Online Payslip Service

ROSCLAR

Política de Seguridad

Portal del empleado

Tabla de contenido

Contenido

Objeto del documento

El presente documento tiene por objeto establecer la Política de Seguridad de la Información de Rosclar en relación al servicio prestado por Rosclar acerca de la aplicación a3HRgo, basada en los requisitos dispuestos en el estándar ISO/IEC 27001:2013.

Objetivo de la seguridad

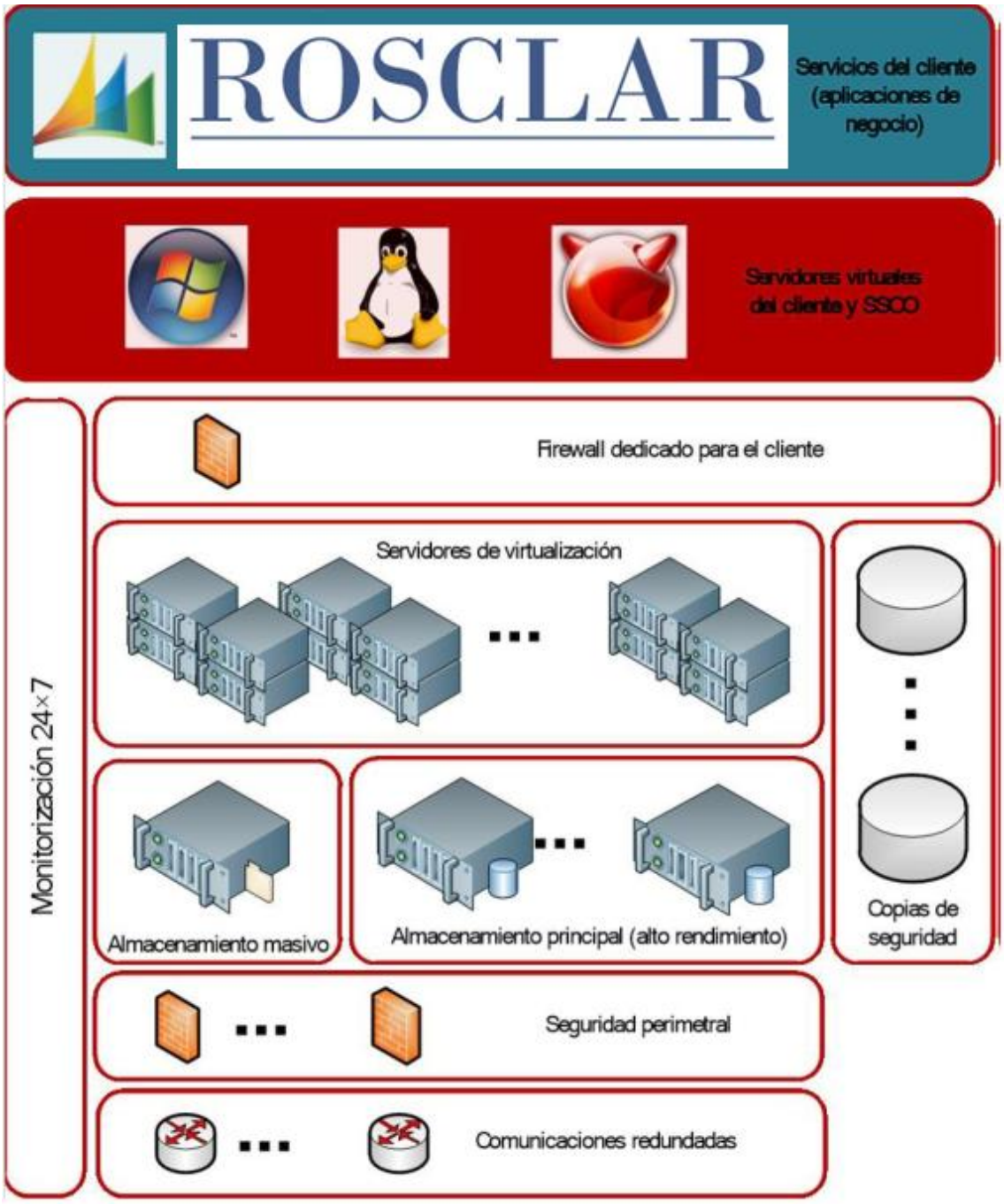
Tiene por objetivo asegurar que todos los tratamientos de información relativos a los procesos de negocio indicados en el alcance se realizan de forma segura y únicamente por personal autorizado.
De esta forma se garantizan la confidencialidad, la integridad y la disponibilidad de la información de los clientes de Rosclar.

