segundo a Fundação Instituto de Administração (FIA, 2019) os projetos sociais se tornaram cada vez mais importantes para garantir os direitos básicos e assistência social para as populações desamparadas ou afetadas por crises humanitárias. Nesse sentido, os trabalhos dos projetos sociais tornaram-se fundamentais para garantir o mínimo de dignidade a uma população que vive nas sombras, em extrema vulnerabilidade, garantindo também amparo às pessoas em espaços que o estado não consegue chegar.

Tendo em vista a importância dos projetos sociais, e as limitações das aplicações encontradas, voltadas a um uso leigo, cada nova aplicação criada é uma ferramenta que auxilia os projetos a alcançarem seus objetivos. Este projeto tem como objetivo aplicar o `framework` Flutter no desenvolvimento de uma aplicação mobile para o terceiro setor. Sua relevância no âmbito social é significativa, pois almeja criar oportunidade para que pessoas ou organizações que compartilham dos mesmos valores possam, de forma prática, se colocarem em contato e compartilhar recursos. O horizonte da aplicação desenvolvida é o seu uso real, na qual pode-se contribuir na disseminação de valores para uma sociedade mais solidária e sustentável.

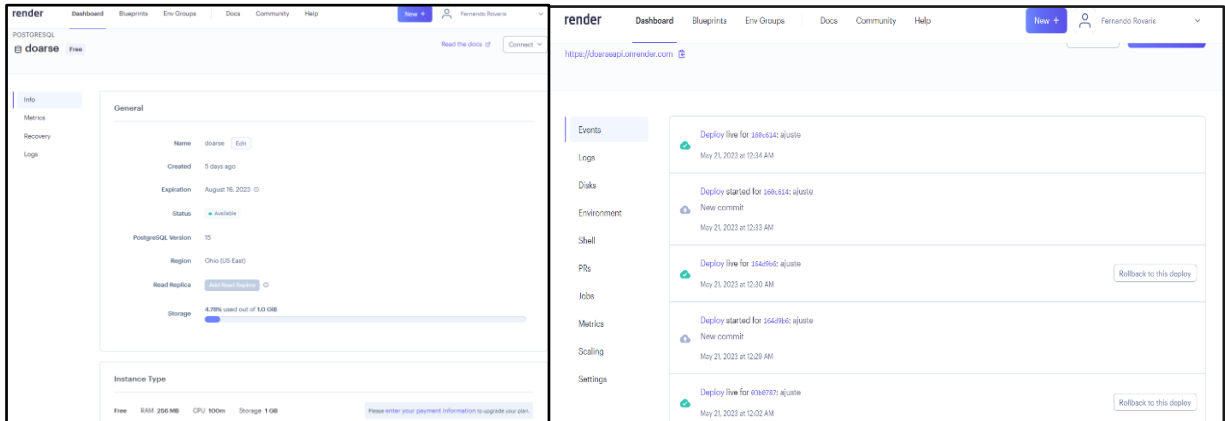
2 MATERIAIS E MÉTODOS

Foram utilizados como ferramentas na construção do aplicativo, um notebook da marca Dell modelo G3 15 3590 com 8 GB de memória RAM Dual Channel, processador Intel(R) Core (TM) i5-9300H CPU @ 2.40GHz, placa gráfica de NVIDIA GeForce GTX 1650, rodando sistema operacional Windows 11.

Como ferramenta de edição foi utilizado o programa de código fonte aberto desenvolvido pela Microsoft, Visual Studio Code na versão 1.78.2 que contém uma grande quantidade de extensões, permitindo atribuir várias funcionalidades diferentes, como indentação automática, autocomplete na escrita, e versionamento de código juntamente com GitHub, utilizado para hospedar o código fonte da aplicação.

