

## Tabelas do Banco de Dados da Plataforma Ybatinga

Versão 2.0 17/06/2016

Eduardo Alcântara

### Colunas Específicas

Para não poluir de informações os scripts de tabelas do banco, aqui serão descritas as colunas com nomes menos intuitivos.

1. usuario.super\_sn - se o usuário é ou não é um Super Usuário (que possui nenhuma restrição de acesso e pode se logar como outra pessoa);
2. usuario.sistema\_sn - se o usuário é ou não é um usuário padrão de sistema (robô) usado para acessos à páginas públicas, sem usuário logado;
3. procedure.tipos - lista com o tipo de cada parâmetro de entrada da procedure informada no registro: S(str), N(int), B(bool);
4. servicos\_pacote.valor\_unitario - preço mínimo pela uso de um pacote de serviços (serviço) que pode ser quantidade (1) ou tempo (mês)
5. \*.ativo\_sn - se o registro está ativo ele pode ser visível e calculável, se estiver inativo, comporta-se como se tivesse sido apagado do banco;
6. \*.disponivel\_sn - se o registro está disponível para interação com o usuário (exemplo: um pacote de assinatura que não está mais à venda);
7. pacote\_perfis.valor\_relativo - indica uma porcentagem em relação ao valor do pacote, que pode conter um desconto ou aumento, dependendo do ciclo de assinatura e da quantidade de usuários permitidas nesse perfil de pacote;
8. pacote\_perfis.valor\_fechado - indica que aquela oferta de perfil de pacote não tem valor relativo a algum percentual do valor mínimo do pacote, mas sim um valor inventado pela criador do perfil, que pode ser maior, menor ou igual ou valor base do pacote;

```
---- para compatibilidade com maioria dos sgbd não usaremos enumeradores como tipos de coluna, e sim, tabelas.  
-- create type bloqueio_tipo as enum ('confirmação','judicial','financeiro','próprio','senha');  
-- create type bloqueio_fonte as enum ('plataforma','serviço','sistema','administração','judicial','usuário');  
-- create type perfil_tipo as enum ('padrão','suporte','administrador','desenvolvedor','diretor','super','sistema');  
-- create type servico_tipo as enum ('painel','mensageria','modulo','procedure','repositorio','acesso','streaming');  
-- create type procedure_tipo as enum ('criar','listar','atualizar','apagar');  
-- create type procedure_saida as enum ('tabela','registro','célula','variado');  
-- create type credito_status as enum ('aberto','compensado','recusado','cancelado','reembolsado');
```

```
set schema 'ybatinga';
```

---- *lista para a tabela usuarios\_bloqueio.*

```
drop table if exists ybatinga.lista_bloqueio_tipo;
create table if not exists ybatinga.lista_bloqueio_tipo (
    id                bigserial,
    chave             varchar(32) not null,
    rotulo_singular   varchar(31) not null,
    rotulo_plural     varchar(32),
    descricao         varchar(255),
    constraint lista_bloqueio_tipo_id primary key(id)
);
create unique index lista_bloqueio_tipo_key_ix on ybatinga.lista_bloqueio_tipo (chave);
create unique index lista_bloqueio_tipo_ls_ix on ybatinga.lista_bloqueio_tipo (rotulo_singular);
create unique index lista_bloqueio_tipo_lp_ix on ybatinga.lista_bloqueio_tipo (rotulo_plural);

insert into ybatinga.lista_bloqueio_tipo (chave, rotulo_singular, rotulo_plural, descricao) values
('confirmacao','Confirmação','Confirmações',''),
('judicial','Judicial','Judiciais',''),
('financeiro','Financeiro','Financeiros',''),
('administrativo','Administrativo','Administrativos',''),
('senha','Senha','Senhas','');
```

