

PROTÓTIPO DE APLICATIVO PARA ACOMPANHAMENTO E CONTROLE DE GLICEMIA

FURB - UNIVERSIDADE REGIONAL DE BLUMENAU

CURSO DE SISTEMAS DE INFORMAÇÃO

ACADÊMICO: TIAGO DIONESTO WILLRICH DA SILVA

ORIENTADOR: MAURO MARCELO MATTOS



ROTEIRO

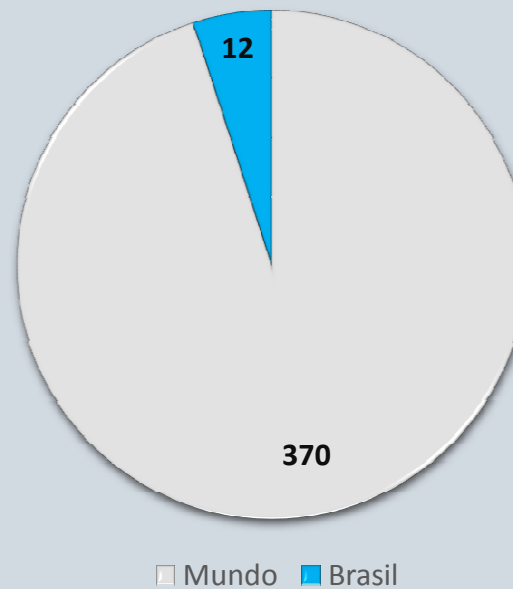
1. Introdução e Objetivos
2. Fundamentação Teórica
3. Especificação
4. Implementação
5. Operacionalidade
6. Resultados e Discussões
7. Conclusão

Introdução e Objetivos

INTRODUÇÃO

- Crescimento no número de diabéticos
- Problemas no tratamento
- Ferramentas disponíveis atualmente

Quantidade de diabéticos (em milhões)



OBJETIVOS

- **Objetivo geral:**

Disponibilizar um aplicativo Android que auxilie o usuário no acompanhamento e controle da glicemia, permitindo que o endocrinologista também acompanhe o tratamento a distância.

- **Objetivos específicos:**

- a) disponibilizar ao paciente recurso para registro de glicemias e de administração de medicamentos periódicos;
- b) disponibilizar ao endocrinologista recurso para visualização das informações fornecidas por seus pacientes;
- c) disponibilizar um servidor para sincronia de dados entre paciente e endocrinologista.

Fundamentação Teórica

APLICATIVOS MÓVEIS

- Definição
- Mercado
 - Brasil: 38,8 milhões de pessoas com *smartphones*
 - Mundo: 192 bilhões de *downloads* em 2014
- Disponibilização
 - Google Play
 - AppStore

MOBILE HEALTH

- Definição
- Mercado
 - AppStore: 4 mil mHealth *apps* em 2010 para 15 mil em 2011
- Brasil possui o 3º maior sistema de saúde do mundo
 - 7 mil hospitais
 - 25 mil laboratórios
 - 17 mil clínicas
 - 125 mil consultórios médicos

