

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PROTOTIPO
PARA EL CAMBIO DE CONTRASEÑA DE UNA CUENTA DEL
DIRECTORIO ACTIVO, SINCRONIZADO CON OFFICE 365,
UTILIZANDO MECANISMOS DE VALIDACIÓN DE IDENTIDAD**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

**David Eduardo Abad Dávila
david.abad@epn.edu.ec**

**Director: Ing. Gabriel Roberto López Fonseca, MSc.
gabriel.lopez@epn.edu.ec**

**Codirector: Ing. Franklin Leonel Sánchez Catota MSc.
franklin.sanchez@epn.edu.ec**

Quito, Julio 2019

DECLARACIÓN

Yo, David Eduardo Abad Dávila, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

David Eduardo Abad Dávila

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por David Eduardo Abad Dávila, bajo nuestra supervisión.

Ing. Gabriel López, MSc.

DIRECTOR

Ing. Franklin Sanchez, MSc.

CODIRECTOR

AGRADECIMIENTOS

Agradezco a mi madre Gina por brindarme su paciencia y ánimos, a mis hermanos por José Miguel y Juan Camilo por su apoyo, a mis compañeros por su amistad, a Daniela Arévalo por su infinita paciencia y amor incondicional, a los Ing. Gabriel López, y Franklin Sánchez por su guía para el desarrollo de la presente tesis.

David Eduardo Abad Dávila

DEDICATORIA

Dedico este trabajo, a mi padre German Abad, que desde el cielo me brindó las fuerzas necesarias para salir adelante cada día, y poder llegar a este momento tan especial de mi vida, a mi madre por su paciencia, amor que nunca me faltaron de su parte.

David Eduardo Abad Dávila

ÍNDICE DE CONTENIDO

DECLARACIÓN	I
CERTIFICACIÓN	II
AGRADECIMIENTOS	III
DEDICATORIA.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	IX
LISTA DE TABLAS	XIII
LISTA DE CÓDIGOS	XIV
LISTA DE ANEXOS	XVI
RESUMEN	XVII
PRESENTACIÓN.....	XVIII
CAPÍTULO 1	1
1 FUNDAMENTO TEÓRICO.....	1
1.1 TECNOLOGÍAS UTILIZADAS	1
1.1.1 ASP.NET.....	1
1.1.1.1 MARCO DE TRABAJO DE PÁGINA Y CONTROLES	2
1.1.1.2 COMPILADOR DE ASP.NET	3
1.1.1.3 INFRAESTRUCTURA DE SEGURIDAD	3
1.1.1.4 FACILIDADES DE ADMINISTRACIÓN DE ESTADO.....	4
1.1.1.5 CONFIGURACIÓN DE ASP.NET	4
1.1.1.6 SUPERVISIÓN DE ESTADO Y CARACTERÍSTICAS DE RENDIMIENTO	5
1.1.1.7 CAPACIDAD DE DEPURACIÓN.....	6
1.1.1.8 FRAMEWORK DE SERVICIOS WEB XML	6
1.1.1.9 ENTORNO DE HOST EXTENSIBLE Y ADMINISTRACIÓN DEL CICLO DE VIDA DE LAS APLICACIONES	7
1.1.1.10 ENTORNO DE DISEÑADOR EXTENSIBLE	7
1.1.1.11 APLICACIÓN WEB DE ASP.NET	9
1.1.2 .NET FRAMEWORK	11
1.2 SERVICIO DE DIRECTORIO ACTIVO DE MICROSOFT.....	12
1.2.1 COMPONENTES FÍSICOS.....	13
1.2.1.1 ALMACENAMIENTO DE DATOS.....	13
1.2.1.2 CONTROLADORES DE DOMINIO	15
1.2.1.3 CATÁLOGO GLOBAL.....	15
1.2.1.4 RODCS.....	16
1.2.2 COMPONENTES LÓGICOS.....	16
1.2.2.1 PARTICIONES.....	16
1.2.2.2 DOMINIOS.....	17
1.2.2.3 ÁRBOL DE DOMINIOS	17
1.2.2.4 BOSQUES DE DOMINIO.....	18
1.2.2.5 SITIOS	18