Seguridad

Protección física del servicio

Los elementos físicos esenciales para ofrecer el servicio, como son los servidores físicos o los sistemas de almacenamiento principales, se encuentran redundados y configurados en alta disponibilidad, según sea adecuado dada su naturaleza. Se dispone de recursos suficientes para que los fallos simples en elementos hardware no afecten de forma significativa al servicio mientras las anomalías estén presentes.
La infraestructura física para alojar el servicio IaaS de Rosclar se encuentra alojada en centros de datos que ofrecen las características necesarias para asegurar:

- la temperatura correcta de los distintos elementos hardware;
- la humedad adecuada del ambiente;
- la disposición de un ambiente libre de elementos contaminantes como polvo en suspensión;
- la disponibilidad de energía de forma ininterrumpida, incluso en caso de fallo del suministro eléctrico principal;
- la redundancia de las comunicaciones con operadores diferentes;
- la protección adecuada ante eventos que puedan causar daño a las infraestructuras, como incendios;
- la seguridad física necesaria en los edificios, las salas, los pasillos y los armarios, con controles de acceso 24×7.



Protección lógica del servicio

Se previene el acceso no autorizado a los sistemas operativos, y se requiere su actualización para corregir vulnerabilidades detectadas; se proveerán de las medidas técnicas de seguridad oportunas.
Estará restringido y controlado el uso de aplicaciones no autorizadas que puedan invalidar las medidas de seguridad implantadas.