Para a hospedagem da aplicação do servidor e da Base de Dados foi utilizada a plataforma de nuvem render como mostra a figura 1, vinculada ao repositório do GitHub o render detecta toda alteração no código fonte, e faz Deploy automático para nuvem, mostrando log de todo processo no Dashboard da própria plataforma. O render também foi utilizado na criação da base de dados utilizando sistema de gerenciamento PostgreSQL na versão 15, porém o plano gratuito da plataforma restringe a base a 1GB de armazenamento e 256 MB de memória RAM, com expiração da base em seis meses. Após este período a base é apagada da plataforma.

Figura 1 – Tela Ferramenta `render`[DMM2].

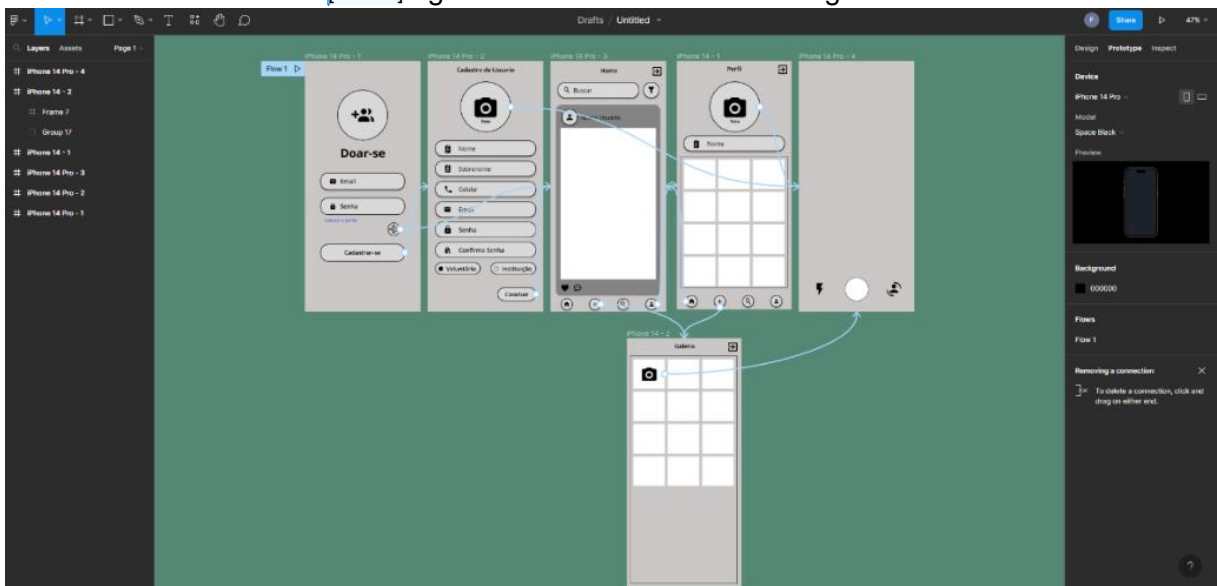


Fonte: Do autor

Durante o desenvolvimento da aplicação foram utilizados como ferramentas de teste um Smartfone da marca Motorola modelo moto g62 5G executando o sistema operacional Android versão 12, 4GB de memória RAM, 128 GB de armazenamento e tela de 6.5 polegadas com uma resolução de 2400x1080 pixels, também foi utilizado o navegador Google Chrome na versão 113.0.5672.127 64 bits.

2.1 Criação da Interface

[DMM3]Figura 2 – Tela da ferramenta Figma.