DIABETES

- Definição
- Tipos
- Consequências
- Tratamento



Símbolo da Diabetes

CONTROLE GLICÊMICO

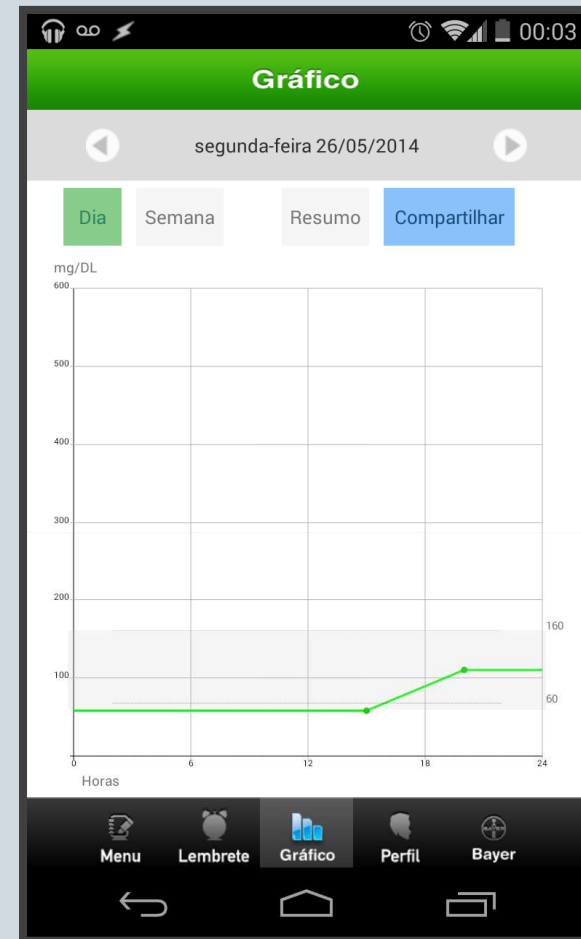
- Definição
- Rotina
- Ferramentas



TRABALHO S CORRELAT OS

GlicoCare

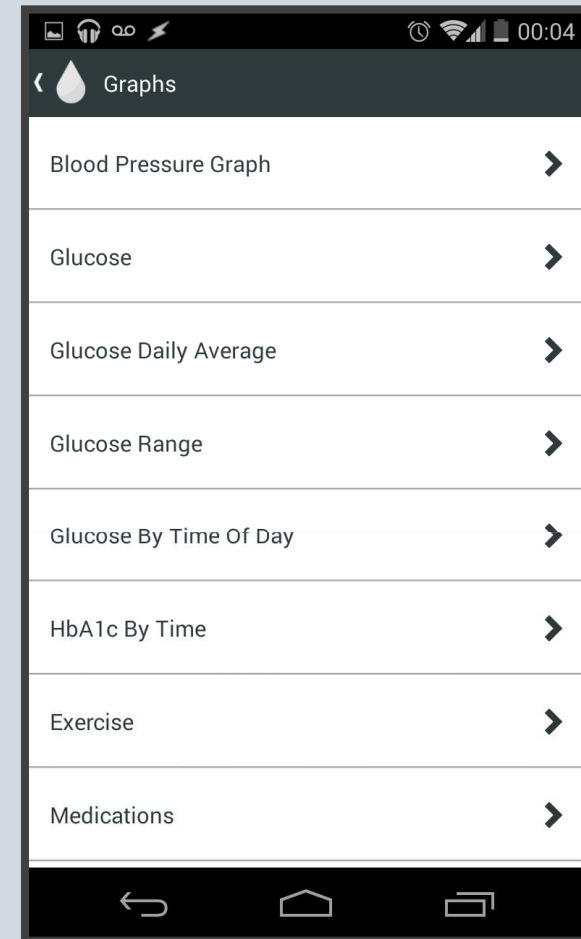
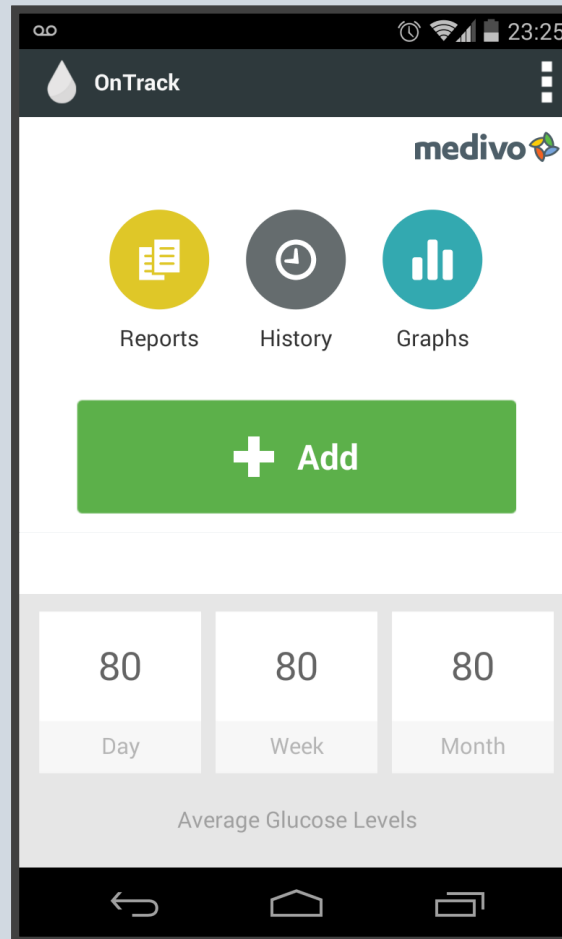
Bayer



TRABALHO S CORRELAT OS

OnTrack

Medivo



TRABALHO S CORRELAT OS

Diabetes
Control

Wíden



Especificação

REQUISITOS FUNCIONAIS

RF01: O sistema deverá permitir ao visitante se cadastrar no sistema.

RF02: O sistema deverá permitir ao usuário efetuar o *login*.

RF03: O sistema deverá permitir ao usuário efetuar o *logoff*.

RF04: O sistema deverá permitir ao paciente configurar um endocrinologista para acompanhar o seu tratamento.

RF05: O sistema deverá permitir ao paciente manter seu histórico de glicemias.

RF06: O sistema deverá permitir ao paciente manter seu histórico de uso de medicamentos.

RF07: O sistema deverá permitir ao paciente configurar itens de rotina.

RF08: O sistema deverá permitir ao paciente a parametrização do intervalo de glicemia desejado para suas medidas.

RF09: O sistema deverá permitir ao paciente e ao endocrinologista conversarem entre si por *chat* com histórico.

REQUISITOS FUNCIONAIS

RF10: O sistema deverá permitir a visualização de gráficos de performance do paciente.

RF11: O sistema deverá permitir a geração de *e-mails* contendo os gráficos de performance.

RF12: O sistema deverá permitir ao endocrinologista visualizar a lista de seus pacientes.

RF13: O sistema deverá permitir ao endocrinologista visualizar as informações de um paciente.

RF14: O sistema deverá permitir ao endocrinologista a remoção de um paciente da sua lista de acompanhamento.

RF15: O sistema deverá permitir ao endocrinologista visualizar os históricos de seus pacientes.

RF16: O sistema deverá permitir ao endocrinologista visualizar a rotina de seus pacientes.