Seguridad ligada a los recursos humanos

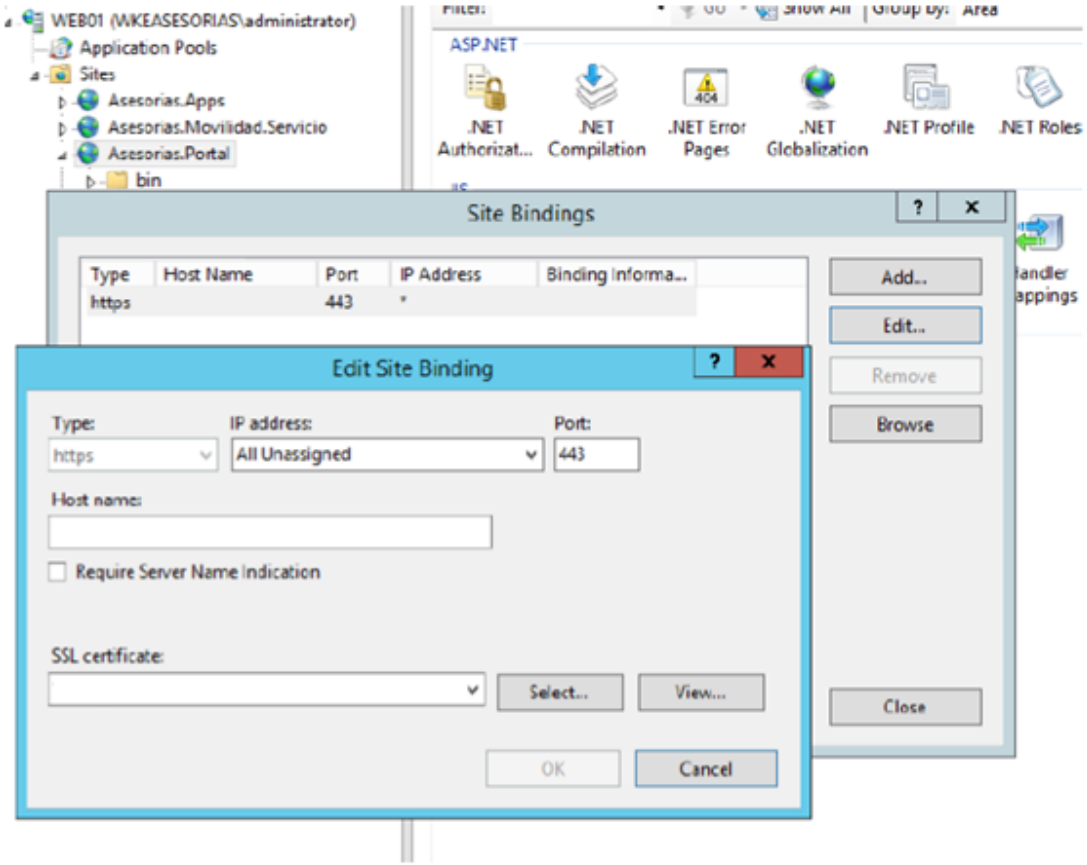
Se asegura que todos los empleados, contratistas y los terceros entienden sus responsabilidades y son adecuados para llevar a cabo las funciones que les corresponden, así como para reducir el riesgo de robo, fraude o de uso indebido de los recursos puestos a su disposición.
Se asegura que todos los empleados, contratistas y los terceros son conscientes de las amenazas y problemas que afectan a la seguridad de la información y de sus responsabilidades y obligaciones, y de que están preparados para cumplir la política de seguridad de la organización en el desarrollo habitual de su trabajo, y para reducir el riesgo de error humano. Se asegura que todos los empleados, contratistas y los terceros abandonan la organización o cambian de puesto de trabajo de forma ordenada y sin comprometer la seguridad de la misma.

Política de credenciales de acceso a aplicación

A continuación se muestran una serie de pruebas de todos los pasos que afectan a la política de credenciales de acceso, que sea segura, que las contraseñas tengan la complejidad requerida, y todo lo relativo a bloqueo de cuentas y sesiones inactivas.

Acceso HTTPS/SSL

En la imagen se puede ver la configuración relativa a los protocolos seguros HTTPS y SSL que tienen los servidores.



Generación de contraseñas y su gestión

La emisión de contraseñas automáticas genera una contraseña de 10 caracteres siguiendo el siguiente patrón:

