SENADO FEDERAL
Secretaria Especial do Interlegis - SINTER
Subsecretaria de Tecnologia da Informação - SSTIN

Produto IV

Sistema de Informações Gerenciais do Interlegis

APO-CASA

Guilherme Mesquita Gondim

Contrato N°: 2008/000471

Sumário

1 Introdução **1**

2 Desenvolvimento web com Django **2**
2.1 Objetivos . . . 2
2.2 Conteúdo programático . . . 2
2.3 Requisitos . . . 2
2.4 Duração do treinamento . . . 2

3 Estendendo o SIGI **3**
3.1 Objetivos . . . 3
3.2 Conteúdo programático . . . 3
3.3 Requisitos . . . 3
3.4 Duração do treinamento . . . 3

4 Utilizando o SIGI **4**
4.1 Objetivos . . . 4
4.2 Conteúdo programático . . . 4
4.3 Requisitos . . . 4
4.4 Duração do treinamento . . . 4

5 Bibliografia **5**

6 Anexos **6**
6.1 Visão Geral . . . 6
6.2 Organização do sistema . . . 7
6.2.1 Listagem do conteúdo dos diretórios . . . 7
6.2.2 Descrição dos arquivos e diretórios . . . 9
6.2.3 Descrição das aplicações . . . 10
6.2.4 Relacionamento entre as aplicações . . . 10
6.3 Casos de Uso . . . 10
6.3.1 Casos de Uso do SIGI . . . 10
6.4 Modelo de dados . . . 13
6.4.1 Aplicação sigi.apps.casas (figura 3) . . . 13
6.4.2 Aplicação sigi.apps.contatos (figura 4) . . . 14
6.4.3 Aplicação sigi.apps.convenios (figura 5) . . . 14
6.4.4 Aplicação sigi.apps.inventario (figura 6) . . . 14
6.4.5 Aplicação sigi.apps.mesas (figura 7) . . . 14
6.4.6 Aplicação sigi.apps.parlamentares (figura 8) . . . 14
6.4.7 Aplicação sigi.apps.servicos (figura 9) . . . 14
6.5 Esquema de dados . . . 14
6.5.1 Aplicação sigi.apps.casas . . . 14
6.5.2 Aplicação sigi.apps.contatos . . . 15
6.5.3 Aplicação sigi.apps.convenios . . . 19

6.5.4 Aplicação sigi.apps.inventario 19
6.5.5 Aplicação sigi.apps.mesas 20
6.5.6 Aplicação sigi.apps.parlamentares 22
6.5.7 Aplicação sigi.apps.servicos 22
6.6 Manual de Instalação 23
6.6.1 Preparação do ambiente para a instalação 23
6.6.2 Instalação e configuração do SIGI 24

Lista de Figuras

1 Relacionamento entre as aplicações 11
2 Casos de Uso 12
3 Diagrama de Classes da aplicação sigi.apps.casas 13
4 Diagrama de Classes da aplicação sigi.apps.contatos 14
5 Diagrama de Classes da aplicação sigi.apps.convenios 15
6 Diagrama de Classes da aplicação sigi.apps.inventario 16
7 Diagrama de Classes da aplicação sigi.apps.mesas 17
8 Diagrama de Classes da aplicação sigi.apps.parlamentares . . . 18
9 Diagrama de Classes da aplicação sigi.apps.servicos 18

1 Introdução

Este documento detalha as atividades desenvolvidas durante a quarta etapa do projeto de desenvolvimento do Sistema de Informações Gerenciais do Interlegis (SIGI).

A quarta etapa consiste nos treinamentos de utilização da tecnologia "Django Web Framework", aplicada no desenvolvimento do SIGI, e sobre o uso do SIGI.

As seções 2, 3 e 4 apresentam, respectivamente, os objetivos e conteúdo programático do treinamento em Django, desenvolvimento para o SIGI e uso do SIGI.

A seção 5 possui a bibliografia necessária para o treinamento em desenvolvimento web com Django e do SIGI.

Na seção 6, estão anexadas a documentação técnica produzida para o SIGI.

2 Desenvolvimento web com Django

2.1 Objetivos

O treinamento visa capacitar desenvolvedores web no uso do framework Django para a construção de sistemas online.

2.2 Conteúdo programático

1. Introdução
2. Conceitos básicos
   (a) Model-view-controller (MVC)
   (b) Model-template-view (MTV)
3. Desenvolvimento de uma aplicação
   (a) Models
   (b) Views
   (c) Templates
4. Projetos x Aplicações
   (a) Organização de projetos
   (b) Organização de aplicações

2.3 Requisitos

Para o treinamento, o aluno deverá possuir conhecimentos básicos em desenvolvimento web e Python.

2.4 Duração do treinamento

O treinamento tem duração estimada em 4 horas.

3 Estendendo o SIGI

3.1 Objetivos

O treinamento visa capacitar a equipe de desenvolvimento de software do Interlegis a manter e estender o SIGI.

3.2 Conteúdo programático

1. Aplicação Admin do Django
   (a) CSS
   (b) Templates
   (c) Arquivo sites.py
2. Modelo de dados
3. Configurações locais
4. Criando novas aplicações

3.3 Requisitos

Para o treinamento, o aluno deverá possuir conhecimentos básicos em desenvolvimento web com Python/Django.

3.4 Duração do treinamento

O treinamento tem duração estimada em 2 hora.

4 Utilizando o SIGI

4.1 Objetivos

O treinamento visa capacitar os funcionários do Interlegis a utilizar o SIGI aproveitando todos os seus recursos.

4.2 Conteúdo programático

1. Dashboard
2. Aplicações
   (a) Pesquisa
   (b) Adição/Edição/Atualização
3. Histórico de atividades
4. Relatórios
5. Gerenciamento de usuários

4.3 Requisitos

Para o treinamento, o aluno deverá possuir conhecimentos básicos em Internet e navegação web.

4.4 Duração do treinamento

O treinamento tem duração estimada em 1 hora.

5 Bibliografia

- Documentação Oficial do Django 1.0
- Django Book

6 Anexos

6.1 Visão Geral

O **SIGI** é um projeto para um Sistema de Informações Gerenciais do Interlegis, escrito na linguagem de programação Python com o framework para desenvolvimento web Django.

