



## AULA 08 - Vídeo 04

### Desenvolvimento

Como vimos no último vídeo o “cadastrar” já está fazendo a criptografia da senha, agora precisamos garantir que ele está fazendo a busca certificando que são as mesmas senhas na hora da comparação, e para isso usaremos um método do **bcrypt** chamado de “compare”

Então agora iremos corrigir a comparação de senha do sistema quando ela for devolvida, como os dados do usuário está no arquivo “perfil service” tanto a parte de senha quanto a de login. Como no arquivo “perfil” temos o email de usuário como **unique**, entendemos que somente um email é válido para o usuário e sendo assim podemos buscar o perfil por esse dado único. Apagaremos o perfil de usuário e senha, pois agora não temos mais filtro para array, agora buscamos no banco de dados.

trazendo agora os dados do “perfilModel” para usar o await na busca.

```
const perfilModel = require("../models/Perfil")

module.exports = {
  autenticar: async (req, res) =>{
    let usuario = req.body

    if(usuario.email && usuario.senha){
      let usuarioEncontrado = await perfilModel.findOne({

        usuario: {
          email: usuario.email
        }
      })
    }
  }
}
```

```
}
```

Para fazermos a comparação de senha com o **bcrypt** temos antes que configurar a função para que não seja usado o token fixo, pois isso será gerado pela nossa **json web token**. Buscando o “Perfil” do usuário.

```
const perfilModel = require("../models/Perfil")
const bcrypt = require("bcrypt")

module exports = {
  autenticar: async (req, res) => {
    let usuario = req.body

    if (usuario.email && usuario.senha) {
      let perfilEncontrado = await perfilModel.findOne({
        usuario: {
          email: usuario.email
        }
      })

      if (perfilEncontrado) {
        const match = await bcrypt.compare(usuario.senha,
perfilEncontrado.usuario.senha)

        if (match) {
          //LOGIN
        } else {
          //vetar a passagem
        }
      }
    }
  }
}
```