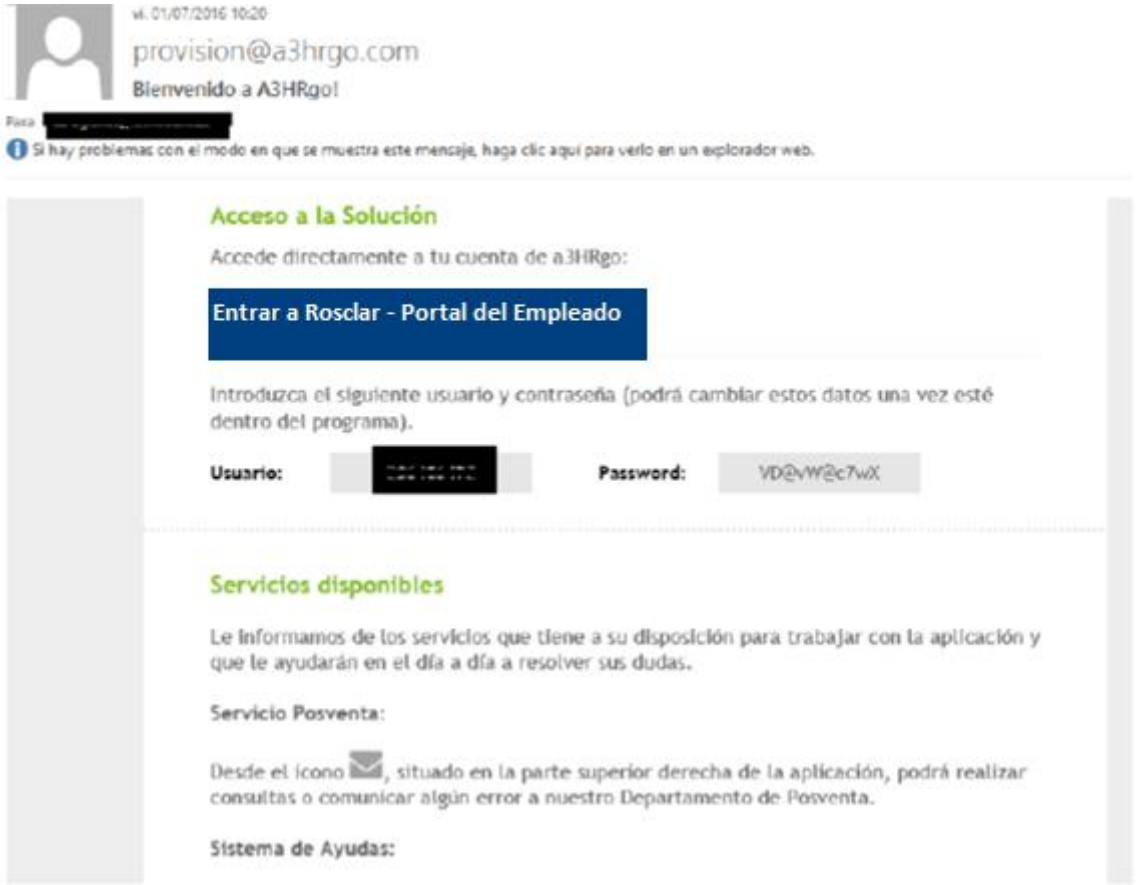
- Utiliza el método de .NET System.Web.Security.Membership.GeneratePassword (https://msdn.microsoft.com/es-es/library/system.web.security.membership.generatepassword(v=vs.110).aspx) pasando como parámetros 7 y 1 (7 caracteres de contraseña y al menos un carácter no alfanumérico
- Sustituye el carácter no alfanumérico generado por el carácter '@' (Regex.Replace(result, "[^0-9a-zA-Z]+", "@")
- Añade al final '7wX' (para forzar que haya siempre un número, una letra minúscula y una letra mayúscula).

Prueba del proceso de creación de contraseñas aleatorias

En la imagen se muestra el código que genera las contraseñas aleatorias tal y como se han descrito.