Página do projeto: http://colab.interlegis.gov.br/wiki/ProjetoSigi

Características

Lista das principais características do SIGI:

- Serviço web cliente/servidor, podendo ser disponibilizado tanto na internet quanto na intranet;
- Multi-plataforma;
- Baseia-se na interface de administração nativa do Django (`django.contrib.admin`, maiores informações em http://docs.djangoproject.com/en/dev/ref/contrib/admin/);
- Gerencia convênios, equipamentos e inventários, serviços prestados e composição de Mesas Diretoras das Casas Legislativas;
- Autenticação no sistema baseada em usuários e grupos, com perfis diferentes;
- Emissão de relatórios.

Requisitos básicos de software

- Python >= 2.4 < 3.0
- Django 1.0

Licença de uso

O SIGI é disponibilizado como software livre, isto significa que você pode redistribuí-lo e/ou modifica-lo dentro dos termos da Licença Pública Geral GNU (GPL) como publicada pela Fundação do Software Livre (FSF); na versão 3 da Licença, ou (na sua opinião) em qualquer versão mais recente.

Veja o arquivo `LEIA-ME` para maiores informações a respeito das condições de cópia e redistribuição.

6.2 Organização do sistema

6.2.1 Listagem do conteúdo dos diretórios

```
.
|-- COPYING
|-- LEIA-ME -> README
|-- README
|-- devel.db
|-- docs
|   |-- arquivos
|   |   |-- casosdeuso.dia
|   |   |-- esquema.sql
|   |   `-- models.png
|   |-- instalacao.txt
|   |-- relatorios
|   |   `-- [...]
|   `-- visaogeral.txt
|-- etc
|   |-- apache
|   |   |-- apache.conf
|   |   `-- django.wsgi
|   `-- patchs
|       `-- django-maintenancemode
|           `-- paths_with_access_allowed.r10.patch
|-- media
|   |-- css
|   |   `-- base_site.css
|   |-- images
|   |   |-- default-bg.gif
|   |   `-- logo-interlegis.png
|   `-- js
|       |-- jquery-1.2.6.js
|       `-- jquery-1.2.6.min.js
`-- sigi
    |-- __init__.py
    |-- admin
    |   |-- __init__.py
    |   `-- filterspecs.py
    |-- apps
    |   |-- __init__.py
    |   |-- casas
    |   |   |-- __init__.py
    |   |   |-- admin.py
    |   |   |-- forms.py
```

```
|   |   |-- models.py
|   |   `-- views.py
|   |-- contatos
|   |   |-- __init__.py
|   |   |-- admin.py
|   |   `-- models.py
|   |-- convenios
|   |   |-- __init__.py
|   |   |-- admin.py
|   |   `-- models.py
|   |-- inventario
|   |   |-- __init__.py
|   |   |-- admin.py
|   |   `-- models.py
|   |-- mesas
|   |   |-- __init__.py
|   |   |-- admin.py
|   |   `-- models.py
|   |-- parlamentares
|   |   |-- __init__.py
|   |   |-- admin.py
|   |   `-- models.py
|   `-- servicos
|       |-- __init__.py
|       |-- admin.py
|       `-- models.py
|-- local_settings.template
|-- locale
|   `-- pt_BR
|       `-- LC_MESSAGES
|           |-- django.mo
|           `-- django.po
|-- manage.py
|-- settings.py
|-- sites.py
|-- templates
|   |-- 503.html
|   |-- admin
|   |   `-- base_site.html
|   |-- app_index.html
|   |-- index.html
|   |-- login.html
|   `-- snippets
|       `-- modules
```

```
|           |-- actions.html
|           `-- user.html
`-- urls.py
```

6.2.2 Descrição dos arquivos e diretórios

COPYING: Arquivo com a licença do sistema (GPLv3).

LEIA-ME/README: Contém informações do sistema, nota de copyright e instruções de instalação.

devel.db: Base de dados SQLite utilizada para desenvolvimento do sistema.

docs/: Diretório com toda documentação escrita para o sistema (relatórios, manual de instalação, casos de uso, esquema de base de dados, etc).

etc/: Diretório com arquivos variados: configuração do Apache, *patchs* de códigos, etc.

media/: Diretório com arquivos estáticos e de mídia do sistema (imagens, CSS, javascripts).

sigi/: Pacote Python do sistema.

sigi/admin/: Pacote de código relacionado com a aplicação `admin` do Django (`django.contrib.admin`).

sigi/apps/: Pacote de aplicações do sistema, maiores informações na Seção 6.2.3.

sigi/settings.py: Configurações padrões do sistema.

sigi/local_settings.template: Exemplo de configurações locais do sistema (pertinentes à instalação). Deverá ser copiado para `sigi/local_settings.py` para sua utilização.

sigi/locale/: Diretório com localização local do projeto.

sigi/manage.py: Script de gerenciamento do projeto, gerado pelo framework Django.

sigi/templates/: Diretório com templates HTML do Django.

sigi/urls.py: Modulo de configuração de URLs.

6.2.3 Descrição das aplicações

O SIGI é composto de algumas aplicações Django, cada uma com um propósito bem definido.

Uma aplicação Django é um pacote Python modular, podendo ser reaproveitado em outros projetos.

Algumas aplicações possuem algum nível de relacionamento com as outras.

Segue descrição de cada aplicação:

sigi.apps.casas: Gerência de Casas Legislativas.

sigi.apps.contatos: Gerência de Contatos do Interlegis com Casas Legislativas, fornecedores de equipamentos, serviços e etc.

sigi.apps.convenios: Convênios do Interlegis com as Casas Legislativas.

sigi.apps.inventario: Inventário de equipamentos disponibilizados pelo Interlegis para as Casas Legislativas.

sigi.apps.mesas: Composição das Mesas Diretoras das Casas Legislativas.

sigi.apps.parlamentares: Gerência de Parlamentares.

sigi.apps.servicos: Serviços prestados às Casas Legislativas conveniadas ao Interlegis.

6.2.4 Relacionamento entre as aplicações

A Figura 1 demonstra o relacionamento entre as aplicações e seus *models*.

Uma seta direcional representa uma relação *muitos para um*. Uma seta bidirecional representa uma relação *muitos para muitos*.

A seta pontilhada representa uma relação genérica, como descrito na documentação do Django:
http://docs.djangoproject.com/en/dev/ref/contrib/contenttypes/#id1

6.3 Casos de Uso

Esta seção descreve a utilização do sistema através de *Casos de Uso*, representando os principais atores e suas interações com o sistema.