Fonte: Do autor

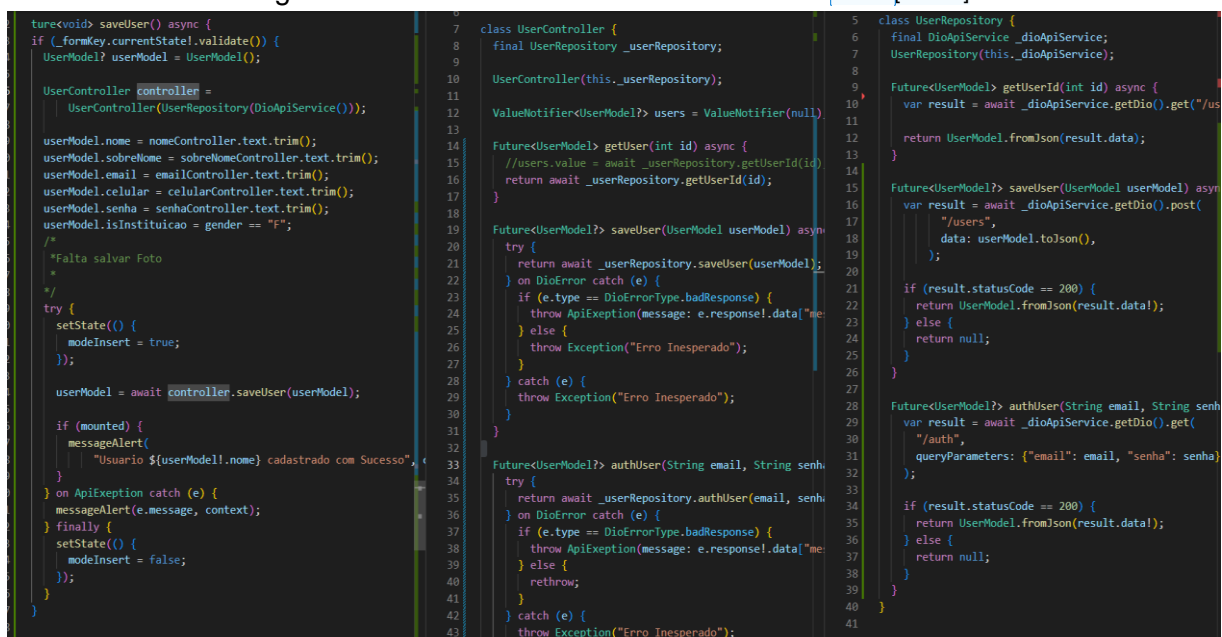
Para a criação do design de interface foi utilizado a ferramenta [Figma](#) ^[DMM4] na sua versão gratuita, com objetivo de visualizar e confeccionar telas de forma rápida, podendo reutilizar telas já existentes, além de gerenciar o fluxo entre elas, como demonstrado na figura 2.

O processo de construção e organização das telas foi essencial na construção da aplicação, pois garante ao programador que as decisões visuais já foram tomadas, sendo necessário implementar o mais próximo possível do planejado.

2.2 Padrão de Projeto

Para a construção da aplicação foi adotada como base a arquitetura de *software* modelo MVC nas camadas de *front-end* e *back-end*.

