**Session e Middleware**

**Instalando o express-session**

**- npm install express-session**

**index.js:**

//*Importando gerador de sessões do express*

*import* session *from* "express-session"

**index.js:**

// *Configurando o express-session*

*app.use*(*session*({

  secret: "lojasecret",

  cookie: {maxAge: 30000}, //*Sessão expira em 30 segundos (mudar depois)*

  saveUninitialized: false,

  resave: false

}))

**UsersController.js:**

*if*(correct){

        // *AUTORIZA O LOGIN*

        req*.session.*user = {

          id: user*.*\_id,

          email: user*.*email

        }

        res*.send*(`Usuário logado:<br> ID: ${req*.session.*user['id']} <br> E-mail: ${req*.session.*user['email']}`)

Texto

Descrição gerada automaticamente

**UsersController.js:**

*if*(correct){

        // *AUTORIZA O LOGIN*

        req*.session.*user = {

          id: user*.*id,

          email: user*.*email

        }

        res*.redirect*("/")

**Middleware**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Auth.js:**

function *Auth*(req, res, next) {

*if*(req*.session.*user *!=* undefined){

*next*()

    }*else*{

        res*.render*("login")

    }

}

*export* *default* *Auth*

**ClientesController.js**

*import* Auth *from* "../middleware/Auth.js"

// *ROTA CLIENTES*

router*.get*("/clientes",  *Auth*, (req, res) => {

*ClienteService.SelectAll*()*.then*((clientes) => {

      res*.render*("clientes", {

        clientes: clientes,

      })

    })

  })

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**index.js:**

// *Configurando o express-session*

*app.use*(*session*({

  secret: "lojasecret",

  cookie: {maxAge: 3600000}, //*Sessão expira em 1 hora*

  saveUninitialized: false,

  resave: false

}))