Os Casos de Uso servem de auxílio à compreensão dos usuários quanto à implementação e uso do sistema.

6.3.1 Casos de Uso do SIGI

A Figura 2 apresenta os Casos de Uso do SIGI de maneira simplificada.

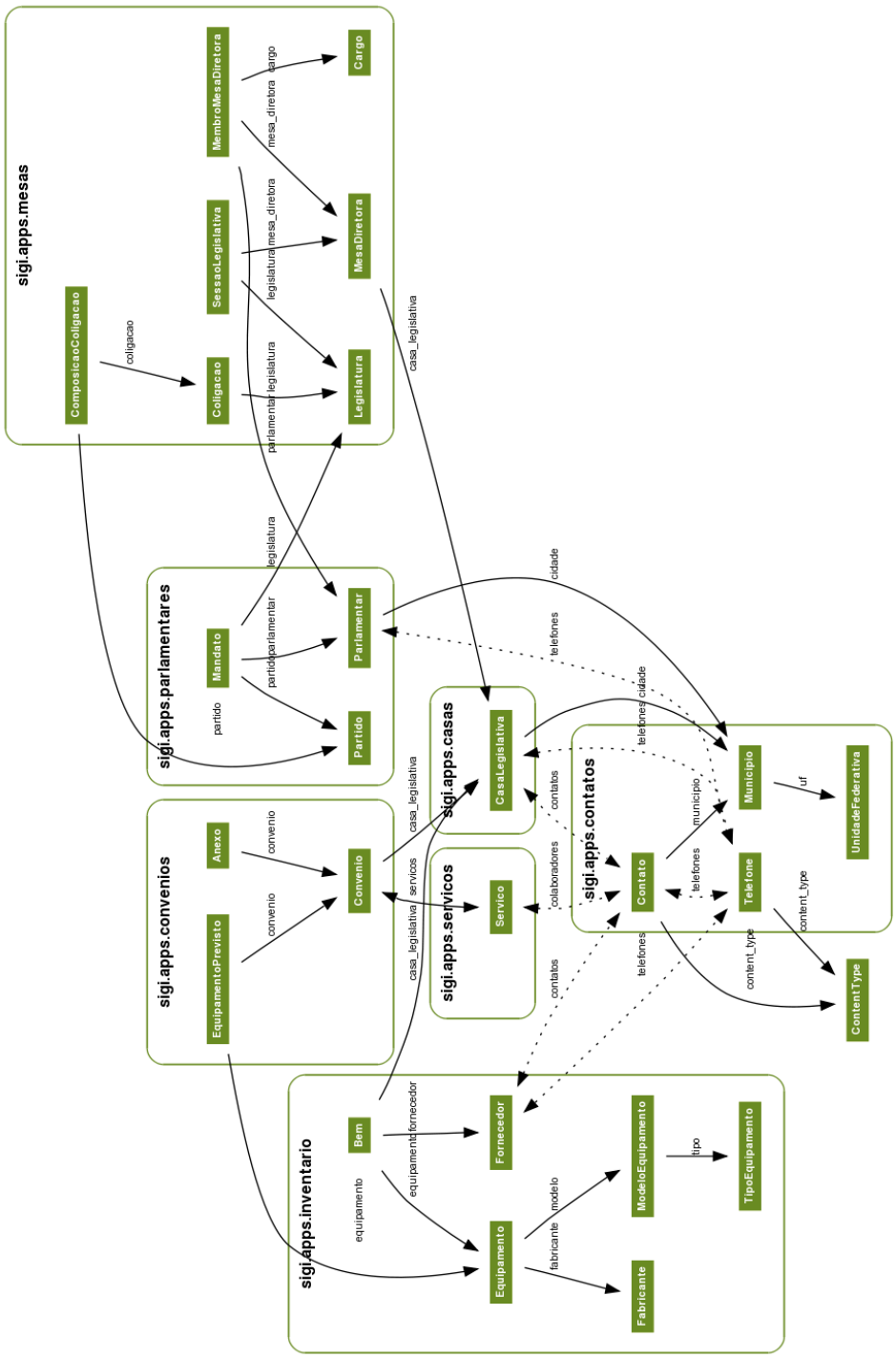

Figura 1: Relacionamento entre as aplicações

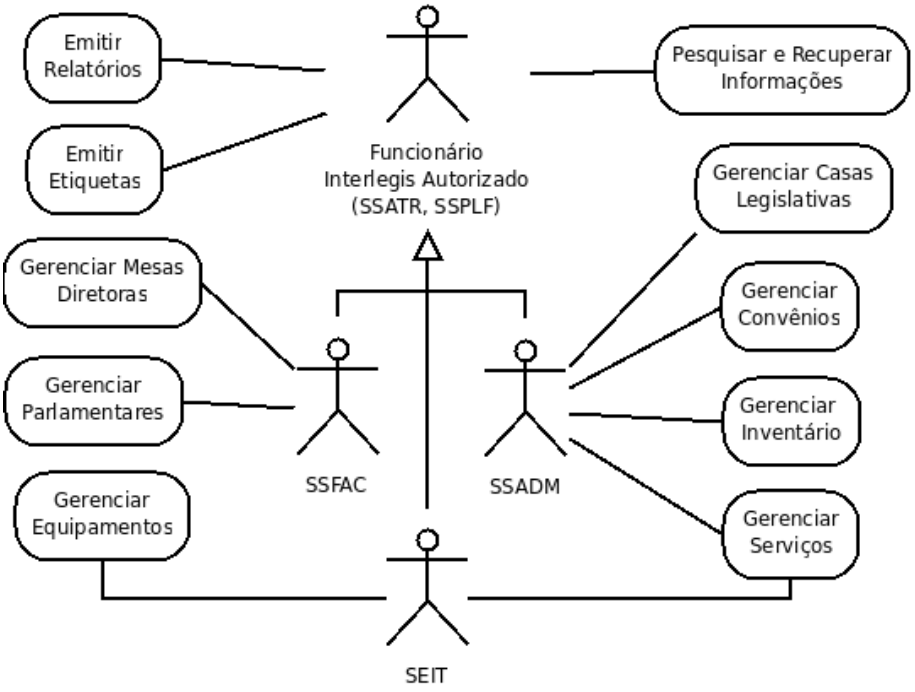

Figura 2: Casos de Uso

Descrição dos Atores

Funcionário Interlegis Autorizado: usuário apenas com atribuições de leitura no sistema. Inclui pessoal da Subsecretaria de Apoio Técnico e Relações Institucionais (SSATR) e Subsecretaria de Planejamento e Fomento (SSPLF).