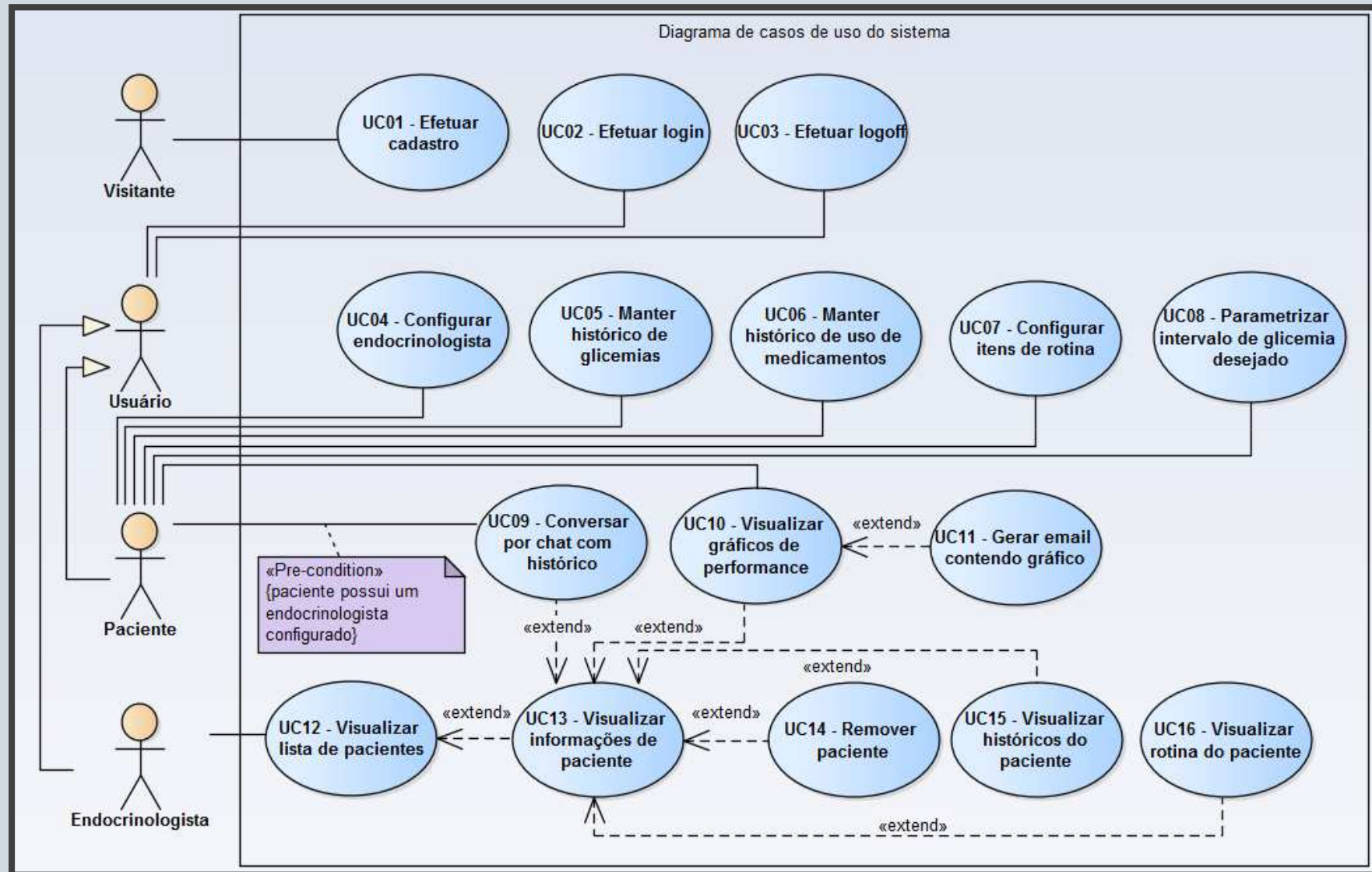
REQUISITOS NAO FUNCIONAIS

RNF01: O aplicativo deverá ser desenvolvido em Java, para a plataforma Android.

