**AWS SECRET MANAGER KEYS**

Debemos crear un Secret Manager y configurar nuestros proyectos para que lo utilicen.

El siguiente ejemplo utiliza un Secret Manager para poder recuperar las claves sin que estén expuestas en la aplicación.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

{

"MySql":"server=awsmysqlpaco.ctinakxu7pmf.us-east-1.rds.amazonaws.com;database=personajesaws;user id=adminsql;password=Admin123",

"ApiPersonajes":<https://4h2rkz2jy0.execute-api.us-east-1.amazonaws.com/Prod/>

,"S3BucketName:":"bucket-personajes-labs"

,"usermail":[micorreotestingcloudpgs@hotmail.com](mailto:micorreotestingcloudpgs@hotmail.com)

,"passwordmail":"Hotmail12345"

,"host":"smtp.office365.com"

,"S3BucketUrl":<https://bucket-personajes-labs.s3.amazonaws.com/>

}

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El siguiente paso es utilizar nuestro **Api Gateway** (Por ejemplo) para leer los secretos.

Vamos a modificar el proyecto **MvcPersonajes**

Agregamos el siguiente Nuget

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Sobre **Models**, creamos una nueva clase llamada **KeysModel**

**KEYSMODEL**

**public** **class** KeysModel

{

**public** **string** MySql { **get**; **set**; }

**public** **string** ApiPersonajes { **get**; **set**; }

**public** **string** S3BucketName { **get**; **set**; }

**public** **string** S3BucketUrl { **get**; **set**; }

**public** **string** UserMail { **get**; **set**; }

**public** **string** PasswordMail { **get**; **set**; }

**public** **string** Host { **get**; **set**; }

}