SSFAC: Subsecretaria de Formação e Atendimento à Comunidade do Legislativo.

SSADM: Subsecretaria de Administração.

SEIT: Serviço de Infra-estrutura Tecnológica.

Descrição das Atividades

Emitir Relatórios: consiste em obter informações e emitir relatórios de diversas partes do sistema.

Emitir Etiquetas: consiste em obter informações e emitir etiquetas de algumas partes do sistema.

Pesquisar e Recuperar Informações: habilidades de pesquisa e obtenção de informações da base de dados do sistema.

Gerenciar Mesas Diretoras: consiste em inserir, atualizar e remover *Mesas Diretoras*, *Sessões Legislativas* e modificar a *Composição das Mesas Diretoras*.

Gerenciar Parlamentares: consiste em inserir, atualizar e remover *Parlamentares* e *Partidos*.

Gerenciar Equipamentos: consiste em inserir, atualizar e remover *Equipamentos* e *Fornecedores*.

Gerenciar Casas Legislativas: consiste em inserir, atualizar e remover *Casas Legislativas*.

Gerenciar Convênios: consiste em inserir, atualizar e remover *Convênios*.

Gerenciar Inventário: consiste em atualizar o *Inventário* das Casas Legislativas.

Gerenciar Serviços: consiste em inserir, atualizar e remover *Serviços* prestados às Casas Legislativas.

6.4 Modelo de dados

6.4.1 Aplicação sigi.apps.casas (figura 3)

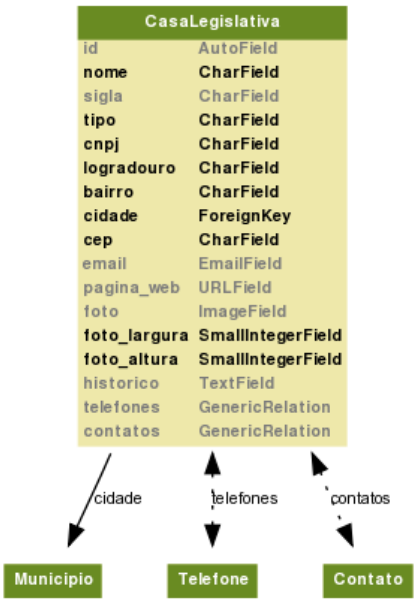

Figura 3: Diagrama de Classes da aplicação sigi.apps.casas

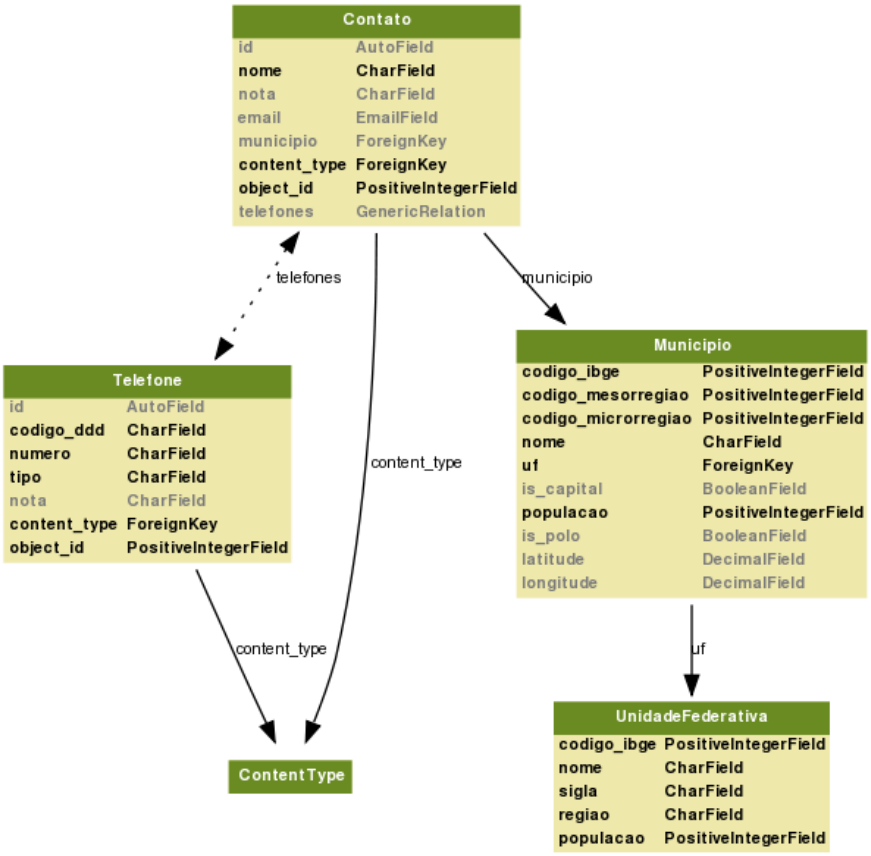

Figura 4: Diagrama de Classes da aplicação sigi.apps.contatos

6.4.2 Aplicação sigi.apps.contatos (figura 4)

6.4.3 Aplicação sigi.apps.convenios (figura 5)

6.4.4 Aplicação sigi.apps.inventario (figura 6)

6.4.5 Aplicação sigi.apps.mesas (figura 7)

6.4.6 Aplicação sigi.apps.parlamentares (figura 8)

6.4.7 Aplicação sigi.apps.servicos (figura 9)

6.5 Esquema de dados

Nesta seção estão descritos os esquemas de criação das entidades e de seus relacionamentos do banco de dados *SQL* para cada aplicação do SIGI.

