

Confirmando e liberando acesso do usuário

Tendo enviado o email de cadastro para o usuário, precisamos confirmar o cadastro! Já temos a rota definida, então vamos criá-la e criar também a lógica do controller.

```
GET /usuario/confirma/:email/:codigo controllers.UsuarioController.confirmaCadastro(email, cod:
```

```
@Inject
private TokenDeCadastroDAO tokenDeCadastroDAO;
@Inject
private UsuarioDAO usuarioDAO;

public Result confirmaCadastro(String email, String codigo) {
    Optional<TokenDeCadastro> possivelToken = tokenDeCadastroDAO.comCodigo(codigo);
    Optional<Usuario> possivelUsuario = usuarioDAO.comEmail(email);
    return ok();
}
```

Já que precisamos de um DAO para o token de cadastro, já podemos criá-lo também.

```
package daos;
public class TokenDeCadastroDAO {
    private Finder<Long, TokenDeCadastro> tokens = new Finder<>(TokenDeCadastro.class);
    public Optional<TokenDeCadastro> comCodigo(String codigo) {
        TokenDeCadastro token = tokens.where().eq("codigo", codigo).findUnique();
        return Optional.ofNullable(token);
    }
}
```

A idéia ao passar o email junto com o token é fazer uma verificação mais cuidadosa: o token precisa ser válido, o email precisa ser válido e o usuário atribuído ao token precisa ser o correto, ou seja, seu email precisa ser igual ao email passado no link!

```
public Result confirmaCadastro(String email, String codigo) {
    Optional<TokenDeCadastro> possivelToken = tokenDeCadastroDAO.comCodigo(codigo);
    Optional<Usuario> possivelUsuario = usuarioDAO.comEmail(email);
    if (possivelToken.isPresent() && possivelUsuario.isPresent()) {
        TokenDeCadastro token = possivelToken.get();
        Usuario usuario = possivelUsuario.get();
        if (token.getUsuario().equals(usuario)) {
            // sucesso
        }
    }
    // fracasso
    return ok();
}
```

Agora só precisamos decidir o que acontece em cada caso: se a validação falhou, mostramos uma mensagem de erro e redirecionamos o usuário para a página de login; se for bem sucedida apagamos o token, marcamos o usuário como verificado, mandamos uma mensagem de sucesso, fazemos login por ele (ainda não vai acontecer pois a lógica ainda não está pronta) e o redirecionamos para o painel de usuário (que ainda não existe, como o login).

```
public Result confirmaCadastro(String email, String codigo) {
    Optional<TokenDeCadastro> possivelToken = tokenDeCadastroDAO.comCodigo(codigo);
    Optional<Usuario> possivelUsuario = usuarioDAO.comEmail(email);
    if (possivelToken.isPresent() && possivelUsuario.isPresent()) {
        TokenDeCadastro token = possivelToken.get();
        Usuario usuario = possivelUsuario.get();
        if (token.getUsuario().equals(usuario)) {
            token.delete();
            usuario.setVerificado(true);
            usuario.update();
            flash("success", "Seu usuário foi confirmado com sucesso! Bem vindo!");
            // TODO logar usuario
            return redirect("/usuario/painel"); // TODO rota
        }
    }
    flash("danger", "Algo deu errado ao tentar confirmar seu cadastro.")
    return redirect("/login"); // TODO rota
}
```

E pronto! Já podemos experimentar clicar no link do email de cadastro para ver o que acontece!