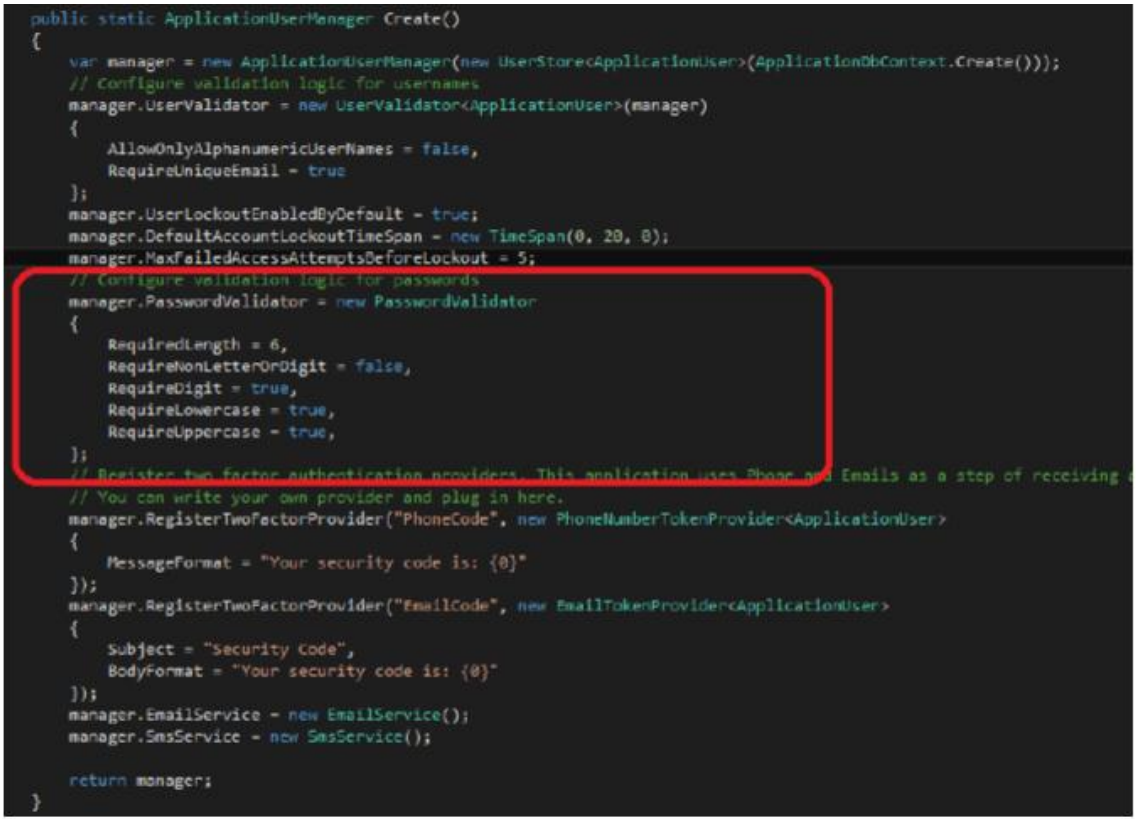
```
public static string GeneratePassword()
{
    string result = string.Empty;

    result = System.Web.Security.Membership.GeneratePassword(7, 0);
    result = Regex.Replace(result, "[^0-9a-zA-Z]+", "@");
    result += "7wX";

    return result;
}
```

Y una muestra del email que se envía:



Prueba de la política de contraseñas implementada

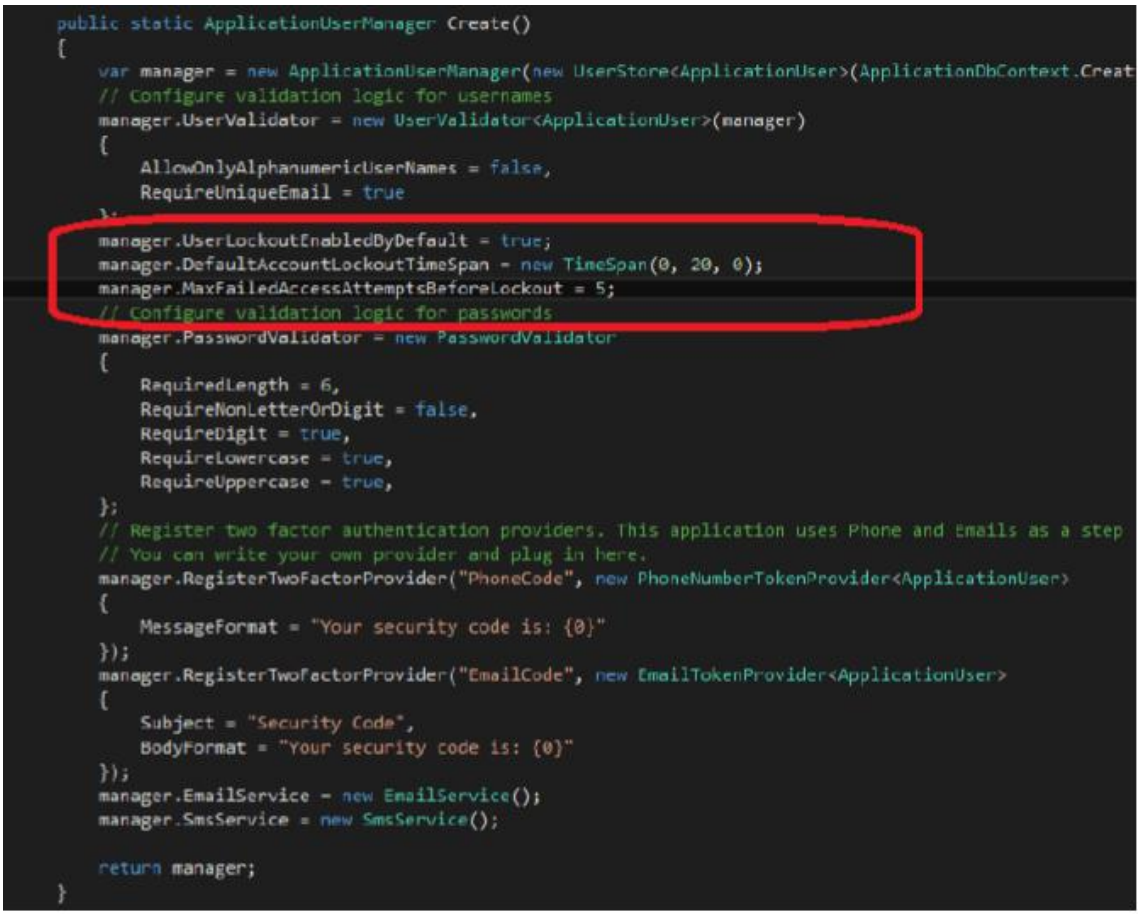
En la imagen se muestra el código que rige la política de complejidad de las contraseñas



```
public static ApplicationUserManager Create()
{
    var manager = new ApplicationUserManager(new UserStore<ApplicationUser>(ApplicationDbContext.Create()));
    // Configure validation logic for usernames
    manager.UserValidator = new UserValidator<ApplicationUser>(manager)
    {
        AllowOnlyAlphanumericUserNames = false,
        RequireUniqueEmail = true
    };
    manager.UserLockoutEnabledByDefault = true;
    manager.DefaultAccountLockoutTimeSpan = new TimeSpan(0, 20, 0);
    manager.MaxFailedAccessAttemptsBeforeLockout = 5;
    // Configure validation logic for passwords
    manager.PasswordValidator = new PasswordValidator
    {
        RequiredLength = 6,
        RequireNonLetterOrDigit = false,
        RequireDigit = true,
        RequireLowercase = true,
        RequireUppercase = true,
    };
    // Register two factor authentication providers. This application uses Phone and Emails as a step of receiving
    // You can write your own provider and plug in here.
    manager.RegisterTwoFactorProvider("PhoneCode", new PhoneNumberTokenProvider<ApplicationUser>
    {
        MessageFormat = "Your security code is: {0}"
    });
    manager.RegisterTwoFactorProvider("EmailCode", new EmailTokenProvider<ApplicationUser>
    {
        Subject = "Security Code",
        BodyFormat = "Your security code is: {0}"
    });
    manager.EmailService = new EmailService();
    manager.SmsService = new SmsService();

    return manager;
}
```

Prueba de la política de bloqueo de cuentas de usuario.

