Gestor_Contrasenas

Hemos creado los certificados

usando: **Key considerations for algorithm "RSA" ≥ 2048-bit**
`openssl genrsa -out server.key 2048`

** Key considerations for algorithm "ECDSA" ≥ secp384r1 List ECDSA the supported curves (openssl ecparam -list_curves)** `openssl ecparam -genkey -name secp384r1 -out server.key`
Generation of self-signed(x509) public key (PEM-encodings .peml.crt) based on the private (.key)

`openssl req -new -x509 -sha256 -key server.key -out server.crt -days 3650`

```
func main() {
    log.SetFlags(log.Lshortfile)

    cer, _ := tls.LoadX509KeyPair("server.crt", "server.key")

    config := &tls.Config{Certificates: []tls.Certificate{cer}}
    ln, _ := tls.Listen("tcp", ":443", config)

    defer ln.Close()

    for {
        conn, _ := ln.Accept()

        go handleConnection(conn)
    }
}

//conexion entre el cliente y el servidor
func handleConnection(conn net.Conn) {
    defer conn.Close()
    r := bufio.NewReader(conn)
    for {
        msg, _ := r.ReadString('\n')

        println("Mensaje recibido:")
        println(msg)
        println("mensaje a responder(enviar):")
        var linea string
        fmt.Scanf("%s\n", &linea)

        conn.Write([]byte(linea + "\n"))
        //n, _err := conn.Write([]byte(linea + "\n"))

    }

}
```