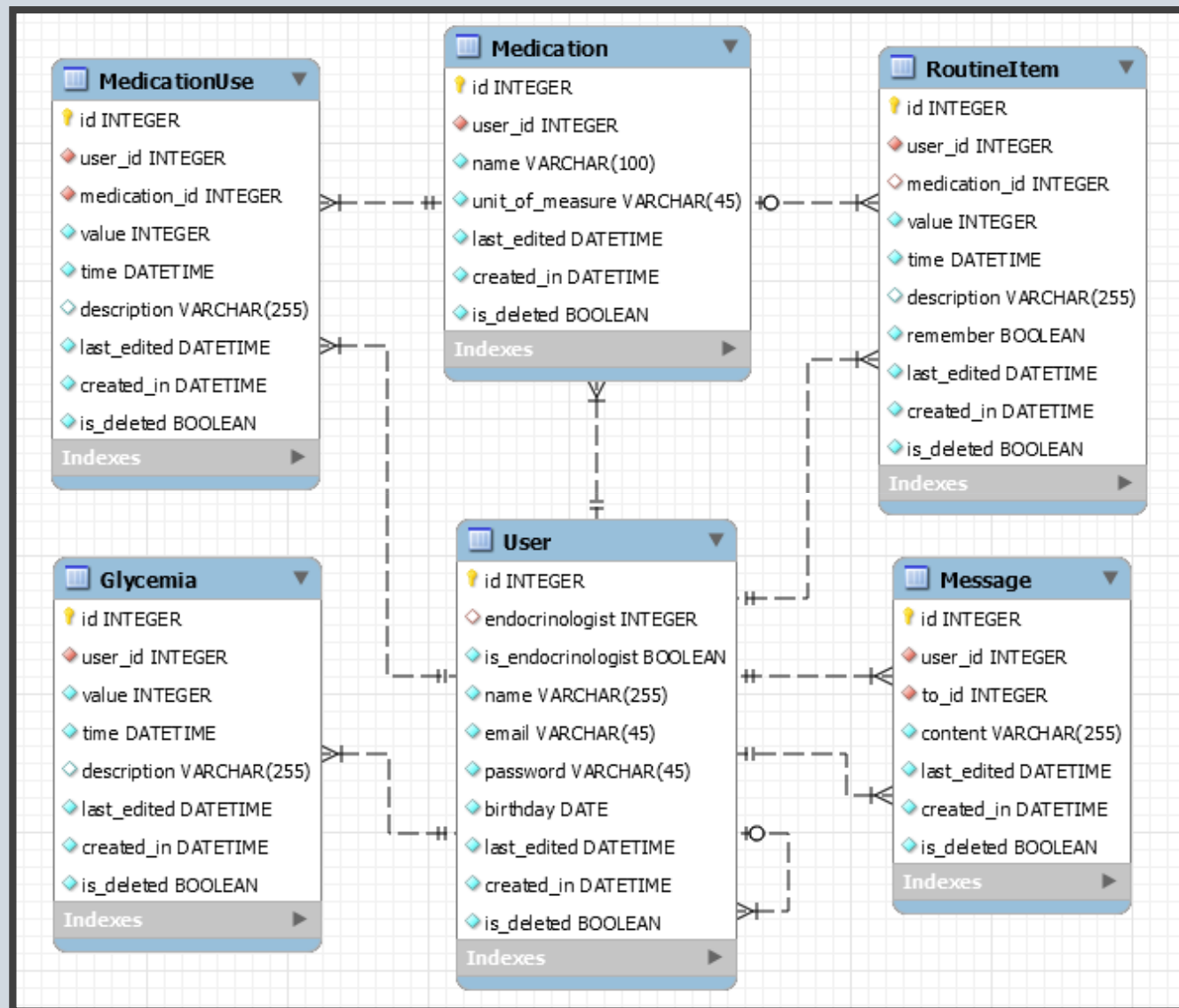
RNF02: O aplicativo deverá utilizar o banco de dados SQLite.

RNF03: O servidor deverá utilizar o banco de dados MySQL.