1.2.2.6	UNIDADES ORGANIZATIVAS OU.....	19
1.3	OFFICE 365.....	20
1.3.1	SERVICIOS DE OFFICE 365.....	21
1.3.1.1	EXCHANGE ONLINE.....	21
1.3.1.2	SHAREPOINT ONLINE.....	21
1.3.1.3	SKYPE EMPRESARIAL ONLINE.....	23
1.3.1.4	OFFICE ONLINE.....	23
1.3.1.5	DIRSYNC.....	23
1.4	ESTADO DEL ARTE DE LAS TÉCNICAS DE AUTENTICACIÓN Y CONTROL DE IDENTIDAD DIGITAL.....	23
1.4.1	ANÁLISIS DE LOS MECANISMOS DE AUTENTICACIÓN.....	27
1.4.1.1	SISTEMAS BASADOS EN CONOCIMIENTO.....	27
1.4.1.2	SISTEMAS BASADOS EN ALGO POSEÍDO POR USUARIOS.....	29
1.4.1.3	SISTEMAS BASADOS EN LO QUE ES EL USUARIO.....	30
1.4.2	ANÁLISIS DE LOS MECANISMOS DE AUTENTICACIÓN PARA EL RESETEO DE CONTRASEÑA.....	33
1.4.2.1	PROCEDIMIENTOS PARA CAMBIAR LA CONTRASEÑA.....	36
	CAPÍTULO 2.....	37
2	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PROTOTIPO.....	37
2.1	DISEÑO.....	38
2.1.1	EXPLORACIÓN.....	38
2.1.1.1	HISTORIAS DE USUARIO.....	38
2.1.2	PLANIFICACIÓN.....	41
2.1.2.1	PRIORIDADES DE HISTORIA.....	41
2.1.3	ITERACIÓN.....	42
2.1.3.1	ANÁLISIS DE ITERACIONES.....	42
2.1.3.2	ENTREGABLES POR ITERACIÓN Y TIEMPOS ESTIMADOS.....	42
2.1.4	ITERACIÓN 1.....	42
2.1.4.1	IDENTIFICACIÓN DE OBJETOS.....	42
2.1.4.2	DIAGRAMA DE CLASES.....	45
2.1.4.3	DISEÑO DE BASE DE DATOS.....	45
2.1.4.4	CASOS DE USO.....	46
2.1.4.5	MÓDULO INGRESO DE DATOS DE VERIFICACIÓN.....	46
2.1.4.6	MÓDULOS DE ADMINISTRACIÓN.....	50
2.1.4.7	MÓDULO DE CAMBIO DE CONTRASEÑA.....	52
2.2	PREPARACIÓN DE UN AMBIENTE CONTROLADO.....	56
2.2.1	MÁQUINAS VIRTUALES.....	56
2.2.2	CARACTERÍSTICAS DE LOS EQUIPOS.....	57
2.2.3	CONFIGURACIÓN DE LA VNET.....	58
2.2.3.1	SUBREDES.....	58
2.2.4	CONFIGURACIONES DE LAS VNIC DE LAS MV.....	59
2.2.5	IMPLEMENTACIÓN DE UN CONTROLADOR DE DOMINIO.....	60
2.2.6	CONFIGURACIÓN DEL DNS EN LA VNET DE AZURE.....	68
2.2.7	CONFIGURACIÓN DE UNA ENTIDAD CERTIFICADORA DE WINDOWS.....	69
2.2.8	CONFIGURACIÓN DE UN SERVIDOR DE APLICACIONES WEB.....	79