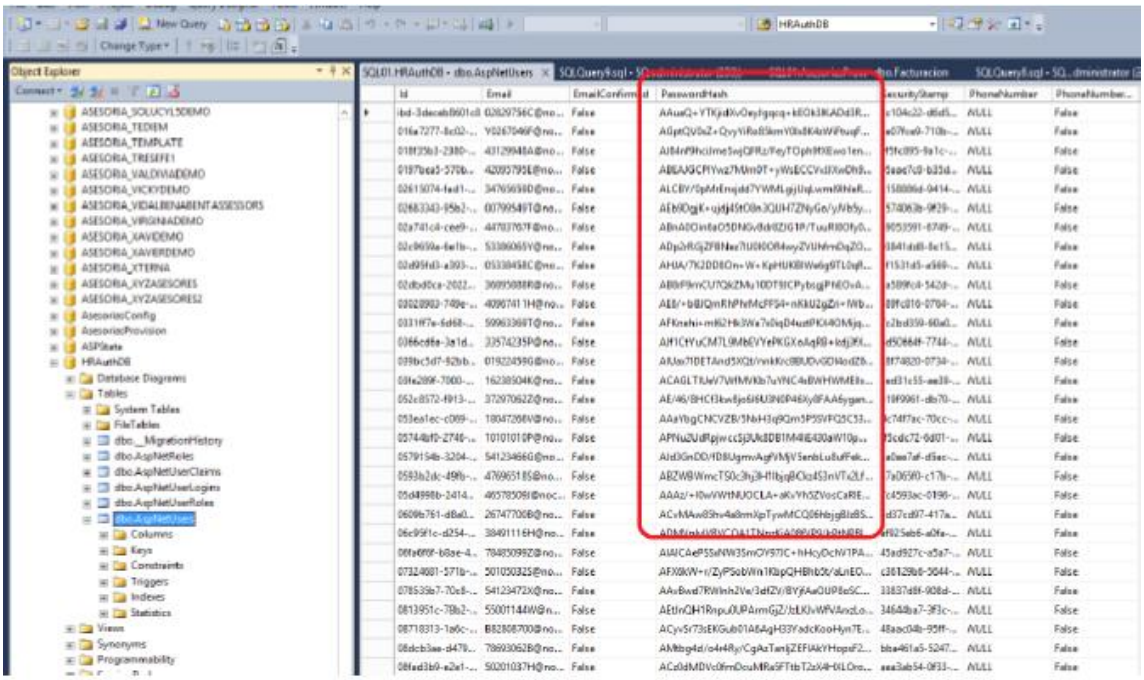
En la imagen se muestra el código que implementa el bloqueo de cuentas y el tiempo necesario de inactividad para ejecutarlo. 1 0



```
public static ApplicationUserManager Create()
{
    var manager = new ApplicationUserManager(new UserStore<ApplicationUser>(ApplicationDbContext.Creat
    // Configure validation logic for usernames
    manager.UserValidator = new UserValidator<ApplicationUser>(manager)
    {
        AllowOnlyAlphanumericUserNames = false,
        RequireUniqueEmail = true
    };
    manager.UserLockoutEnabledByDefault = true;
    manager.DefaultAccountLockoutTimeSpan = new TimeSpan(0, 20, 0);
    manager.MaxFailedAccessAttemptsBeforeLockout = 5;
    // Configure validation logic for passwords
    manager.PasswordValidator = new PasswordValidator
    {
        RequiredLength = 6,
        RequireNonLetterOrDigit = false,
        RequireDigit = true,
        RequireLowercase = true,
        RequireUppercase = true,
    };
    // Register two factor authentication providers. This application uses Phone and Emails as a step
    // You can write your own provider and plug in here.
    manager.RegisterTwoFactorProvider("PhoneCode", new PhoneNumberTokenProvider<ApplicationUser>
    {
        MessageFormat = "Your security code is: {0}"
    });
    manager.RegisterTwoFactorProvider("EmailCode", new EmailTokenProvider<ApplicationUser>
    {
        Subject = "Security Code",
        BodyFormat = "Your security code is: {0}"
    });
    manager.EmailService = new EmailService();
    manager.SmsService = new SmsService();

    return manager;
}
```

Prueba que las contraseñas se guardan hasheadas en base de datos

En la imagen se puede ver que las contraseñas no se guardan en base de datos como texto plano, si no que se guardan hasheadas, para mayor seguridad.