Figura 3 – Telas da ferramenta visual studio [code](#) ^[DMM5].



```

1 ture<void> saveUser() async {
2   if (!_formKey.currentState.validate()) {
3     UserModel? userModel = UserModel();
4
5     UserController controller =
6       UserController(UserRepository(DioApiService()));
7
8     userModel.nome = nomeController.text.trim();
9     userModel.sobrenome = sobreNomeController.text.trim();
10    userModel.email = emailController.text.trim();
11    userModel.celular = celularController.text.trim();
12    userModel.senha = senhaController.text.trim();
13    userModel.isInstituicao = gender == "F";
14  }
15  /*Falta salvar Foto
16  */
17  try {
18    setState(() {
19      modeInsert = true;
20    });
21
22    userModel = await controller.saveUser(userModel);
23
24    if (mounted) {
25      messageAlert(
26        "Usuario ${userModel.nome} cadastrado com Sucesso",
27      );
28    }
29  } on ApiException catch (e) {
30    messageAlert(e.message, context);
31  } finally {
32    setState(() {
33      modeInsert = false;
34    });
35  }
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fonte: Do autor.

A figura 3 contém trechos do código fonte com as três camadas do modelo aplicadas. A camada *view* representada pela função *saveUser* na imagem da esquerda faz uma chamada para o método de mesmo nome da classe *UserController*, que representa nossa camada de *controler* e conseqüentemente chama o método da

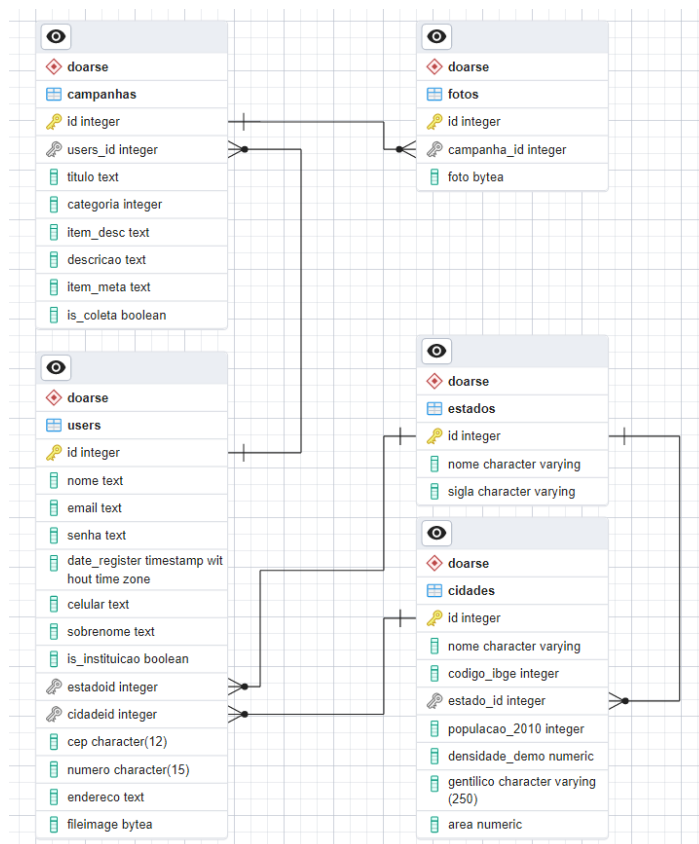
classe *UserRepository*, representando a camada *model*, que por sua vez faz a requisição dos dados ao servidor.

2.3 Definição da Base de Dados

O sistema de gerenciamento de banco de dados (SGBD) utilizado na construção da aplicação foi *PostgreSQL*, pelo fato de ser um projeto de código aberto gratuito, muito utilizado no mercado para aplicações de pequeno e grande porte.

Como o *PostgreSQL* é um sistema gerenciador de dados relacional, a definição da base de dados e suas tabelas foram definidas no formato demonstrado pelo diagrama ER da figura 4. É importante ressaltar que foram criados índices para todas as chaves estrangeiras com o intuito de otimizar a busca em eventuais procuras relacionadas, como usuários e suas campanhas. Para construção do diagrama foi utilizada a ferramenta pgAdmin 4 na versão 7.

Figura 4 – Tela da ferramenta pgAdmin.



Fonte: Do autor

2.4 Servidor

Para desenvolver o servidor foi utilizada a linguagem de programação JavaScript juntamente com a plataforma Node.js na versão 18.15.0. Também foram utilizadas dependências como a biblioteca *pg-promise* para comunicação com a base de dados *PostgresSQL*, e *framework* Express para gerenciamento de requisição web e construção de API Rest.

Como citado acima, a construção do servidor utilizou uma arquitetura de *software* em modelo MVC, com todas as divisões bem definidas, facilitando a leitura e a manutenção do código.

O *framework* Express é responsável por manter o servidor no ar, sendo responsável em requisitar respostas da API de integração. A partir deste ponto toda requisição que chega ao servidor, passa por arquivo de rota que podemos definir como camada de *view* da arquitetura de *software*, e conseqüentemente esta camada faz uma chamada à camada de controle, que para cada rota tem um método específico, como buscar todos os usuários através da rota *"/users"* do método GET do HTTP ou buscar apenas um usuário através da rota *"/users/:id"* com o mesmo método. A camada *model* do servidor é dividida em duas classes sendo a classe de serviços a que contém a regra de negócio, onde se valida um e-mail que já foi cadastrado, por exemplo, e a classe repositório, que faz a busca diretamente na base de dados, utilizando código SQL.

2.5 Flutter

A construção da aplicação para terceiro setor utilizando o *framework* Flutter na sua versão 3.3.10 no canal estável está representada na figura 5, juntamente com todas as dependências necessárias para o seu funcionamento.

Figura 5 – Tela do terminal do Windows.