---- *lista para a tabela usuarios\_bloqueio.*

```
drop table if exists ybatinga.lista_bloqueio_fonte;
create table if not exists ybatinga.lista_bloqueio_fonte (
    id                bigserial,
    chave              varchar(32) not null,
    rotulo_singular    varchar(31) not null,
    rotulo_plural      varchar(32),
    descricao          varchar(255),
    constraint lista_bloqueio_fonte_id primary key(id)
);
create unique index lista_bloqueio_fonte_key_ix on ybatinga.lista_bloqueio_fonte (chave);
create unique index lista_bloqueio_fonte_ls_ix on ybatinga.lista_bloqueio_fonte (rotulo_singular);
create unique index lista_bloqueio_fonte_lp_ix on ybatinga.lista_bloqueio_fonte (rotulo_plural);

insert into ybatinga.lista_bloqueio_fonte (chave, rotulo_singular, rotulo_plural, descricao) values
('plataforma', 'Plataforma', 'Plataformas', ''),
('servico', 'Serviço', 'Serviços', ''),
('sistema', 'Sistema', 'Sistemas', ''),
('administracao', 'Administração', 'Administrativos', ''),
('judicial', 'Judicial', 'Judiciais', ''),
('usuario', 'Usuário', 'Usuários', '');
```

---- lista para a tabela perfil\_usuario.

```
drop table if exists ybatinga.lista_perfil_tipo;
```

```
create table if not exists ybatinga.lista_perfil_tipo (
```

```
    id                bigserial,
```

```
    chave             varchar(32) not null,
```

```
    rotulo_singular   varchar(31) not null,
```

```
    rotulo_plural     varchar(32),
```

```
    descricao         varchar(255),
```

```
    constraint lista_perfil_tipo_id primary key(id)
```

```
); create unique index lista_perfil_tipo_key_ix on ybatinga.lista_perfil_tipo (chave);
```

```
create unique index lista_perfil_tipo_ls_ix on ybatinga.lista_perfil_tipo (rotulo_singular);
```

```
create unique index lista_perfil_tipo_lp_ix on ybatinga.lista_perfil_tipo (rotulo_plural);
```

```
insert into ybatinga.lista_perfil_tipo (chave, rotulo_singular, rotulo_plural, descricao) values
```

```
('padrão','Padrão','Padrões',''),
```

```
('suporte','Suporte','Suportes',''),
```

```
('administrador','Administrador','Administradores',''),
```

```
('desenvolvedor','Desenvolvedor','Desenvolvedores',''),
```

```
('diretor','Diretor','Diretores',''),
```

```
('super','Super','Supers',''),
```

```
('sistema','Sistema','Sistemas','');
```

---- lista para a tabela servico.

```
drop table if exists ybatinga.lista_servico_tipo;
create table if not exists ybatinga.lista_servico_tipo (
    id                bigserial,
    chave             varchar(32) not null,
    rotulo_singular   varchar(31) not null,
    rotulo_plural     varchar(32),
    descricao         varchar(255),
    constraint lista_servico_tipo_id primary key(id)
);
create unique index lista_servico_tipo_key_ix on ybatinga.lista_servico_tipo (chave);
create unique index lista_servico_tipo_ls_ix on ybatinga.lista_servico_tipo (rotulo_singular);
create unique index lista_servico_tipo_lp_ix on ybatinga.lista_servico_tipo (rotulo_plural);
```

```
insert into ybatinga.lista_servico_tipo (chave, rotulo_singular, rotulo_plural, descricao) values
('painel','Painel Administrativo','Paineis Administrativos',''),
('mensageria','Mensageria','Mensagerias',''),
('executavel','Módulo Executável','Módulos Executáveis',''),
('web','Aplicação Web','Aplicações Web',''),
('banco','Banco de Dados','Bancos de Dados',''),
('repositorio','Repositório de Arquivos','Repositórios de Arquivos',''),
('acesso','Acesso, Autenticação e Sessão','Acesso, Autenticação e Sessões','');
```

---- *lista para a tabela procedure.*

```
drop table if exists ybatinga.lista_procedure_tipo;
create table if not exists ybatinga.lista_procedure_tipo (
    id                bigserial,
    chave             varchar(32) not null,
    rotulo_singular   varchar(31) not null,
    rotulo_plural     varchar(32),
    descricao         varchar(255),
    constraint lista_procedure_tipo_id primary key(id)
);
create unique index lista_procedure_tipo_key_ix on ybatinga.lista_procedure_tipo (chave);
create unique index lista_procedure_tipo_ls_ix on ybatinga.lista_procedure_tipo (rotulo_singular);
create unique index lista_procedure_tipo_lp_ix on ybatinga.lista_procedure_tipo (rotulo_plural);

insert into ybatinga.lista_procedure_tipo (chave, rotulo_singular, rotulo_plural, descricao) values
('criar','Criar','Criar',''),
('listar','Listar','Listar',''),
('atualizar','Atualizar','Atualizar',''),
('apagar','Apagar','Apagar',''),
('verificar','Verificar','Verificar','');
```