2.3	IMPLEMENTACIÓN	80
2.3.1	DESARROLLO DE INTERFACES WEB	80
2.3.2	ITERACIÓN 2.....	82
2.3.2.1	CONECTOR AL DIRECTORIO ACTIVO	84
2.3.2.2	BÚSQUEDA EN EL DIRECTORIO ACTIVO	85
2.3.2.3	BÚSQUEDA DE OBJETO ACTIVE DIRECTORY (AD)	86
2.3.2.4	CAMBIO DE CONTRASEÑA	87
2.3.3	ITERACIÓN 3 (ANILLOS DE SEGURIDAD)	89
2.3.3.1	CÓDIGO ALEATORIO	89
2.3.3.2	ENVIO SMS	89
2.3.3.3	ENVIO DE CORREO ELECTRÓNICO	92
2.3.3.4	VALIDACIÓN DE PREGUNTAS	95
2.3.4	ITERACIÓN 4 FORMULARIO WEB (VALIDACIÓN DE DATOS)	99
2.3.4.1	BIBLIOTECA DE CLASES	100
2.3.4.2	ASIGNACIÓN DEL CERTIFICADO DIGITAL	101
2.3.4.3	AUTENTIFICACIÓN DE WINDOWS	102
2.3.4.4	OBTENCIÓN DE DATOS USUARIO.....	103
2.3.4.5	CONSULTAR PREGUNTAS EXISTENTES	107
2.3.4.6	PREGUNTAS ADICIONALES PARA EL USUARIO	107
2.3.4.7	CARGAR DATOS DE USUARIO.....	108
2.3.4.8	PRESENTACIÓN DE LOS DATOS.....	110
2.3.4.9	ACTUALIZACIÓN DE DATOS	112
2.3.5	ITERACIÓN 5 FORMULARIO WEB (ADMINISTRACIÓN DE USUARIOS)	116
2.3.5.1	PERMISO SOLO A GRUPO DE SEGURIDAD DE AD.....	116
2.3.5.2	ACTUALIZAR USUARIOS DEL AD EN LA BASE DE DATOS	117
2.3.5.3	EDITAR INFORMACIÓN DE USUARIOS	120
2.3.5.4	AÑADIR PREGUNTAS	121
2.3.6	ITERACIÓN 6 RELEASE FORMULARIO WEB (CAMBIO DE CONTRASEÑA)	122
2.3.7	VERIFICACIÓN DE EXISTENCIA DEL USUARIO EN EL AD.....	126
2.3.8	PRIMER ANILLO DE SEGURIDAD.....	128
2.3.9	SEGUNDO ANILLO DE SEGURIDAD	130
2.3.10	FORMULARIO SCP	131
	 CAPÍTULO 3	 133
3	PRUEBAS DE FUNCIONAMIENTO	133
3.1	PRUEBAS DE FUNCIONAMIENTO MÓDULO ADMINISTRACIÓN... ..	133
3.1.1	ACTUALIZACIÓN DE USUARIOS DE AD	133
3.1.2	EDICIÓN DE USUARIO	135
3.1.3	AÑADIR PREGUNTAS	136
3.2	PRUEBA DE FUNCIONAMIENTO MÓDULO USUARIOS	137
3.2.1	CAMBIO DE CORREO Y CELULAR.....	138
3.2.2	AÑADIR PREGUNTAS DE VALIDACIÓN	138
3.2.3	EDITAR RESPUESTA	139
3.3	PRUEBA DE FUNCIONAMIENTO MÓDULO CAMBIO DE CONTRASEÑA	139

3.3.1	BÚSQUEDA DEL USUARIO EN EL AD	139
3.3.2	INGRESO INCORRECTO DEL USUARIO	140
3.3.2.1	ERROR AL TERCER INTENTO	140
3.3.3	PRUEBAS PRIMER ANILLO DE SEGURIDAD	141
3.3.3.1	ENVIO DE CÓDIGO A CELULAR	141
3.3.3.2	ENVIO DE CÓDIGO POR CORREO	142
3.3.3.3	COMPROBACIÓN DEL CÓDIGO	143
3.3.4	PRUEBAS SEGUNDO ANILLO DE SEGURIDAD	144
3.3.4.1	PREGUNTAS DE VALIDACIÓN	144
3.3.5	CAMBIAR CONTRASEÑA	145
	 CAPÍTULO 4	 146
4	CONCLUSIONES Y RECOMENDACIONES	146
4.1	CONCLUSIONES	146
4.2	RECOMENDACIONES	147
	 REFERENCIAS	 149
	 ANEXOS	 ERROR! BOOKMARK NOT DEFINED.