6.5.1 Aplicação sigi.apps.casas

```
BEGIN;
CREATE TABLE "casas_casalegislativa" (
    "id" integer NOT NULL PRIMARY KEY,
```

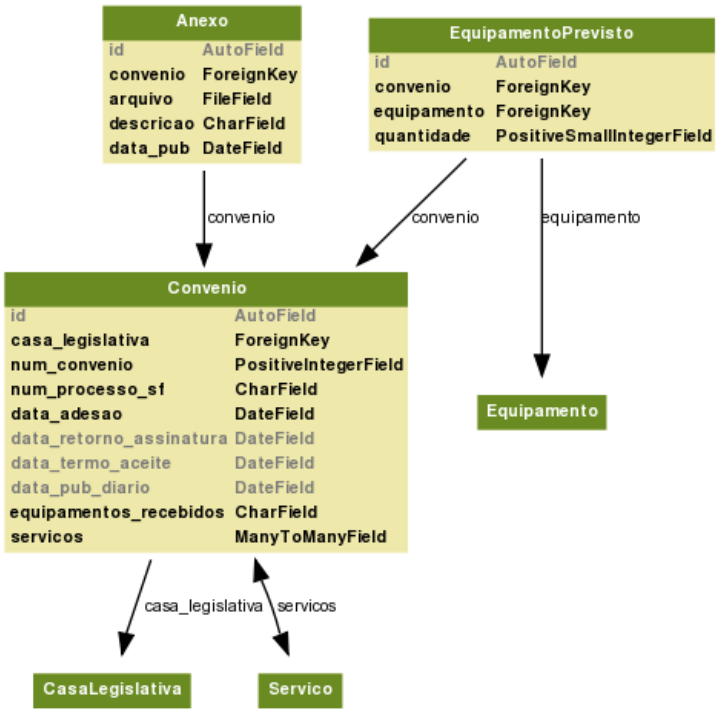

Figura 5: Diagrama de Classes da aplicação sigi.apps.convenios

```
    "nome" varchar(60) NOT NULL,
    "sigla" varchar(30) NOT NULL,
    "tipo" varchar(2) NOT NULL,
    "cnpj" varchar(18) NOT NULL,
    "logradouro" varchar(100) NOT NULL,
    "bairro" varchar(40) NOT NULL,
    "cidade_id" integer NOT NULL,
    "cep" varchar(9) NOT NULL,
    "email" varchar(75) NOT NULL,
    "pagina_web" varchar(200) NOT NULL,
    "foto" varchar(100) NOT NULL,
    "foto_largura" smallint NULL,
    "foto_altura" smallint NULL,
    "historico" text NOT NULL
)
;
COMMIT;
```

6.5.2 Aplicação sigi.apps.contatos

```
BEGIN;
CREATE TABLE "contatos_unidadefederativa" (
    "codigo_ibge" integer unsigned NOT NULL PRIMARY KEY,
```

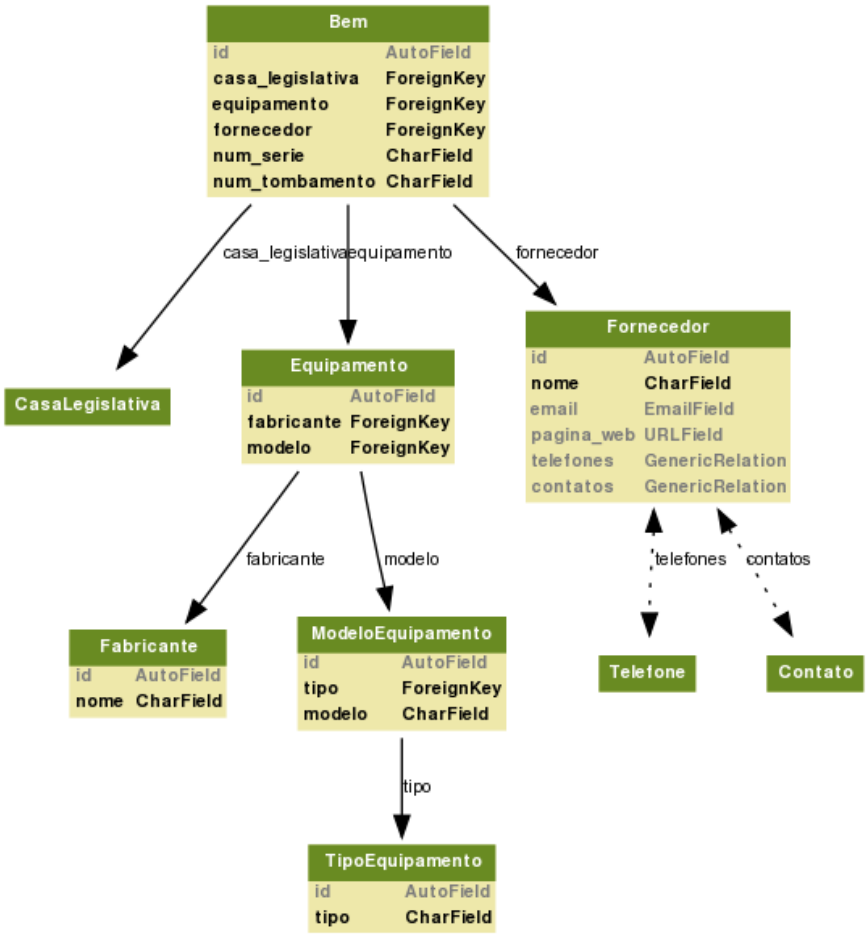

Figura 6: Diagrama de Classes da aplicação sigi.apps.inventario

```
    "nome" varchar(25) NOT NULL,
    "sigla" varchar(2) NOT NULL,
    "regiao" varchar(2) NOT NULL,
    "populacao" integer unsigned NOT NULL
)
;
CREATE TABLE "contatos_telefone" (
    "id" integer NOT NULL PRIMARY KEY,
    "codigo_ddd" varchar(2) NOT NULL,
    "numero" varchar(9) NOT NULL,
    "tipo" varchar(1) NOT NULL,
    "nota" varchar(70) NOT NULL,
    "content_type_id" integer NOT NULL,
    "object_id" integer unsigned NOT NULL,
    UNIQUE ("codigo_ddd", "numero", "tipo")
)
;
```