---- *lista para a tabela procedure.*

```
drop table if exists ybatinga.lista_procedure_saida;
create table if not exists ybatinga.lista_procedure_saida (
    id                bigserial,
    chave             varchar(32) not null,
    rotulo_singular   varchar(31) not null,
    rotulo_plural     varchar(32),
    descricao         varchar(255),
    constraint lista_procedure_saida_id primary key(id)
);
create unique index lista_procedure_saida_key_ix on ybatinga.lista_procedure_saida (chave);
create unique index lista_procedure_saida_ls_ix on ybatinga.lista_procedure_saida (rotulo_singular);
create unique index lista_procedure_saida_lp_ix on ybatinga.lista_procedure_saida (rotulo_plural);
```

```
insert into ybatinga.lista_procedure_saida (chave, rotulo_singular, rotulo_plural, descricao) values
('tabela','Tabela','Tabelas',''),
('registro','Registro','Registros',''),
('celula','Célula','Células',''),
('variada','Variada','Variadas',''),
('texto','Texto','Textos',''),
('html','HTML','HTML',''),
('numero_inteiro','Número Inteiro','Números Inteiros',''),
('numero_fracionado','Número Fracionado','Números Fracionados',''),
('data','Data','Datas',''),
('hora','Hora','Horas',''),
('momento','Data e Hora','Datas e Horas',''),
('binario','Binário','Binários','');
```

---- lista para a tabela usuario\_credito.

```
drop table if exists ybatinga.lista_credito_status;
create table if not exists ybatinga.lista_credito_status (
    id                bigserial,
    chave             varchar(32) not null,
    rotulo_singular   varchar(31) not null,
    rotulo_plural     varchar(32),
    descricao         varchar(255),
    constraint lista_credito_status_id primary key(id)
);
create unique index lista_credito_status_key_ix on ybatinga.lista_credito_status (chave);
create unique index lista_credito_status_ls_ix on ybatinga.lista_credito_status (rotulo_singular);
create unique index lista_credito_status_lp_ix on ybatinga.lista_credito_status (rotulo_plural);

insert into ybatinga.lista_credito_status (chave, rotulo_singular, rotulo_plural, descricao) values
('em_aberto','Em Aberto','Em Aberto',''),
('compensado','Compensado','Compensados',''),
('recusado','Recusado','Recusados',''),
('cancelado','Cancelado','Cancelados',''),
('reembolsado','Reembolsado','Reembolsados','');
```

-----

-----

-----

-----



---- cadastro de um usuário que acessa a plataforma para administrá-la ou consumir algum de seus serviços.

```
drop table if exists ybatinga.usuario;
create table if not exists ybatinga.usuario (
    id                bigserial,
    pessoa_id        int8 constraint usuario_pessoa_fk references pessoas.pessoa(id),
    nome              varchar(99) not null,
    senha             varchar(32) not null,
    super_sn          bool default false,
    sistema_sn        bool default false,
    ativo_sn          bool default true,
    constraint usuario_id primary key(id)
); create unique index usuario_nome_ix on ybatinga.usuario (nome);
```

---- cadastro de bloqueios e desbloqueios que ocorreram a usuários.

```
drop table if exists ybatinga.usuario_bloqueio;
create table if not exists ybatinga.usuario_bloqueio (
    id                bigserial,
    usuario_id        int8 constraint usuario_bloqueio_fk references ybatinga.usuario(id),
    momento_dt        timestampz default current_timestamp,
    bloqueado_sn       bool not null,
    tipo              varchar(32) constraint usuario_bloqueio_tipo_ls references ybatinga.lista_bloqueio_tipo(chave),
    fonte              varchar(32) constraint usuario_bloqueio_fonte_ls references ybatinga.lista_bloqueio_fonte(chave),
    motivo             varchar(99),
    constraint usuario_bloqueio_id primary key(id)
); create index usuario_bloqueio_ix on ybatinga.usuario_bloqueio (usuario_id);
```