ÍNDICE DE FIGURAS

FIGURA 1.1	DECLARACIÓN DE VARIABLES ARCHIVO DE CONFIGURACIÓN.....	5
FIGURA 1.2	VARIABLES DE CONFIGURACIÓN	5
FIGURA 1.3	PRIMERA SOLICITUD PARA LA APLICACIÓN	8
FIGURA 1.4	GENERACIÓN DEL OBJETO HTTPAPPLICATION	8
FIGURA 1.5	ACTIVE DIRECTORY DOMAIN SERVICE (ADDS)	12
FIGURA 1.6	BASE DE DATOS DEL AD	14
FIGURA 1.7	CARPETA SYSVOL	14
FIGURA 1.8	REPRESENTACIÓN DE UN CONTROLADOR DE DOMINIO CON EL ROLL DE CATÁLOGO GLOBAL.....	15
FIGURA 1.9	REPRESENTACIÓN DE UN CONTROLADOR DE DOMINIO DE SOLO LECTURA	16
FIGURA 1.10	REPRESENTACIÓN DE UNA BOSQUE, ÁRBOL, DOMINIO Y SUBDOMINIOS.	18
FIGURA 1.11	ESTRUCTURA DE OUS	19
FIGURA 1.12	CLASIFICACIÓN DE LA AUTENTICACIÓN	25
FIGURA 1.13	IDENTIDAD.....	26
FIGURA 1.14	IDENTIDAD DIGITAL.....	27
FIGURA 1.15	MECANISMO DE AUTENTICACIÓN	33
FIGURA 1.16	RELACIÓN COSTO Y SEGURIDAD VS. COMPLEJIDAD	34
FIGURA 1.17	RELACIÓN ACCESIBILIDAD VS. COMPLEJIDAD MECANISMO.....	34
FIGURA 2.1	CICLO DE VIDA DE XP	37
FIGURA 2.2	FORMATO DE HISTORIA DE USUARIO	38
FIGURA 2.3	DISEÑO UML DE ACUERDO CON LOS OBJETOS IDENTIFICADOS	45
FIGURA 2.4	DISEÑO DE LA BASE DE DATOS	46
FIGURA 2.5	DIAGRAMA CASOS DE USO.....	48
FIGURA 2.6	INGRESO MÓDULO USUARIO.....	49
FIGURA 2.7	OPCIONES DE MÓDULO USUARIOS	49
FIGURA 2.8	ACTUALIZACIÓN DE DATOS	50
FIGURA 2.9	INGRESO MÓDULO DE ADMINISTRACIÓN	51
FIGURA 2.10	PROCESOS DE MÓDULO ADMINISTRACIÓN.....	52
FIGURA 2.11	INFORMACIÓN PARA VALIDACIÓN	53
FIGURA 2.12	INGRESO AL CAMBIO DE CONTRASEÑA.....	54
FIGURA 2.13	PRIMER ANILLO DE SEGURIDAD	55
FIGURA 2.14	SEGUNDO ANILLO DE SEGURIDAD	56
FIGURA 2.15	SUSCRIPCIÓN DE MSDN.....	56
FIGURA 2.16	MÁQUINAS PROVISTAS.....	57
FIGURA 2.17	CARACTERÍSTICAS DE LAS MV.....	57
FIGURA 2.18	ESPACIO DE DIRECCIONES DE VNET	58
FIGURA 2.19	SUBREDES DE AZURE	59
FIGURA 2.20	ASIGNACIÓN DE IP ESTÁTICA.....	59
FIGURA 2.21	SERVER MANAGER	60
FIGURA 2.22	AÑADIR ROLES Y CARACTERÍSTICAS.....	61

FIGURA 2.23	SERVER ROLES	61
FIGURA 2.24	CARACTERÍSTICAS NECESARIAS PARA EL SERVICIO DE DOMINIO.	62
FIGURA 2.25	INFORMACIÓN DE SERVICIO A IMPLEMENTAR.....	62
FIGURA 2.26	CONFIRMACIÓN DE SERVICIO A INSTALAR.....	63
FIGURA 2.27	PROMOVER AL SERVIDOR COMO UN CONTROLADOR DE DOMINIO	63
FIGURA 2.28	AÑADIR NUEVO BOSQUE DE AD	64
FIGURA 2.29	OPCIONES DEL CONTROLADOR DE DOMINIO	64
FIGURA 2.30	NOMBRE NETBIOS DEL DOMINIO.	65
FIGURA 2.31	ESPECIFICACIÓN DE UBICACIÓN BASE DE DATOS Y CARPETA COMPARTIDA	65
FIGURA 2.32	RESUMEN DE CONFIGURACIÓN CONTROLADOR DE DOMINO	66
FIGURA 2.33	CHEQUEO DE PRERREQUISITOS	67
FIGURA 2.34	CONSOLA DE ACTIVE DIRECTORY USERS AND COMPUTERS	67
FIGURA 2.35	CONSOLA DE ACTIVE DIRECTORY SITES AND SERVICE.....	68
FIGURA 2.36	CONFIGURACIÓN DE DNS	68
FIGURA 2.37	ACTIVACIÓN DEL SERVICIO DE ENTIDAD CERTIFICADORA.....	69
FIGURA 2.38	CARACTERÍSTICAS DE ENTIDAD CERTIFICADORA	69
FIGURA 2.39	ENROLAMIENTO WEB DE CERTIFICADOS	70
FIGURA 2.40	RESUMEN DE CONFIGURACIÓN ENTIDAD CERTIFICADORA RAÍZ.....	70
FIGURA 2.41	CONFIGURACIÓN DEL SERVICIO DE ENTIDAD CERTIFICADORA.....	71
FIGURA 2.42	ASISTENTE DE CONFIGURACIÓN ROLES DEL SERVIDOR.....	71
FIGURA 2.43	CONFIGURACIÓN DE ENTIDAD CERTIFICADORA	72
FIGURA 2.44	ENTIDAD CERTIFICADORA ENTERPRISE.....	72
FIGURA 2.45	ENTIDAD CERTIFICADORA RAÍZ	73
FIGURA 2.46	CREACIÓN DE LLAVE PRIVADA.....	73
FIGURA 2.47	CODIFICACIÓN DE LLAVE PRIVADA.....	74
FIGURA 2.48	NOMBRE DE LA ENTIDAD CERTIFICADORA.....	74
FIGURA 2.49	VIGENCIA DE CLAVE PRIVADA.....	75
FIGURA 2.50	UBICACIÓN DE LA DB DE CERTIFICADOS Y SUS LOGS.	75
FIGURA 2.51	RESUMEN DE CONFIGURACIÓN ENTIDAD CERTIFICADORA.....	76
FIGURA 2.52	CONFIGURACIÓN DE ENROLAMIENTO WEB	76
FIGURA 2.53	ACTIVACIÓN DE ENROLAMIENTO WEB.....	77
FIGURA 2.54	CARPETA DE IIS CERTSRV	77
FIGURA 2.55	PORTAL DE ENROLAMIENTO WEB.	78
FIGURA 2.56	VER EL CERTIFICADO RAÍZ EN EL ALMACENAMIENTO DE CERTIFICADOS	78
FIGURA 2.57	CERTIFICADO RAÍZ.....	78
FIGURA 2.58	HABILITACIÓN DE IIS.....	79
FIGURA 2.59	CONSOLA DE ADMINISTRACIÓN DE IIS.....	79
FIGURA 2.60	DIVISIÓN DE FORMULARIO WEB.....	80