Prueba de la política de sesiones inactivas

En la imagen se puede ver el código que limita el tiempo de inactividad de la sesión antes de considerarla como expirada.

```
public void ConfigureAuth(IAppBuilder app)
{
    // Configure the db context and user manager to use a single instance per request
    app.CreatePerOwinContext(ApplicationDbContext.Create);
    app.CreatePerOwinContext<ApplicationUserManager>(ApplicationUserManager.Create);

    // Enable the application to use a cookie to store information for the signed in user
    // and to use a cookie to temporarily store information about a user logging in with a third party login provider
    // Configure the sign in cookie
    app.UseCookieAuthentication(new CookieAuthenticationOptions
    {
        AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
        LoginPath = new PathString("/Account/Login"),
        Provider = new CookieAuthenticationProvider
        {
            OnValidateIdentity = SecurityStampValidator.OnValidateIdentity<ApplicationUserManager, ApplicationUser>(
                validateInterval: TimeSpan.FromMinutes(20),
                regenerateIdentity: (manager, user) => user.GenerateUserIdentityAsync(manager))
        },
        ExpireTimeSpan = TimeSpan.FromMinutes(20)
    });
```

Política de backup de los datos de nuestros clientes

De manera estándar se realizan copias de seguridad de la infraestructura virtual del cliente (hardware virtual) mediante el método de *snapshot*. Estas copias de seguridad permiten recuperar una máquina de forma rápida.

La periodicidad de la realización de copias de seguridad estándar y sus **periodos de retención**

se explican a continuación:

- se realizará una copia de seguridad mediante el método de *snapshot* cada noche;
- se guardarán las copias de seguridad de los últimos 14 días;

Gestión de comunicaciones y operaciones

La política de seguridad relativa a la gestión de comunicaciones y operaciones se basa en las siguientes partes:

- asegurar el funcionamiento correcto y seguro de los sistemas de información;
- implantar y mantener el nivel apropiado de seguridad de la información en la provisión de servicios, manteniendo la consonancia con los posibles acuerdos de provisión de servicios de terceros;
- minimizar el riesgo de fallos de los sistemas;
- proteger la integridad del software y de la información;
- mantener la integridad y disponibilidad de la información y de los recursos de tratamiento de la información;
- asegurar la protección de la información en las redes y la protección de la infraestructura de soporte;
- evitar la revelación, modificación, retirada o destrucción no autorizada de activos, y la interrupción de las actividades de los clientes;
- mantener la seguridad de la información y del software intercambiado dentro de la organización y con un tercero;
- detectar las actividades de procesamiento de información no autorizadas.

Plan de continuidad de negocio

Establece las acciones necesarias, y los responsables de llevarlas a cabo, para que la organización pueda recuperarse de incidentes de gravedad, tales como robos, incendios, terremotos, inundaciones, etc., con la mayor brevedad posible.
Estas acciones y sus responsables quedan fuera del alcance de este documento para evitar transmitir información confidencial.

Certificaciones de seguirdad y valor añadido

Contamos con un SGSI (Sistema de Gestión de Seguridad de la Información) de acuerdo a las buenas prácticas que marca la norma ISO 27000.
Tenemos planificado la obtención de la certificación por la norma UNE/ISO 27001 para el presente año. Además, contamos con personal con certificaciones ISACA, CCIE.

Cumplimiento legal

En Rosclar cumplimos con la normativa legal vigente, estando adaptados sus procesos y datos a la LOPD (Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal, Real Decreto 1720/2007, de 21 de diciembre, de desarrollo de la LOPD).

ROSCLAR