

Cadastro de usuários

Agora que temos um modelo de **Usuário**, podemos criar o formulário de cadastro

app/views/formularioDeNovoUsuario.scala.html .

```
@(formulario: Form[Usuario])
@import b3.vertical.fieldConstructor;
@main("Cadastro de usuario") {
  @b3.form(routes.UsuarioController.salvaNovoUsuario) {
    <h1>Cadastre-se!</h1>
    @b3.text(formulario("nome"), '_label -> "Nome")
    @b3.email(formulario("email"), '_label -> "Email")
    @b3.password(formulario("senha"), '_label -> "Senha")
    @b3.submit('class -> "btn btn-primary") { Cadastrar }
  }
}
```

Vamos então criar as rotas para o formulário e para salvar o **Usuário**.

```
GET /usuario/novo controllers.UsuarioController.formularioDeNovoUsuario
POST /usuario/novo controllers.UsuarioController.salvaNovoUsuario
```

E depois o controller com as lógicas básicas. Após o cadastro redirecionamos o usuário para a tela de login, mas como ela ainda não existe deixamos a rota como `String` e marcada para fazer depois.

```
package controllers;
import javax.inject.*;
import play.data.*;
import play.mvc.*;
import views.html.*;

public class UsuarioController extends Controller {
  @Inject
  private FormFactory formularios;
  public Result formularioDeNovoUsuario() {
    Form<Usuario> formulario = formularios.form(Usuario.class);
    return ok(formularioDeNovoUsuario.render(formulario));
  }
  public Result salvaNovoUsuario() {
    Form<Usuario> formulario = formularios.form(Usuario.class).bindFromRequest();
    Usuario usuario = formulario.get();
    usuario.save();
    flash("success", "Usuário cadastrado com sucesso");
    return redirect("/login"); // TODO rota
  }
}
```

Agora conseguimos cadastrar novos usuários!