```
C:\Users\ferna>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.3.10, on Microsoft Windows [vers o 10.0.22621.1702], locale pt-BR)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.1)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.4.3)
[✓] Android Studio (version 2021.3)
[✓] VS Code (version 1.78.2)
[✓] Connected device (4 available)
[✓] HTTP Host Availability

• No issues found!
```

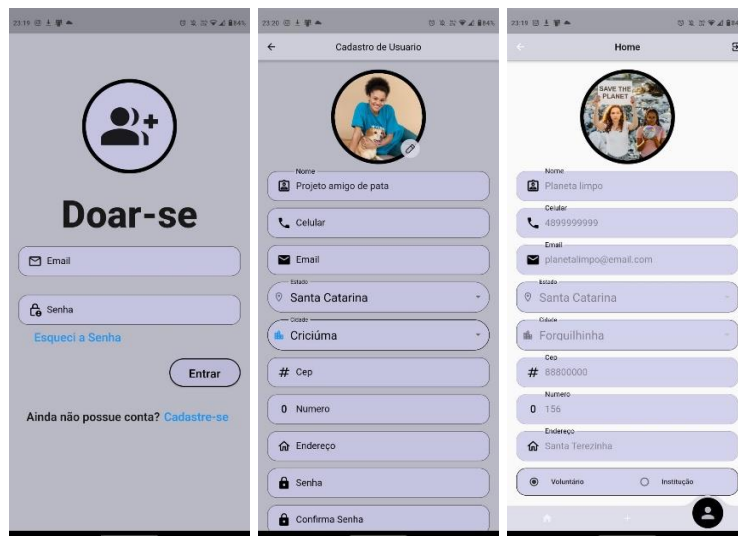
Fonte: Do autor

Foram utilizadas tamb m algumas bibliotecas para auxiliar no desenvolvimento, sendo as principais, a biblioteca *DIO* para comunica o com servidor via requisicao web, a biblioteca *CAMERA* que permite acessar e controlar a c mera dos dispositivos *Android*, *IOS* e *Web*, e a biblioteca *PATH_PROVIDER* para acesso aos diret rios.

A aplica o foi constru da respeitando as camadas do modelo de arquitetura de *software* MVC como demonstrado de se o 2 deste artigo. Foram criados *widgets* personalizados reutilizando *widgets* j  existentes na biblioteca *material* nativa do *framework*, com intuito de explorar a facilidade de criar e personalizar componentes visuais nesta ferramenta.

A aplica o resultou na constru o das seguintes telas: tela de *login*, que permite ao usu rio acessar a aplica o digitando o e-mail e a senha; tela de cadastro, onde o usu rio dever  informar seu endere o e informa es de contato como n mero de celular e e-mail; e tela de perfil, onde o usu rio pode rever as informa es previamente cadastradas.

Figura 6 – Telas do aplicativo



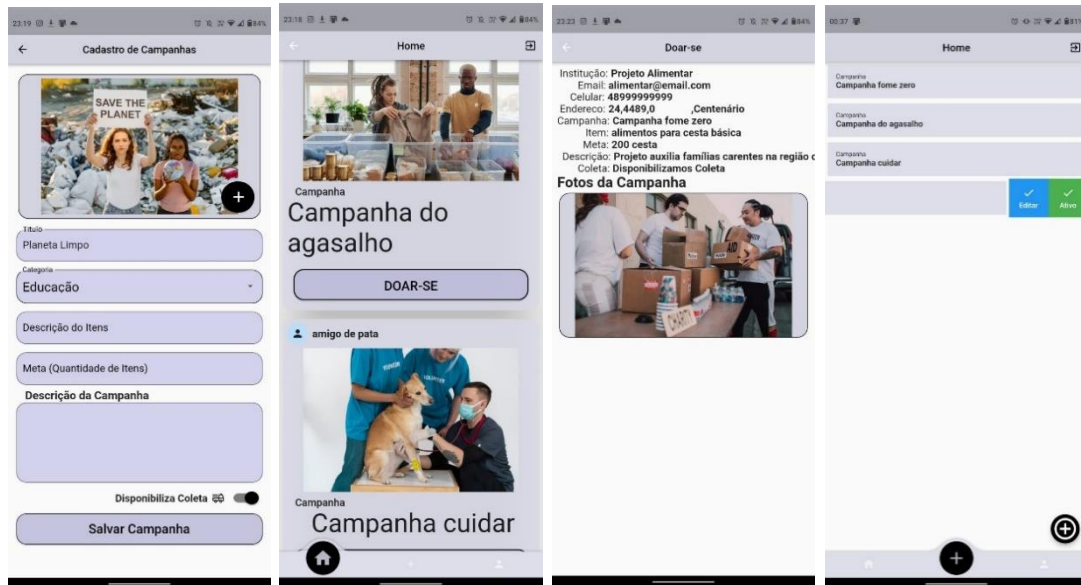
Fonte: Do autor

As demais telas são referentes às campanhas: cadastro de campanhas, listas de campanhas, ativação ou edição das campanhas e detalhamento.

O fluxo da aplicação inicia pela tela de *login*, para que novos usuários criem uma conta a partir do link “cadastre-se”. Em seguida o usuário é direcionado para a tela de cadastro, possibilitando criar uma conta com voluntário ou instituição. Caso já possua uma conta, as credenciais e-mail e senha são validados e redirecionam o usuário para tela *home*, que dá acesso a três outras principais telas. Uma delas traz a lista de todas as campanhas cadastradas, mostra as fotos das campanhas, o título e um botão para doar-se. Este botão detalha as informações da companhia, como o nome da instituição, o *e-mail* e o número para contato, descrição e metas da companhia.

Outra tela disponibilizada a partir da tela *home*, é a de cadastro de campanhas. Nesta tela são criadas todas as campanhas e suas respectivas informações, como fotos, categoria, itens, metas e descrições. Por último, temos a tela de perfil, onde o usuário consegue visualizar suas informações cadastrais como nome, *e-mail*, celular e endereço.

Figura 7 – Telas do aplicativo