---- as sessoes ativas de uso da plataforma que pertencem ao usuario, em cada dispositivo de acesso.

```
drop table if exists ybatinga.usuario_sessao;
create table if not exists ybatinga.usuario_sessao (
    id                bigserial,
    usuario_id        int8 constraint usuario_sessao_fk references ybatinga.usuario(id),
    token             varchar(32) not null,
    chave_publica     varchar(32) null,
    chave_privada     varchar(32) null,
    conexao           varchar(32),
    aplicativo        varchar(32),
    plataforma        varchar(32),
    estatisticas      text,
    inicio_dt         timestampz default current_timestamp,
    acesso_dt         timestampz,
    acessos           int,
    ativo_sn          bool default true,
    constraint usuario_sessao_id primary key(id)
);
create index usuario_sessao_ix on ybatinga.usuario_sessao (usuario_id);
```

---- *perfis de acesso de usuários que definem o papel de cada um na plataforma, base para hierarquia.*

```
drop table if exists ybatinga.perfil_usuario;
create table if not exists ybatinga.perfil_usuario (
    id                bigserial,
    nome              varchar(99) not null,
    codigo            varchar(32) not null,
    tipo              varchar(32) constraint perfil_usuario_tipo_ls references ybatinga.lista_perfil_tipo(chave),
    hierarquia        int,
    ativo_sn          bool default true,
    constraint perfil_usuario_id primary key(id)
);
create unique index perfil_usuario_nome_ix on ybatinga.perfil_usuario (nome);
create unique index perfil_usuario_codigo_ix on ybatinga.perfil_usuario (codigo);
```

---- *relacionamento m:n da tabela usuario com a tabela perfil\_usuario permitindo vários perfis por cada usuário.*

```
drop table if exists ybatinga.usuario_perfis;
create table if not exists ybatinga.usuario_perfis (
    id                bigserial,
    usuario            int8 not null constraint usuario_perfis_mfk references ybatinga.usuario(id),
    perfil_usuario     int8 not null constraint usuario_perfis_nfk references ybatinga.perfil_usuario(id),
    ativo_sn          bool default true,
    constraint usuario_perfis_id primary key(usuario, perfil_usuario)
);
```

---- *serviços disponíveis na plataforma, sua localização e configurações.*

```
drop table if exists ybatinga.servico;
create table if not exists ybatinga.servico (
    id                bigserial,
    nome              varchar(99) not null,
    codigo            varchar(32),
    versao            varchar(32),
    descricao         varchar(255),
    tipo              varchar(32) not null constraint servico_tipo_ls references ybatinga.lista_servico_tipo(chave),
    restful_sn        bool default true,
    localizacao       text,
    configuracao      text,
    publico_sn        bool default false,
    habilitado_sn     bool default false,
    ativo_sn          bool default true,
    constraint servico_id primary key(id)
);
create unique index servico_nome_ix on ybatinga.servico (nome);
create unique index servico_codigo_ix on ybatinga.servico (codigo);
```

*---- procedures acessíveis através da plataforma via REST, contidas nos bancos de dados na lista de serviços.*

```
drop table if exists ybatinga.procedure;
```

```
create table if not exists ybatinga.procedure (
```

```
    id                bigserial,  
    servico_id        int8 not null constraint procedure_servico_fk references ybatinga.servico(id),  
    nome              varchar(64) not null,  
    metodo            varchar(32) constraint procedure_tipo_ls references ybatinga.lista_procedure_tipo(chave),  
    saida              varchar(32) not null constraint procedure_saida_ls references ybatinga.lista_procedure_saida(chave),  
    parametros         int,  
    conteudo           text,  
    publico_sn         bool default true,  
    habilitado_sn      bool default false,  
    ativo_sn           bool default true,  
    constraint procedure_id primary key(id)
```

```
); create unique index procedure_nome_ix on ybatinga.procedure (servico_id, nome);
```

*---- procedures acessíveis através da plataforma via REST, contidas nos bancos de dados na lista de serviços.*

```
drop table if exists ybatinga.parametros_procedure;
```

```
create table if not exists ybatinga.parametros_procedure (
```

```
    id                bigserial,  
    procedure_id       int8 not null constraint procedure_parametro_fk references ybatinga.procedure(id),  
    ordem              int not null,  
    nome               varchar(64) not null,  
    tipo               varchar(1) not null default "S",  
    aceita_null_sn     bool default false,  
    obrigatorio_sn     bool default false,  
    ativo_sn           bool default true,  
    constraint procedure_parametro_id primary key(id)
```

```
); create unique index procedure_parametro_nome_ix on ybatinga.parametros_procedure (procedure_id, nome);
```

```
create unique index procedure_parametro_ordem_ix on ybatinga.parametros_procedure (procedure_id, ordem);
```