Diagrama de Casos de Uso



MER



Implementação

TECNICAS E FERRAMENTAS

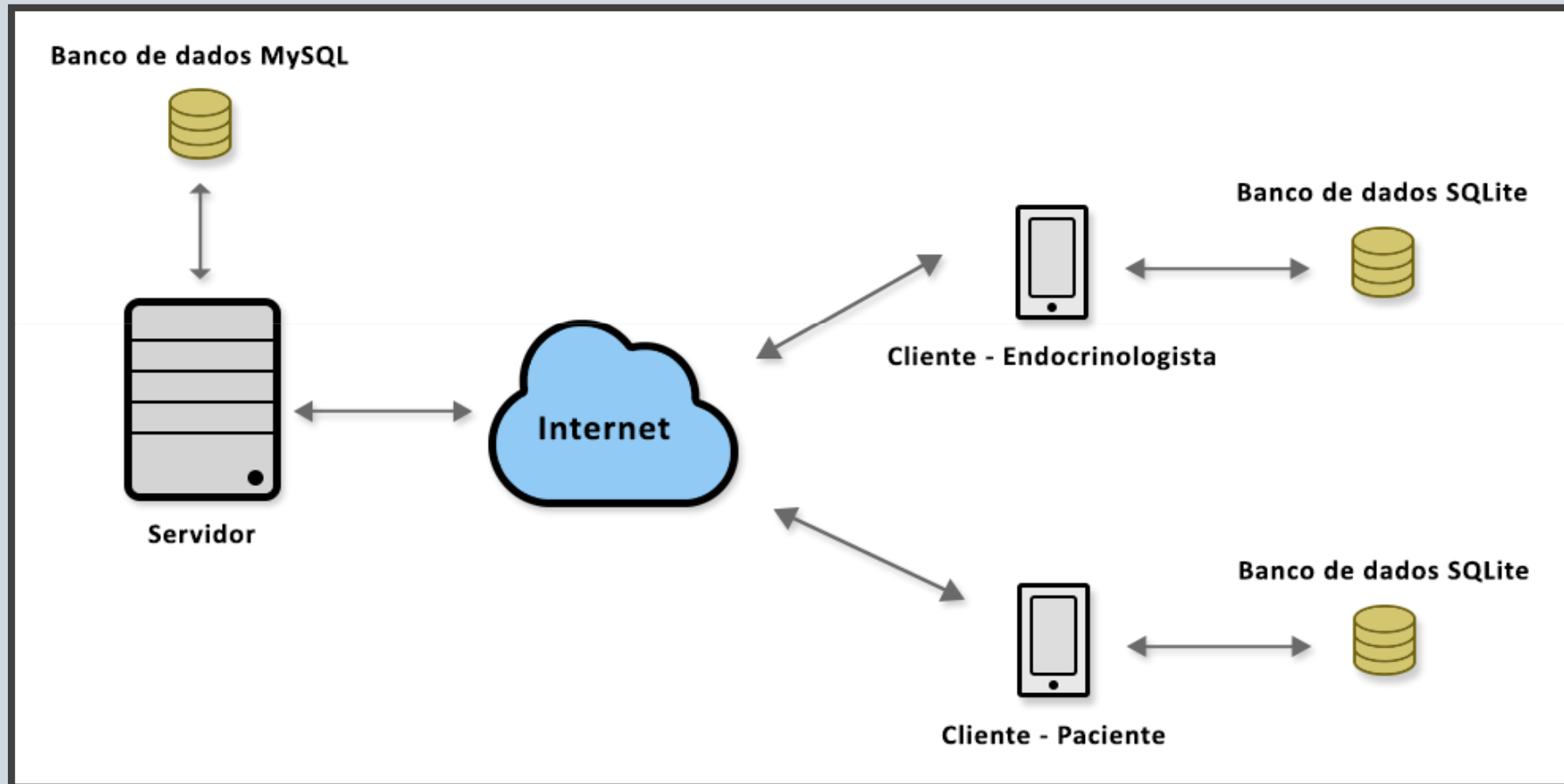
APLICATIVO

- Eclipse
- Java
- Android Developer Tools (ADT)
- Gson
- MPAndroidChart
- Joda Time
- SQLite

SERVIDOR

- Eclipse
- Java
- Jersey
- MySQL

ARQUITETURA



Persistência

- Classe Modelo
- Classe Info
- Classe DAO

```
public class DaoUser extends Dao {  
  
    /* Constructors */  
  
    public DaoUser(Context context) {  
        super(context);  
    }  
  
    /* Methods */  
  
    public long insert(User user) {  
        long time = System.currentTimeMillis();  
        ContentValues values = new ContentValues();  
        values.put(UserTable.COLUMN_ID, user.getId());  
        values.put(UserTable.COLUMN_ENDOCRINOLOGIST, user.getEndoc  
            : null);  
        values.put(UserTable.COLUMN_IS_ENDOCRINOLOGIST, user.getIs  
        values.put(UserTable.COLUMN_NAME, user.getName());  
        values.put(UserTable.COLUMN_EMAIL, user.getEmail());  
        values.put(UserTable.COLUMN_PASSWORD, user.getPassword());  
        values.put(UserTable.COLUMN_BIRTHDAY, user.getBirthday().g  
        values.put(UserTable.COLUMN_LAST_EDITED, time);  
        values.put(UserTable.COLUMN_CREATED_IN, time);  
        values.put(UserTable.COLUMN_IS_DELETED, false);  
        return insert(UserTable.TABLE_NAME, values);  
    }  
  
    public User user(Cursor c) {  
  
    public User get(long id) {  
  
    public Cursor getAll() {  
  
    public long insertFromServer(User user) {  
  
    }
```

Interface

- XML
- Activities

```
public class RegistrationActivity extends FragmentActivity implements
    private final static String DATE_FORMAT = "dd/MM/yyyy";
    private Calendar cal;
    private EditText edtName, edtEmail, edtPassword, edtDate;
    private CheckBox cbIsEndocrinologist;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);

        Toolbar mToolbar = (Toolbar) findViewById(R.id.toolbar);
        setActionBar(mToolbar);
        getActionBar().setDisplayHomeAsUpEnabled(true);

        cal = Calendar.getInstance();
        getFields();
        edtDate.setOnClickListener(this);
    }

    private void getFields() {
        edtName = (EditText) findViewById(R.id.input_name);
        edtEmail = (EditText) findViewById(R.id.input_email);
        edtPassword = (EditText) findViewById(R.id.input_password);
        edtDate = (EditText) findViewById(R.id.input_date);
        cbIsEndocrinologist = (CheckBox) findViewById(R.id.input_

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.cancel, menu);
        getMenuInflater().inflate(R.menu.save, menu);
        return true;
    }
```


Comunicação com servidor

- Requisições
- Recursos
- JSON

```
@Path("/glycemia")
@Produces(MediaType.APPLICATION_JSON)
public class GlycemiaResource {
    @GET
    @Path("/listAllFromUser/{userid}")
    public ArrayList<Glycemia> listAllFromUser(@PathParam("userid")
        return GlycemiaDAO.getInstance().listAllFromUser(userid);
    }

    @GET
    @Path("/{id}")
    public Glycemia getGlycemia(@PathParam("id") long id) {
        return GlycemiaDAO.getInstance().get(id);
    }

    @POST
    @Path("/save")
    @Consumes(MediaType.APPLICATION_JSON)
    public Response saveGlycemia(String json) {
        GlycemiaDAO dao = GlycemiaDAO.getInstance();
        Glycemia obj;
        try {
            obj = JsonUtil.getDefaultMapper().readValue(json, Glycemia.class);
            Long objId = dao.save(obj);
            if (objId != null) {
                return Response.status(200).entity(dao.get(objId)).build();
            } else {
                return Response.status(400).build();
            }
        } catch (Exception e) {
            e.printStackTrace();
            return Response.status(500).build();
        }
    }
}
```