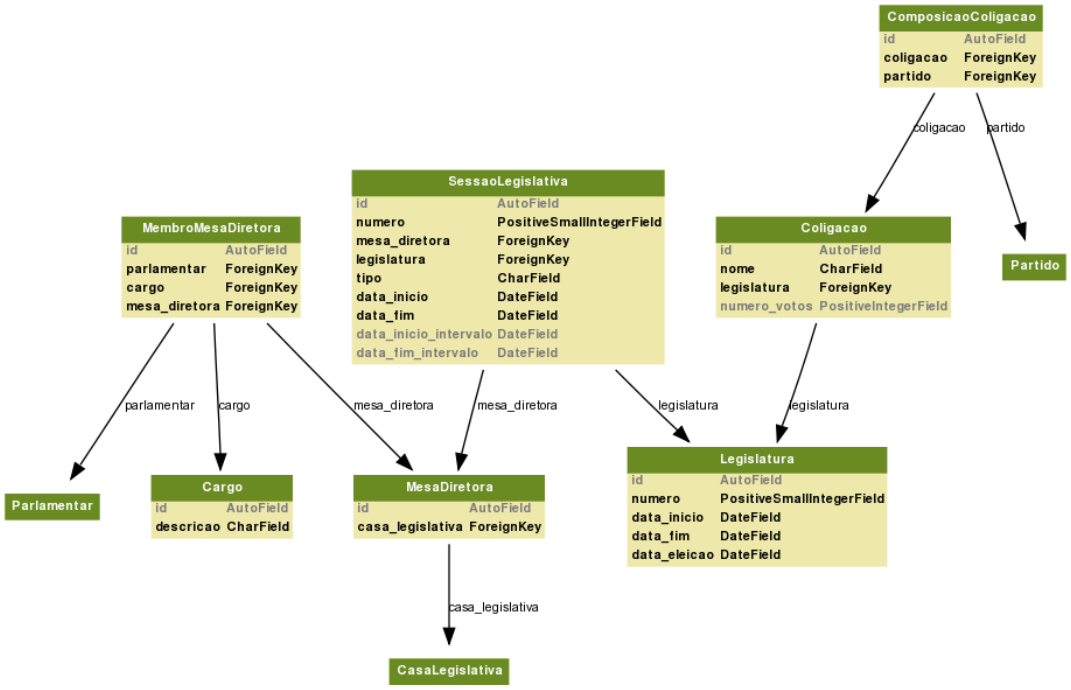

Figura 7: Diagrama de Classes da aplicação sigi.apps.mesas

```
CREATE TABLE "contatos_municipio" (
    "codigo_ibge" integer unsigned NOT NULL PRIMARY KEY,
    "codigo_mesorregiao" integer unsigned NOT NULL,
    "codigo_microrregiao" integer unsigned NOT NULL,
    "nome" varchar(50) NOT NULL,
    "uf_id" integer NOT NULL REFERENCES "contatos_unidadefederativa"
        ("codigo_ibge"),
    "is_capital" bool NOT NULL,
    "populacao" integer unsigned NOT NULL,
    "is_polo" bool NOT NULL,
    "latitude" decimal NULL,
    "longitude" decimal NULL
)
;
CREATE TABLE "contatos_contato" (
    "id" integer NOT NULL PRIMARY KEY,
    "nome" varchar(60) NOT NULL,
    "nota" varchar(70) NOT NULL,
    "email" varchar(75) NOT NULL,
    "municipio_id" integer NULL REFERENCES "contatos_municipio"
        ("codigo_ibge"),
    "content_type_id" integer NOT NULL,
    "object_id" integer unsigned NOT NULL
)
```

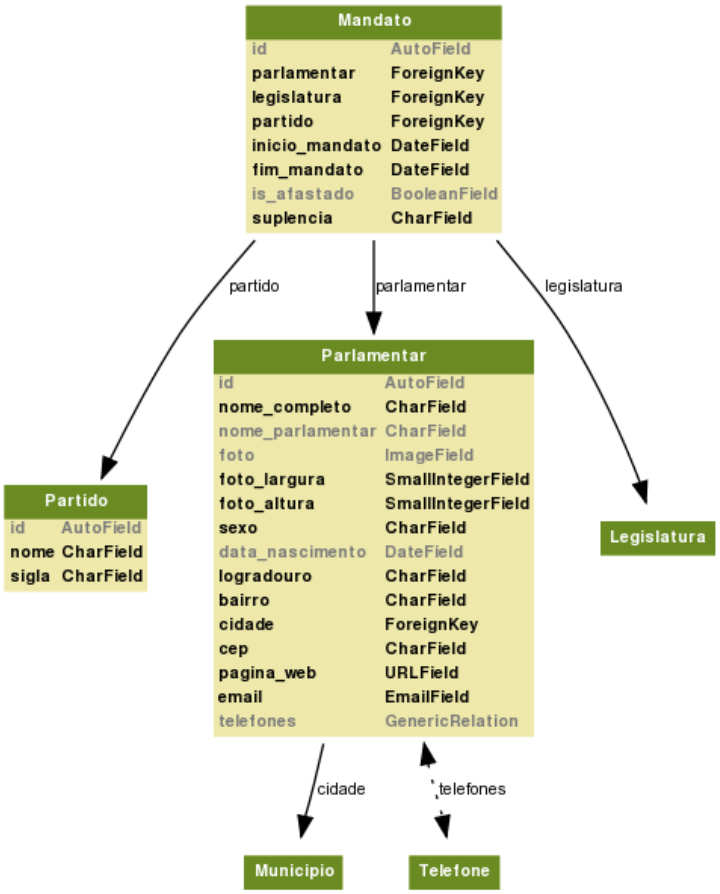

Figura 8: Diagrama de Classes da aplicação sigi.apps.parlamentares

Figura 9: Diagrama de Classes da aplicação sigi.apps.servicos

;

```
COMMIT;
```

6.5.3 Aplicação sigi.apps.convenios

```
BEGIN;
CREATE TABLE "convenios_anexo" (
    "id" integer NOT NULL PRIMARY KEY,
    "convenio_id" integer NOT NULL,
    "arquivo" varchar(100) NOT NULL,
    "descricao" varchar(70) NOT NULL,
    "data_pub" date NOT NULL
)
;
CREATE TABLE "convenios_convenio" (
    "id" integer NOT NULL PRIMARY KEY,
    "casa_legislativa_id" integer NOT NULL,
    "num_convenio" integer unsigned NOT NULL,
    "num_processo_sf" varchar(11) NOT NULL,
    "data_adesao" date NOT NULL,
    "data_retorno_assinatura" date NULL,
    "data_termo_aceite" date NULL,
    "data_pub_diario" date NULL,
    "equipamentos_recebidos" varchar(1) NOT NULL
)
;
CREATE TABLE "convenios_equipamentoprevisto" (
    "id" integer NOT NULL PRIMARY KEY,
    "convenio_id" integer NOT NULL REFERENCES "convenios_convenio" ("id"),
    "equipamento_id" integer NOT NULL,
    "quantidade" smallint unsigned NOT NULL
)
;
CREATE TABLE "convenios_convenio_servicos" (
    "id" integer NOT NULL PRIMARY KEY,
    "convenio_id" integer NOT NULL REFERENCES "convenios_convenio" ("id"),
    "servico_id" integer NOT NULL REFERENCES "servicos_servico" ("id"),
    UNIQUE ("convenio_id", "servico_id")
)
;
COMMIT;
```