---- relacionamento m:n entre a tabela perfil\_usuario e a tabela procedure

---- para permitir acesso a várias procedures através dos perfis de usuário, independente de serviço.

```
drop table if exists ybatinga.perfil_usuario_procedures;
```

```
create table if not exists ybatinga.perfil_usuario_procedures (  
    id                bigserial,  
    perfil_usuario    int8 not null constraint perfil_usuario_procedure_mfk references ybatinga.perfil_usuario(id),  
    procedure          int8 not null constraint perfil_usuario_procedure_nfk references ybatinga.procedure(id),  
    ativo_sn          bool default true,  
    constraint perfil_usuario_procedures_id primary key(perfil_usuario, procedure)  
);
```

---- relacionamento m:n entre a tabela usuario\_perfil e a tabela servico

---- para permitir acesso a vários serviços dependendo do perfil do usuário.

```
drop table if exists ybatinga.perfil_usuario_servicos;
```

```
create table if not exists ybatinga.perfil_usuario_servicos (  
    id                bigserial,  
    perfil_usuario    int8 not null constraint perfil_usuario_servicos_mfk references ybatinga.perfil_usuario(id),  
    servico           int8 not null constraint perfil_usuario_servicos_nfk references ybatinga.servico(id),  
    ativo_sn          bool default true,  
    constraint perfil_usuario_servicos_id primary key(perfil_usuario, servico)  
);
```

---- relacionamento m:n entre a tabela procedure e a tabela servico

---- para permitir acesso a várias procedures através dos serviços da plataforma.

```
drop table if exists ybatinga.servico_procedures;
```

```
create table if not exists ybatinga.servico_procedures (
```

```
    id                bigserial,  
    servico           int8 not null constraint servico_procedures_mfk references ybatinga.servico(id),  
    procedure         int8 not null constraint servico_procedures_nfk references ybatinga.procedure(id),  
    ativo_sn          bool default true,  
    constraint servico_procedures_id primary key(servico, procedure)
```

```
);
```

---- os perfis de licenças disponíveis para os usuários assinarem/adquirirem, em ciclos temporais.

```
drop table if exists ybatinga.perfil_licenca;
```

```
create table if not exists ybatinga.perfil_licenca (
```

```
    id                bigserial,  
    perfil_id         int8 constraint perfil_licenca_hfk references ybatinga.perfil_usuario(id),  
    perfil_pai        int8 constraint perfil_licenca_pai references ybatinga.perfil_licenca(id),  
    ciclo_dias        int,  
    nome              varchar(99) not null,  
    codigo             varchar(32) not null,  
    hierarquia        int,  
    ativo_sn          bool default true,  
    constraint perfil_licenca_id primary key(id)
```

```
); create unique index perfil_licenca_nome_ix on ybatinga.perfil_licenca (nome);
```

```
create unique index perfil_licenca_codigo_ix on ybatinga.perfil_licenca (codigo);
```

---- *os pacotes de serviços adquiridos através de uma licença de usuário.*

```
drop table if exists ybatinga.servicos_pacote;
create table if not exists ybatinga.servicos_pacote (
    id                bigserial,
    nome              varchar(99) not null,
    codigo            varchar(32) not null,
    descricao         varchar(255),
    valor_unitario    numeric (10,2) default 0,
    temporal_sn       bool default true,
    ativo_sn          bool default true,
    disponivel_sn     bool default false,
    constraint servicos_pacote_id primary key(id)
);
create unique index servicos_pacote_nome_ix on ybatinga.servicos_pacote (nome);
create unique index servicos_pacote_codigo_ix on ybatinga.servicos_pacote (codigo);
```

---- *os perfis de acesso exclusivos de cada serviço em que o usuário licenciado (e sub-licenciado) possa se encaixar.*

```
drop table if exists ybatinga.perfil_servico;
create table if not exists ybatinga.perfil_servico (
    id                bigserial,
    perfil_id         int8 constraint perfil_servico_hfk references ybatinga.perfil_usuario(id),
    perfil_pai        int8 constraint perfil_servico_pai references ybatinga.perfil_servico(id),
    nome              varchar(99) not null,
    codigo            varchar(32) not null,
    hierarquia        int,
    ativo_sn          bool default true,
    disponivel_sn     bool default false,
    constraint perfil_servico_id primary key(id)
);
create unique index perfil_servico_nome_ix on ybatinga.perfil_servico (nome);
create unique index perfil_servico_codigo_ix on ybatinga.perfil_servico (codigo);
```