Sincronizaçã

- ServerId
- LastEdited

```
private void syncGlycemias(long id) {
    Class<Glycemia> clazz = Glycemia.class;
    DaoGlycemia dao = new DaoGlycemia(context);
    Cursor c = dao.getAll(id);

    ArrayList<Glycemia> objectsServer = rest.getObjectsFromUser(id, Glycemia.class);

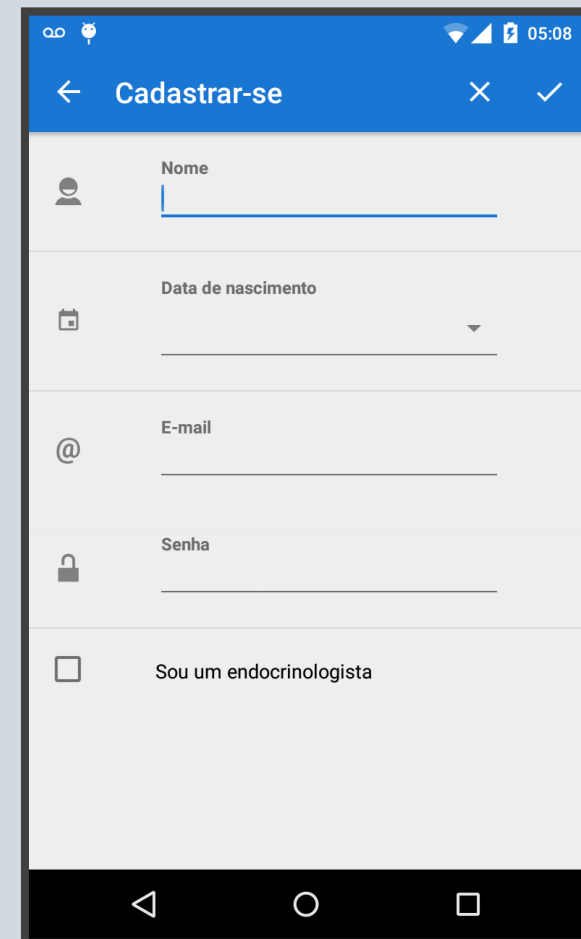
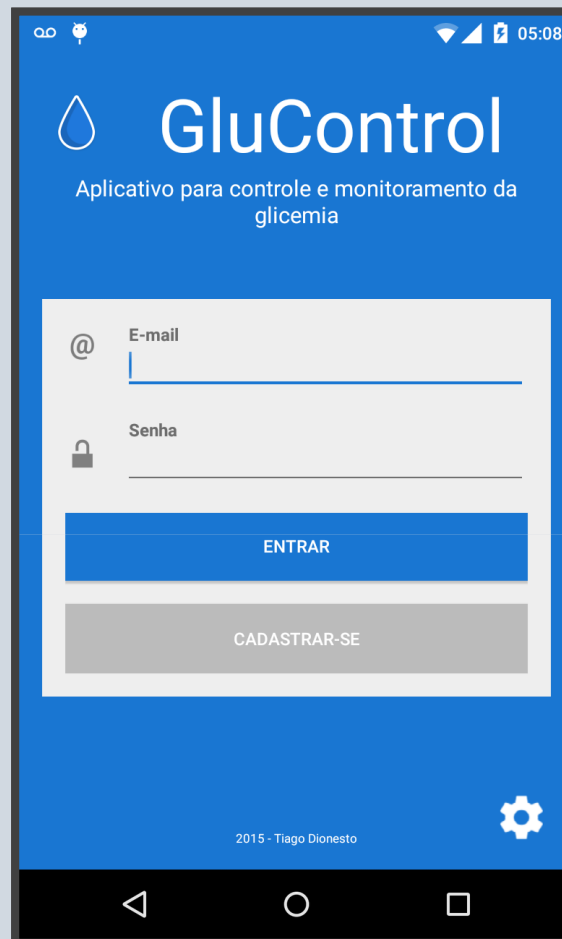
    if (objectsServer != null) {
        for (Glycemia i : objectsServer) {
            Glycemia obj = dao.getByServerId(i.getServerId());
            if (obj == null) {
                Log.i(TAG, "Inserting server glycemia id=" + i.getServerId() + " i
                dao.insertFromServer(i);
            } else {
                DateTime dt1 = new DateTime(obj.getLastEdited().getTime());
                DateTime dt2 = new DateTime(i.getLastEdited().getTime());
                int dif = dt1.compareTo(dt2);
                if (dif < 0) {
                    Log.i(TAG, "Updating local glycemia id=" + obj.getServerId() +
                    + obj.getLastEdited() + ". Server last edited: " + i.g
                    i.setId(obj.getId());
                    dao.updateFromServer(i);
                } else if (dif > 0) {
                    Log.i(TAG, "Updating server glycemia id=" + i.getServerId() +
                    + obj.getLastEdited() + ". Server last edited: " + i.g
                    rest.saveObject(obj, clazz);
                }
            }
        }
    }

    if (c != null) {
        while (c.moveToNext()) {
            Glycemia obj = dao.glycemiaForAdapter(c);
            if (obj.getServerId() == null) {
                Log.i(TAG, "Inserting local glycemia into server");
                Glycemia t = rest.saveObject(obj, clazz);
                t.setId(obj.getId());
                dao.updateFromServer(t);
            }
        }
    }
}
```

