Create an MVP with Phoenix Framework

Hi!

My name is **Iván González**, a.k.a. **dreamingechoes**

I **work** @ bizneo.com
I **tweet** @ twitter.com/dreamingechoes
I **code** @ github.com/dreamingechoes
I **write** @ dreamingecho.es

The Idea

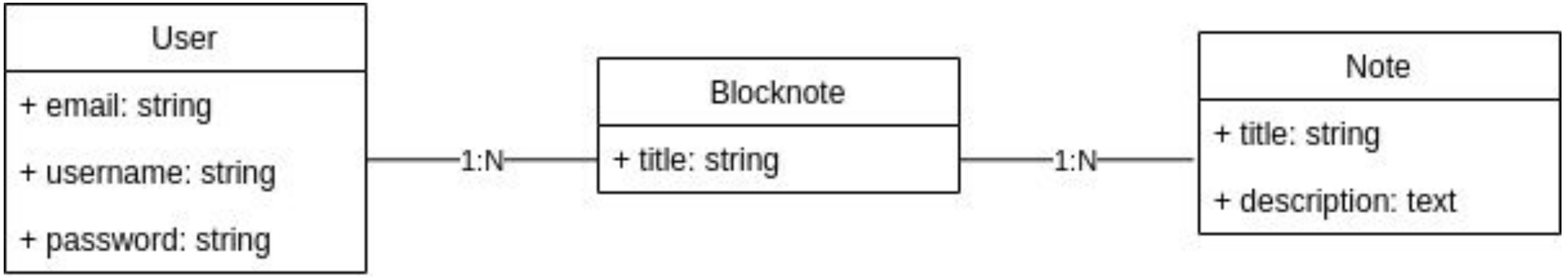
The Idea

The **product** we want to create will be a **simple application** which with we can sync all our **notes** in the cloud.

For the **MVP**, we just need to store the **user's information**, a series of **blocknotes** created by the user, and the **notes** which belongs to the blocknotes.

Always **start with something simple**, and growth on and on.

The Idea


User
+ email: string
+ username: string
+ password: string
1:N
Blocknote
+ title: string
1:N
Note
+ title: string
+ description: text

The Idea

*Patent pending

Environment Setup

Environment Setup

We'll need to install the following **dependencies:**

Erlang

Elixir

Phoenix Framework

Node.js

PostgreSQL

Environment Setup

NATIVELY

Erlang: erlang.org/doc/installation_guide/INSTALL.html

Elixir: elixir-lang.org/install.html

Phoenix Framework: hexdocs.pm/phoenix/installation.html

Node.js: nodejs.org/en/download/package-manager

PostgreSQL: postgresql.org/docs/current/static/tutorial-install.html

Environment Setup

VERSION MANAGER

asdf: Extendable version manager with support for **Ruby**, **Node.js**, **Elixir**, **Erlang** & more.

github.com/asdf-vm/asdf

Environment Setup

DOCKER

We can **recreate** the setup of our servers to **work in development** in the **same conditions.**

Key files: Dockerfile and **docker-compose.yml**

docs.docker.com/install

github.com/dreamingechoes/docker-elixir-phoenix

Phoenix Application Structure

Phoenix Application Structure

CONFIG FILES

We have a series of files in the **root** of the **application**:

mix.exs: main application configuration.

.formatter.exs: configuration for the **mix format** task.

.credo.exs: configuration for **Credo**.

Phoenix Application Structure

ASSETS

Folder with all the stuff related with the **application's assets**:

Brunch config (brunch is used by default by Phoenix)

CSS

Javascript

Static files

Phoenix Application Structure

APPLICATION CONFIG

Folder with all the definitions of the main **application configuration**, as well as the different **environments:**

config.exs

dev.exs

prod.exs

test.exs

Phoenix Application Structure

LIB

Contains the **core** of the application, both **business logic** and **web implementation.**

Phoenix Application Structure

PRIV

Contains **internationalization** and **database** files:

gettext folder: internationalization files.

repo folder: **migrations** and database **seeds**.

Phoenix Application Structure

TEST

All about application tests, both **unit** and **integration**.

Modeling The Business Logic

Modeling The Business Logic

Inside the **LIB** folder, we have a place in which we can define all the **business logic** of our application.

Contexts: hexdocs.pm/phoenix/contexts.html

Schemas: hexdocs.pm/ecto/Ecto.Schema.html

application.ex: hexdocs.pm/elixir/Application.html

repo.ex

Building The Web

Building The Web

Inside the **LIB** folder, we have a place in which we can develop the final web which with will interact the final user.

Controllers: hexdocs.pm/phoenix/controllers.html

Templates: hexdocs.pm/phoenix/templates.html

Views: hexdocs.pm/phoenix/views.html

router.ex: hexdocs.pm/phoenix/Phoenix.Router.html

Shipping To Production

Shipping To Production

DISTILLERY

Simplify deployments in **Elixir**. This is useful if we have a server and we want to **deploy** our application in there.

Distillery will compile the application with the specified configuration, **generate the binaries** and **push it to our server**.

github.com/bitwalker/distillery

Shipping To Production

HEROKU

Heroku provides great integration with **Phoenix** applications, with buildpacks specifically designed for **Elixir** projects.

It’s the **best method** to start a **MVP**. Free **server**, free **database** storage, **zero server configuration.**

heroku.com

Shipping To Production

OTHER ALTERNATIVES

Amazon AWS: aws.amazon.com

Google Cloud: cloud.google.com

Gigalixir: gigalixir.com

Final Product

Final Product

DEMO

That’s all folks!

Thank you! <3