---- *relacionamento m:n entre a tabela servicos e a tabela servicos\_pacote (produtos)*

---- *para permitir serviços diferentes no mesmo pacote por cada perfil.*

```
drop table if exists ybatinga.pacote_servicos;
```

```
create table if not exists ybatinga.pacote_servicos (
```

```
    id                bigserial,
```

```
    pacote            int8 not null constraint pacote_servicos_mfk references ybatinga.servicos_pacote(id),
```

```
    servico           int8 not null constraint pacote_servicos_nfk references ybatinga.servico(id),
```

```
    perfil_maximo     int8 constraint pacote_servicos_perfil_mxfk references ybatinga.perfil_servico(id),
```

```
    usuarios          int,
```

```
    ativo_sn          bool default true,
```

```
    constraint pacote_servicos_id primary key(pacote, servico)
```

```
);
```

---- relacionamento m:n entre a tabela `servicos_pacote` e a tabela `perfil_licenca`

---- para permitir preços diferentes no mesmo pacote por cada perfil.

---- apesar de ser um relacionamento m:n esta tabela contém informações importantes de assinatura de serviços

```
drop table if exists ybatinga.pacote_perfis;
```

```
create table if not exists ybatinga.pacote_perfis (
```

```
    id                bigserial,
```

```
    pacote            int8 not null constraint pacote_perfis_mfk references ybatinga.servicos_pacote(id),
```

```
    perfil_licenca    int8 not null constraint pacote_perfis_nfk references ybatinga.perfil_licenca(id),
```

```
    nome              varchar(99),
```

```
    quantidade        int,
```

```
    temporal_sn       bool default true,
```

```
    relativo_sn       bool default true,
```

```
    valor_relativo    numeric(2,2) null,
```

```
    valor_fechado     numeric(10,2) null,
```

```
    ativo_sn          bool default true,
```

```
    disponivel_sn     bool default false,
```

```
    constraint pacote_perfis_id primary key(id),
```

```
    constraint pacote_perfis_valor_metodo_ck check (valor_relativo <> null or valor_fechado <> null),
```

```
    constraint pacote_perfis_valor_relativo_ck check (valor_relativo <> null and relativo_sn = true),
```

```
    constraint pacote_perfis_valor_fechado_ck check (valor_fechado <> null and relativo_sn = false)
```

```
); create unique index pacote_perfis_opcao on pacote_perfis (pacote, perfil_licenca, quantidade, relativo_sn);
```

---- as licenças que cada usuário possui para acesso como assinante dos serviços da plataforma.

```
drop table if exists ybatinga.usuario_licenca;
create table if not exists ybatinga.usuario_licenca (
    id                bigserial,
    usuario_id        int8 not null constraint usuario_licenca_fk references ybatinga.usuario(id),
    registro_dt       timestampz default current_timestamp,
    pacote_perfil_id  int8 constraint usuario_licenca_perfil_fk references ybatinga.pacote_perfis(id),
    atualizacao_dt    timestampz,
    validade_dt       timestampz,
    fechadura         varchar(32),
    chave             varchar(8),
    ativo_sn          bool default true,
    constraint usuario_licenca_id primary key(id)
);
```

---- relacionamento m:n entre a tabela usuario e a tabela usuario\_licenca

---- para permitir acesso aos serviços licenciados por sub-usuários do assinante.

```
drop table if exists ybatinga.licenca_usuarios;
create table if not exists ybatinga.licenca_usuarios (
    id                bigserial,
    usuario           int8 not null constraint licenca_usuarios_mfk references ybatinga.usuario(id),
    licenca           int8 not null constraint licenca_usuarios_nfk references ybatinga.usuario_licenca(id),
    perfil            int8 constraint licenca_usuarios_perfil_fk references ybatinga.perfil_servico(id),
    ativo_sn          bool default true,
    constraint licenca_usuarios_id primary key(usuario, licenca)
); create index licenca_usuarios_ix on ybatinga.licenca_usuarios (usuario);
```