Operacionalidad e

PRINCIPAIS TELAS

Login e
Cadastro



PRINCIPAIS TELAS

Glicemias

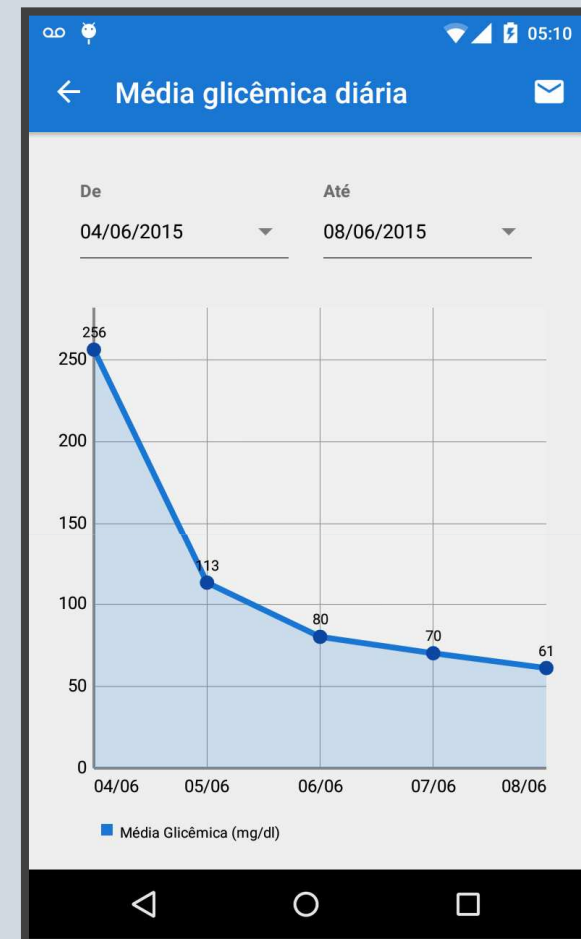
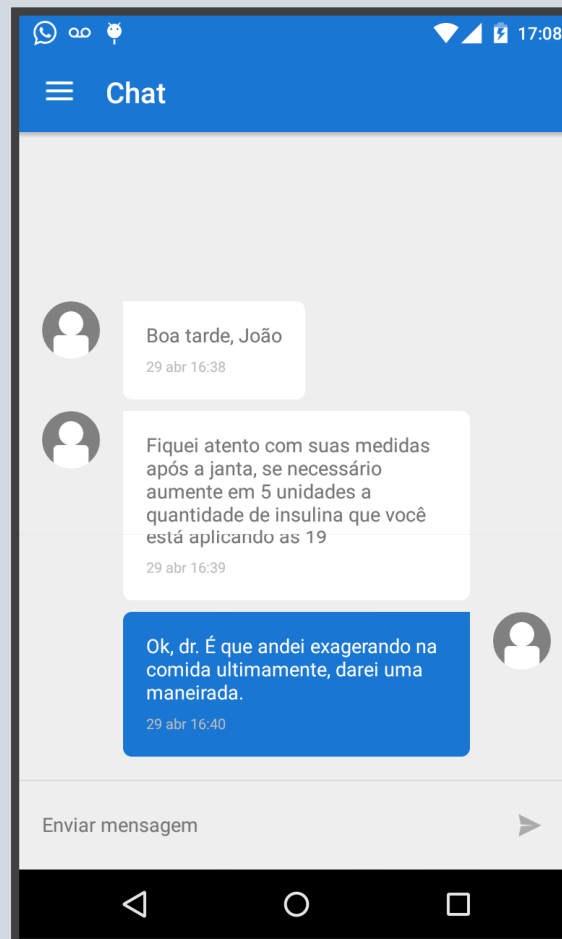
The screenshot shows the main interface of the 'Glicemias' app. At the top, there is a blue header with the title 'Glicemias' and a menu icon. Below the header, there are three date tabs: '28/04/2015', '29/04/2015', and '30/04/2015'. The main content area displays a list of six blood sugar readings, each with a water drop icon, a value in mg/dl, a description, and a time. A blue circular button with a white plus sign is located at the bottom right of the list.

Valor (mg/dl)	Descrição	Tempo	Variação
140	Antes de dormir	23:22	+20.0
60	Antes da janta	19:20	-75.0
145	Nenhuma Descrição	16:10	-
103	Antes do almoço	12:10	+53.0
150	Antes do café	09:00	-
120	Nenhuma Descrição	03:00	-

The screenshot shows the 'Adicionar Glicemia' (Add Blood Sugar) screen. It features a blue header with a back arrow, the title 'Adicionar Glicemia', and a trash icon. The main content area has several input fields: 'Valor' (Value) with a water drop icon and a blue underline, 'Dia' (Day) with a calendar icon and a dropdown arrow, 'Hora' (Hour) with a dropdown arrow, and 'Descrição' (Description) with a document icon and a blue underline. A blue circular button with a white plus sign is located at the bottom right of the screen.