6.5.4 Aplicação sigi.apps.inventario

```
BEGIN;
CREATE TABLE "inventario_bem" (
    "id" integer NOT NULL PRIMARY KEY,
    "casa_legislativa_id" integer NOT NULL,
```

```
    "equipamento_id" integer NOT NULL,
    "fornecedor_id" integer NOT NULL,
    "num_serie" varchar(50) NOT NULL UNIQUE,
    "num_tombamento" varchar(50) NOT NULL UNIQUE
)
;
CREATE TABLE "inventario_fabricante" (
    "id" integer NOT NULL PRIMARY KEY,
    "nome" varchar(40) NOT NULL
)
;
CREATE TABLE "inventario_fornecedor" (
    "id" integer NOT NULL PRIMARY KEY,
    "nome" varchar(40) NOT NULL,
    "email" varchar(75) NOT NULL,
    "pagina_web" varchar(200) NOT NULL
)
;
CREATE TABLE "inventario_tipoequipamento" (
    "id" integer NOT NULL PRIMARY KEY,
    "tipo" varchar(40) NOT NULL
)
;
CREATE TABLE "inventario_equipamento" (
    "id" integer NOT NULL PRIMARY KEY,
    "fabricante_id" integer NOT NULL REFERENCES "inventario_fabricante" ("id"),
    "modelo_id" integer NOT NULL,
    UNIQUE ("fabricante_id", "modelo_id")
)
;
CREATE TABLE "inventario_modeloequipamento" (
    "id" integer NOT NULL PRIMARY KEY,
    "tipo_id" integer NOT NULL REFERENCES "inventario_tipoequipamento" ("id"),
    "modelo" varchar(30) NOT NULL
)
;
COMMIT;
```

6.5.5 Aplicação sigi.apps.mesas

```
BEGIN;
CREATE TABLE "mesas_membromesadiretora" (
    "id" integer NOT NULL PRIMARY KEY,
    "parlamentar_id" integer NOT NULL,
    "cargo_id" integer NOT NULL,
    "mesa_diretora_id" integer NOT NULL
)
```

```
;
CREATE TABLE "mesas_cargo" (
    "id" integer NOT NULL PRIMARY KEY,
    "descricao" varchar(30) NOT NULL
)
;
CREATE TABLE "mesas_coligacao" (
    "id" integer NOT NULL PRIMARY KEY,
    "nome" varchar(50) NOT NULL,
    "legislatura_id" integer NOT NULL,
    "numero_votos" integer unsigned NULL
)
;
CREATE TABLE "mesas_legislatura" (
    "id" integer NOT NULL PRIMARY KEY,
    "numero" smallint unsigned NOT NULL,
    "data_inicio" date NOT NULL,
    "data_fim" date NOT NULL,
    "data_eleicao" date NOT NULL
)
;
CREATE TABLE "mesas_sessaolegislativa" (
    "id" integer NOT NULL PRIMARY KEY,
    "numero" smallint unsigned NOT NULL UNIQUE,
    "mesa_diretora_id" integer NOT NULL,
    "legislatura_id" integer NOT NULL REFERENCES "mesas_legislatura" ("id"),
    "tipo" varchar(1) NOT NULL,
    "data_inicio" date NOT NULL,
    "data_fim" date NOT NULL,
    "data_inicio_intervalo" date NULL,
    "data_fim_intervalo" date NULL
)
;
CREATE TABLE "mesas_mesadiretora" (
    "id" integer NOT NULL PRIMARY KEY,
    "casa_legislativa_id" integer NOT NULL
)
;
CREATE TABLE "mesas_composicaocoligacao" (
    "id" integer NOT NULL PRIMARY KEY,
    "coligacao_id" integer NOT NULL REFERENCES "mesas_coligacao" ("id"),
    "partido_id" integer NOT NULL
)
;
COMMIT;
```

6.5.6 Aplicação sigi.apps.parlamentares

```
BEGIN;
CREATE TABLE "parlamentares_partido" (
    "id" integer NOT NULL PRIMARY KEY,
    "nome" varchar(50) NOT NULL,
    "sigla" varchar(10) NOT NULL
)
;
CREATE TABLE "parlamentares_parlamentar" (
    "id" integer NOT NULL PRIMARY KEY,
    "nome_completo" varchar(60) NOT NULL,
    "nome_parlamentar" varchar(35) NOT NULL,
    "foto" varchar(100) NOT NULL,
    "foto_largura" smallint NULL,
    "foto_altura" smallint NULL,
    "sexo" varchar(1) NOT NULL,
    "data_nascimento" date NULL,
    "logradouro" varchar(100) NOT NULL,
    "bairro" varchar(40) NOT NULL,
    "cidade_id" integer NOT NULL,
    "cep" varchar(9) NOT NULL,
    "pagina_web" varchar(200) NOT NULL,
    "email" varchar(75) NOT NULL
)
;
CREATE TABLE "parlamentares_mandato" (
    "id" integer NOT NULL PRIMARY KEY,
    "parlamentar_id" integer NOT NULL REFERENCES
        "parlamentares_parlamentar" ("id"),
    "legislatura_id" integer NOT NULL,
    "partido_id" integer NOT NULL REFERENCES "parlamentares_partido" ("id"),
    "inicio_mandato" date NOT NULL,
    "fim_mandato" date NOT NULL,
    "is_afastado" bool NOT NULL,
    "suplencia" varchar(1) NOT NULL
)
;
COMMIT;
```

6.5.7 Aplicação sigi.apps.servicos

```
BEGIN;
CREATE TABLE "servicos_servico" (
    "id" integer NOT NULL PRIMARY KEY,
    "tipo" varchar(50) NOT NULL,
    "descricao" text NOT NULL,
    "data_inicio" date NULL,
```

```
    "data_fim" date NULL,
    "situacao" varchar(1) NOT NULL,
    "avaliacao" smallint unsigned NULL
)
;
COMMIT;
```

6.6 Manual de Instalação

Este documento descreve como efetuar a instalação do Sistema de Informações Gerenciais do Interlegis (SIGI). Os passos são baseados nas distribuições GNU/Linux Debian e Ubuntu. Podem ser aplicados sem dificuldades em outras distribuições, mas poderão sofrer alguma adaptação.