---- *todos os registros de entrada de valores para usuário adquirir licenças e pacotes de serviços.*

```
drop table if exists ybatinga.usuario_credito;
```

```
create table if not exists ybatinga.usuario_credito (
```

```
    id                bigserial,
```

```
    usuario_id        int8 references ybatinga.usuario(id),
```

```
    registro_dt       timestamptz default current_timestamp,
```

```
    status            varchar(32) not null constraint status_credito_ls references ybatinga.lista_credito_status(chave) default 'em_aberto',
```

```
    recebido_sn        bool default false,
```

```
    liquidacao_dt     timestamptz null,
```

```
    valor             numeric(10,2) default 0,
```

```
    transacao         varchar(99),
```

```
    api              varchar(99),
```

```
        constraint usuario_credito_id primary key(id)
```

```
); create index usuario_credito_ix on ybatinga.usuario_credito (usuario_id);
```

---- *todos os registros de saída de valores do usuário pela aquisição de licenças e pacotes de serviços.*

```
drop table if exists ybatinga.usuario_debito;
```

```
create table if not exists ybatinga.usuario_debito (
```

```
    id                bigserial,
```

```
    usuario_id        int8 not null constraint usuario_debito_fk references ybatinga.usuario(id),
```

```
    momento_dt       timestamptz default current_timestamp,
```

```
    valor             numeric(10,2) default 0,
```

```
    licenca_id        int8 constraint usuario_debito_licenca_fk references ybatinga.usuario_licenca(id),
```

```
        constraint usuario_debito_id primary key(id)
```

```
); create index usuario_debito_ix on ybatinga.usuario_debito (usuario_id);
```

---- o valor financeiro disponível para usuário adquirir serviços, seja ele positivo, zerado ou negativo.

```
drop table if exists ybatinga.usuario_saldo;
create table if not exists ybatinga.usuario_saldo (
    id                bigserial,
    usuario_id        int8 not null constraint usuario_saldo_fk references ybatinga.usuario(id),
    valor             numeric(10,2) default 0,
    data              timestamptz,
    ultimo_credito     int8 not null constraint usuario_ultimo_credito_fk references ybatinga.usuario_credito(id),
    ultimo_debito      int8 not null constraint usuario_ultimo_debito_fk references ybatinga.usuario_debito(id),
    constraint usuario_saldo_id primary key(id)
);
create index usuario_saldo_ix on ybatinga.usuario_saldo (usuario_id);
```

---- tipos de arquivos que se referenciados na URL são passados como parâmetro do serviço correspondente.

```
drop table if exists ybatinga.servico_arquivos;
create table if not exists ybatinga.servico_arquivos (
    id                bigserial,
    servico_id         int8 not null constraint servico_arquivo_fk references ybatinga.servico(id),
    descricao          varchar(255),
    extensao           varchar(32) not null,
    tipo_mime          varchar(32),
    parametros         varchar(255),
    arquivo_log        varchar(255),
    ativo_sn           bool default true,
    constraint servico_arquivos_id primary key(id)
);
create index servico_arquivos_ix on ybatinga.servico_arquivos(servico_id);
```

---- *programações de execução agendada de serviços dentro da plataforma.*

```
drop table if exists ybatinga.servico_agendamentos;
```

```
create table if not exists ybatinga.servico_agendamentos (  
    id                bigserial,  
    servico_id        int8 not null constraint servico_agendamento_fk references ybatinga.servico(id),  
    usuario_id        int8 not null constraint servico_agendamento_usuario_fk references ybatinga.usuario(id),  
    procedure_id       int8 null constraint servico_agendamento_procedure_fk references ybatinga.procedure(id),  
    procedure_sn       bool default false,  
    descricao          varchar(255),  
    expressao_tempo__cron varchar(255) not null,  
    ciclico_sn         bool default true,  
    tentar_novamente_sn bool default false,  
    tentar_ate_novo_sn  bool default false,  
    houve_falha_sn     bool default false,  
    parametros         varchar(255),  
    arquivo_log        varchar(255),  
    habilitado_sn      bool default true,  
    ativo_sn           bool default true,  
    constraint servico_agendamentos_id primary key(id)  
); create index servico_agendamentos_ix on ybatinga.servico_agendamentos(servico_id);
```

```
VACUUM FULL;
```