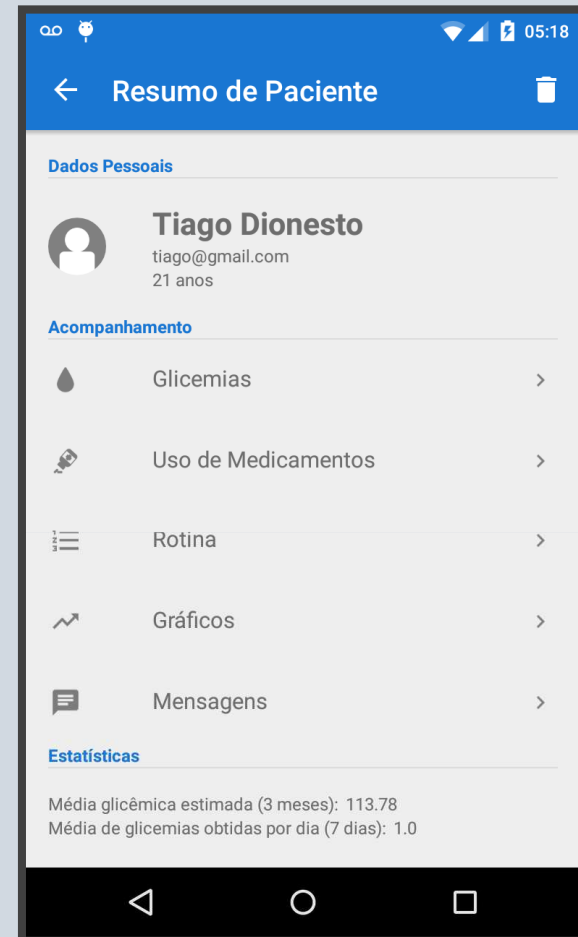
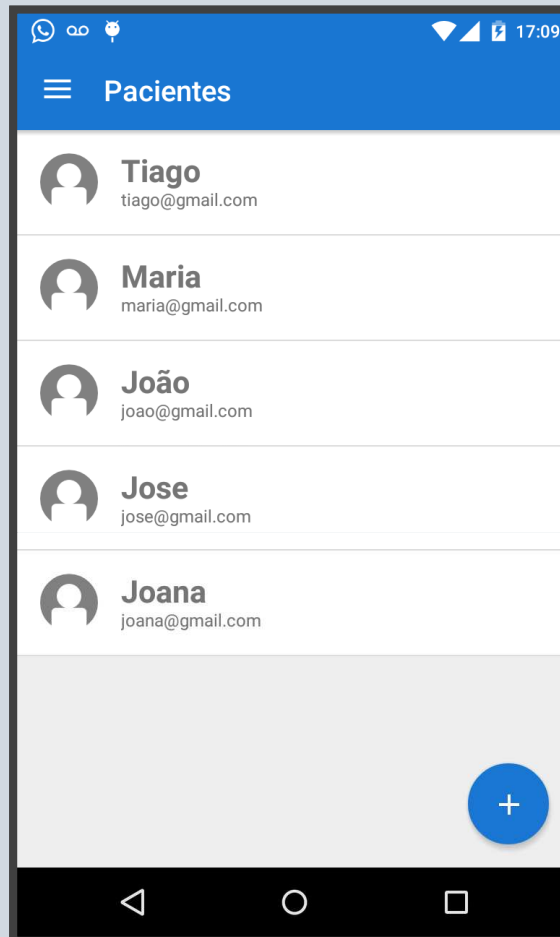
PRINCIPAIS TELAS

Chat e gráficos



PRINCIPAIS TELAS

Pacientes



Resultados e Discussões

Resultados

- Objetivos cumpridos
- Requisitos atendidos

COMPARATIVO

	GlicoCare	OnTrack	Diabetes Control	GluControl
Acompanhamento simultâneo pelo endocrinologista				✓
Armazenamento de dados na nuvem				✓
Configuração de lembretes	✓		✓	✓
Configuração do intervalo de glicemia desejado		✓	✓	✓
Geração de gráficos	✓	✓	✓	✓
Geração de relatórios		✓		
Envio de gráficos por e-mail	✓		✓	✓
Gerenciamento de glicemias	✓	✓	✓	✓
Gerenciamento de medicamentos	✓	✓	✓	✓
Gerenciamento de outras informações (refeições, atividades físicas, etc.)	✓	✓	✓	
Inserção de dados por bluetooth			✓	
Troca de mensagens por chat				✓

Conclusão

EXTENSÕES

- Permitir o gerenciamento de mais informações diárias;
- Mais gráficos, estatísticas e relatórios;
- Sincronização com glicosímetros por *bluetooth*;
- Sincronização com bombas de insulina por *bluetooth*;
- Interface web;
- Disponibilização e configuração de mais notificações;
- Controle de privacidade.

DEMONSTRAÇÃO DA APLICAÇÃO

GLUCONTROL