Fonte: Do autor

3 RESULTADOS E DISCUSSÃO

Os resultados obtidos a partir da exploração das características e funcionalidades do *framework* Flutter, está na facilidade e instalação de configuração desta ferramenta; uma variedade de *widgets* nativos e de terceiros que auxiliam na construção da aplicação; a funcionalidade *hot reloading*, que permite acompanhar as modificações automaticamente em tempo real sem a necessidade de compilar o aplicativo novamente; e apesar de utilizar o *Dart*, que é uma linguagem de programação pouco conhecida, apresentou uma curva de aprendizado rápido.

A escolha do *framework* Flutter se mostrou eficiente na construção da aplicação multiplataforma, com os testes realizados nos dispositivos *mobile* e *web* foram executados a partir do mesmo código fonte, de acordo com Rissi & Dallilo (2022), o Flutter é um *framework* excelente em relação à sua proposta, permitindo um desenvolvimento único para múltiplas plataformas, que ajusta ao ambiente de acordo com a experiência do usuário final, além de apresentar uma integração fácil com diversas ferramentas.

No entanto foram encontradas algumas limitações, principalmente quando se trata de recursos nativos das plataformas como acesso à câmera, *gps* e locais de arquivos que são específicos das plataformas, embora existam alguns *plugins* e pacotes disponíveis para suprir essa limitação, nem sempre eles conseguem cobrir todas as funcionalidades que o desenvolvedor necessita.

A escolha de construir um serviço *back-end* na plataforma Node.js, resultou em um esforço dobrado no desenvolvimento do aplicativo, considerando que foram criadas aplicações distintas onde qualquer incremento resultava na manutenção de ambas, além de própria manutenção da base de dados, soluções como plataforma de aplicativos Firebase, teriam abstraído a construção do *back-end*.

No artigo “desenvolvimento mobile utilizando flutter e hasura”, desenvolvido por Silva & Rollwagen (2021), conclui-se que o Flutter é um *framework* eficiente e de fácil aprendizado, onde apesar de outras ferramentas utilizarem linguagens de programação mais conhecidas, o Dart mostrou-se muito eficiente, de fácil desenvolvimento e aprendizado. A estrutura de *widgets* torna mais prático o desenvolvimento orientado aos objetos e ao aproveitamento de código fonte, porém, pelo fato de ser uma tecnologia recente, esta sofre constante atualização, conforme algumas bibliotecas são atualizadas, resultando em *bugs* e uma comunidade pequena de desenvolvedores.