6.6.1 Preparação do ambiente para a instalação

Há dois tipos de instalação do SIGI, uma para desenvolvimento e outra para produção.

Para ambas instalações, certifique-se que os seguintes softwares estejam instalados em seu sistema:

- Python >= 2.4 < 3.0
- Django 1.0

O Python pode ser instalado a partir do pacote `python` com uma ferramenta de instalação de pacotes como o `apt-get` ou `aptitude`.

Se a sua distribuição não possui o pacote `python-django`, para o Django na versão 1.0, será necessário obter e configurar o mesmo manualmente. A próxima seção traz mais detalhes para esta tarefa.

Instalação do Django

O Django 1.0 pode ser obtido através da página de download do Django Project via *tarball* (`tar.gz`) ou via Subversion (revisão 8961).

Obtendo e instalando o Django

Primeiramente baixe o tarball do Django 1.0 ou dê *checkout* na tag 1.0 disponível no repositório do Subversion do projeto:

```
svn checkout http://code.djangoproject.com/svn/django/tags/releases/1.0/
```

Após isso será necessário colocar o pacote Python do `django` (diretório `django` disponível dentro do diretório/tarball baixado) no *path* do Python.

Para saber quais são os diretórios que estão no path do Python, execute em linha de comando:

```
python -c 'import sys; print sys.path'
```

Você tem opção de mover o pacote `django` para algum diretório coberto pelo path ou adicionar um outro local no path utilizando a variável de ambiente `PYTHONPATH` do usuário que rodará o sistema. O formato do `PYTHONPATH` é o mesmo do `PATH` do sistema. Exemplo:

```
PYTHONPATH=/path/to/django:$PYTHONPATH
```

De maneira simplificada, a instalação do Django é como a instalação de qualquer outro pacote Python.

Preparando o ambiente para desenvolvimento

Se você irá desenvolver o SIGI, é necessário possuir os seguintes pacotes instalados em sua máquina:

- sqlite3, para SQLite 3;
- python-pysqlite2, interface Python para SQLite 3.

Preparando o ambiente para produção

Para disponibilizar o SIGI em um ambiente para produção você necessitará do MySQL Server ou PostgreSQL, ou qualquer outro Sistema Gerenciador de Banco de Dados (SGBD) compatível com Django 1.0, e de suas seguintes interfaces para Python, como os pacotes `python-mysqldb` e `python-psycopg2` (respectivamente para MySQL e PostgreSQL).

Será necessário também possuir o Apache (>= 2.2) instalado para disponibilizar o sistema via HTTP, de forma cliente/servidor, ou outro servidor web compatível com Python e WSGI (ou FastCGI).

Opcionalmente, você pode utilizar o servidor web Lighttpd para servir os arquivos estáticos do sistema.

Você encontrará configurações para Apache com WSGI neste documento, juntamente com instruções para configurar o SIGI com o banco de dados escolhido.

6.6.2 Instalação e configuração do SIGI

O SIGI está disponível no Colab, um portal colaborativo para a gerência dos projetos de software do Interlegis.

A página do projeto SIGI no Colab pode ser acessada através do link http://colab.interlegis.gov.br/wiki/ProjetoSigi.

Obtendo e instalando o SIGI

Para baixar o SIGI, é necessário ter o Subversion instalado em sua máquina (pacote `subversion`). Para *checkout* do sistema via Subversion, execute o comando abaixo:

```
svn checkout http://repositorio.interlegis.gov.br/SIGI/trunk/ \
  /path/to/SIGI
```

Substitua `/path/to/SIGI` para o local onde deseja instalar o SIGI. (Iremos considerar o diretório SIGI como a raiz do projeto durante o restante deste documento.)

Configuração do SIGI

Dentro da raiz do projeto, encontra-se um diretório de nome `sigi` (pacote Python do projeto). Sua configuração padrão, se encontra no arquivo `settings.py`.

Para alterar as configurações do projeto, é recomendado que copie o arquivo `local_settings.template` para `local_settings.py` e altere os parâmetros neste arquivo.

Instalação em ambiente para desenvolvimento

As configurações padrão do projeto já estão direcionadas para um ambiente de desenvolvimento e não exige demais configurações.

(Os comandos abaixo deverão ser executados dentro do diretório `sigi`, onde se encontra o arquivo `manage.py`.)

Base de dados em ambiente para desenvolvimento

Antes de executar o projeto, é necessário preencher o banco de dados com suas tabelas e valores padrão.

Considerando que possua o SQLite instalado, basta executar o comando abaixo:

```
python manage.py syncdb
```

Execução do projeto em ambiente para desenvolvimento

Para rodar o SIGI, execute em linha de comando:

```
python manage.py runserver
```

O projeto poderá ser acessado através do endereço http://127.0.0.1:8000/.

Instalação em ambiente para produção

Para instalar o SIGI em ambiente para produção, devemos configurar um Sistema Gerenciador de Banco de Dados (SGBD), o qual armazenará os dados do sistema, e configurar um servidor HTTP, como o Apache, para disponibilizar o sistema na web.

Configuração do SGDB

Modifique em seu `local_settings.py` as variáveis `DATABASE_ENGINE`, `DATABASE_NAME`, `DATABASE_USER`, `DATABASE_PASSWORD`, `DATABASE_HOST` e `DATABASE_PORT` de acordo com o banco de dados disponibilizado para o SIGI.

Em `DATABASE_ENGINE`, suas opções são `postgresql_psycopg2`, `postgresql`, `mysql`, `sqlite3` ou `ado_mssql`.

Após isso, é necessário preencher o banco de dados com suas tabelas e valores padrão. Para tal, execute em linha de comando:

```
python manage.py syncdb
```

Configuração do servidor web Apache com WSGI

Exemplos de configurações, de um `VirtualHost`, estão disponíveis no diretório `etc/apache/` dentro do diretório raiz do projeto (`SIGI/`).