Rissi & Dallilo (2022) fizeram algumas comparações entre o *framework* Flutter e o React Native, e consideraram o React Native como tendo uma sintaxe mais simples e um código fonte de melhor leitura, porém para aplicações mais robustas e que necessitem de programação nativa foi indicada a utilização do Flutter. A vantagem de não ficar criando um código fonte para cada ambiente, sobrepõe as dificuldades de sintaxe validando o uso do *framework* como positivo.

A aplicação desenvolvida possibilitou a criação de contas para voluntários e instituição, após a criação da conta ou login do usuário, é disponibilizada uma lista com todas as campanhas já existentes, o cadastro e modificação das campanhas também foi disponibilizado pela aplicação com possibilidade de inserir diversas imagens, escolher categorias, dar um título e uma descrição para campanha. Assim que um usuário encontra uma campanha para doar-se ele terá acesso a todas as informações necessárias para entrar em contato com a instituição.

4 CONCLUSÃO

O desenvolvimento de uma aplicação voltada ao terceiro setor, utilizando o *framework* Flutter, mostrou-se eficaz na exploração e identificação das funcionalidades do *framework*, como a integração com diversas plataformas com apenas um código fonte. A estrutura do projeto baseado na arquitetura MVC contribui muito para a organização do programa, separando as responsabilidades em camadas.

A elaboração do aplicativo disponibilizou a criação de campanhas para arrecadação de serviços e materiais, além de informar sobre os resultados e objetivos de cada companhia.

Vivemos um cenário onde cada vez mais as pessoas buscam por soluções práticas, levando a uma constante evolução de tecnologias e informações. Neste sentido, a construção de aplicações capazes de aproximar pessoas dispostas a ajudar e pessoas que precisam de ajuda, contribui para a formação de uma sociedade mais igualitária, proporcionando novas oportunidades e projetos de vida para muitas pessoas. Cada novo projeto desenvolvido é uma tentativa de transformar a realidade de alguém e, quem sabe, fazer seu mundo melhor.

Com trabalhos futuros pode-se dar continuidade ao desenvolvimento da aplicação, como implementar a integração da base de dados do *firebase*, melhorar a *interface* gráfica, implantar relatórios e monitoramento de recursos e validar a usabilidade juntamente com instituições sociais e voluntários.

REFERÊNCIAS

ANDRADES, Júlio Ferreira de; MACIEL, Estélvia Rosandra Portilio; CANSI, Francine. Relação da fome com a violência: uma proposta para a proteção da criança e adolescente. **Brazilian Journal of Development**, Curitiba, v. 7, n. 1, p. 11071-11083, jan. 2021. Disponível em: <<https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/23960/19226%3E>>. Acesso em: mar. 2022.

ARAÚJO, Matheus Felipe Gonçalves Oliveira de. **Solução de Software para Sistema de Doações**. Trabalho de Conclusão de Curso (Sistemas de Informação) - Universidade Federal de Uberlândia. Uberlândia, 2021. Disponível em:

<<https://repositorio.ufu.br/bitstream/123456789/31837/4/Solu%c3%a7%c3%a3oSoftwareSistema.pdf>>. Acesso em: jan. 2023.

COSTA, Simone da Silva. Pandemia e desemprego no Brasil. **Revista de Administração Pública**, v. 54, n. 4, p. 969-978, jul./ago. 2020. Disponível em: <<https://www.scielo.br/j/rap/a/SGWCFyFzjzrDwgDJYKcdhNt/?format=pdf&lang=pt>>. Acesso em: mai. 2022.

LIMA, Edinício Oliveira; STETTINER, Caio Flávio; FERREIRA JÚNIOR, Samuel. Balanço social e o “full disclosure” no terceiro setor. **Revista de tecnologia aplicada (RTA)**, v. 10, n. 1, p. 23-39, jan./abr. 2021. Disponível em: <<https://www.cc.faccamp.br/ojs-2.4.8-2/index.php/RTA/article/view/1734/792>>. Acesso em: mar. 2022.

LIMA, Fernando Fortunato de. **Avaliação de Frameworks para o Desenvolvimento de Aplicações Híbridas**. Trabalho de Conclusão de Curso (Engenharia de Software) - Universidade Federal do Pampa. Alegrete, 2019. Disponível em: <<https://repositorio.unipampa.edu.br/jspui/bitstream/riiu/4224/1/Fernando%20Fortunato%20de%20Lima%20-%202019.pdf>>. Acesso em: jun. 2023.

MACIEL, Walery Luci da Silva. **Projetos sociais universidade do sul de Santa Catarina**. Palhoça, Unisul Virtual, 2015. Disponível em: <https://www.uaberta.unisul.br/repositorio/recurso/14690/pdf/projetos_sociais.pdf>.

MENDONÇA, Érika de Sousa; MENEZES, Jaileila de Araújo. Organizações não governamentais como dispositivos de poder: do bem-estar ao controle social. **Fractal: Revista de Psicologia**, v. 29, n. 3, p. 272-279, set./dez. 2017. Disponível em: <<https://periodicos.uff.br/fractal/article/view/5175>>. Acesso em: mar. 2022.

PARANÁ PORTAL. Doação de roupas e cobertores poderá ser feito por aplicativo no Paraná. Online. 24 de junho de 2020. Disponível em: <<https://paranaportal.uol.com.br/cidades/doacao-roupas-cobertores-aplicativo>>. Acesso em: nov. 2022.

PARANÁ SOLIDÁRIO. Secretaria da Justiça e Cidadania, Governo do Paraná. Online. 2020. Disponível em: <<https://www.justica.pr.gov.br/Pagina/Aplicativo-Parana-Solidario>>. Acesso em: nov. 2022.

POBREZA. Nações Unidas. Disponível em: <<https://unric.org/pt/eliminar-a-pobreza/>>.

SILVA, Darlan Michel da; ROLLWAGEN, André Fernando. **Desenvolvimento mobile utilizando flutter e hasura**. Trabalho de Conclusão de Curso (Tecnologia em Sistemas para Internet) - Instituto Federal Sul Rio-grandense. Passo Fundo, 2020. Disponível em:

<<https://painel.passofundo.ifsul.edu.br/uploads/arq/202104201525561269175147.pdf>>. Acesso em: nov. 2022.

RISSI, Matheus; DALLILO, Felipe Diniz. Flutter um framework para desenvolvimento mobile. **Revista científica multidisciplinar**, v. 3, n. 11, 2022. Disponível em: <<https://recima21.com.br/index.php/recima21/article/view/2230/1666>>. Acesso em: nov. 2022.