FIGURA 2.61	CREACIÓN DE HOJA DE ESTILO	81
FIGURA 2.62	HOJA DE ESTILO CSS	81
FIGURA 2.63	REFERENCIA DEL ESTILO DE LA PAGINA.....	82
FIGURA 2.64	VISTA WEB DEL FORMULARIO.....	82
FIGURA 2.65	FORMULARIO WEB EN ASP.NET.....	83
FIGURA 2.66	FORMULARIO DEL USUARIO	84
FIGURA 2.67	BÚSQUEDA DEL OBJETO EN EL DIRECTORIO ACTIVO	86
FIGURA 2.68	CAMBIO DE CONTRASEÑA	87
FIGURA 2.69	NUEVA CONTRASEÑA.....	87
FIGURA 2.70	CUENTA Y TOKEN DE TWILIO.....	90
FIGURA 2.71	INSTALACIÓN DE LIBRERÍA "TWILIO" EN VISUAL STUDIO	90
FIGURA 2.72	FORMULARIO ASP.NET ENVIO DE SMS	91
FIGURA 2.73	CÓDIGO DE VALIDACIÓN	92
FIGURA 2.74	USUARIO PARA RELAY	93
FIGURA 2.75	FORMULARIO WEB ENVIO DE CORREO.....	94
FIGURA 2.76	ENVIO DE CORREO	94
FIGURA 2.77	CLAVE DE COMPROBACIÓN.....	95
FIGURA 2.78	TABLA PREGUNTAS	95
FIGURA 2.79	DROPBOXLIST	97
FIGURA 2.80	VALIDACIÓN DE RESPUESTAS.....	98
FIGURA 2.81	EJEMPLO DE VALIDACIÓN DE RESPUESTAS	99
FIGURA 2.82	OBJETOS QUE REPRESENTAN EL PROGRAMA.....	100
FIGURA 2.83	BIBLIOTECA DE CLASES	101
FIGURA 2.84	LIBRERÍA TECABAD.....	101
FIGURA 2.85	ASIGNACIÓN DE CERTIFICADO.....	101
FIGURA 2.86	AUTENTIFICACIÓN DE WINDOWS.....	102
FIGURA 2.87	CREDENCIALES DE WINDOWS	102
FIGURA 2.88	CARGAR DATOS DE USUARIO	109
FIGURA 2.89	PRESENTACIÓN DE DATOS.....	110
FIGURA 2.90	INFORMACIÓN DE USUARIO	111
FIGURA 2.91	EDITAR PREGUNTAS.....	111
FIGURA 2.92	AÑADIR PREGUNTA.....	112
FIGURA 2.93	PERMISOS SOLO A UN GRUPO DE SEGURIDAD.....	116
FIGURA 2.94	GRUPO PERMITIDO PARA ADMINISTRAR SITIO WEB	117
FIGURA 2.95	ACTUALIZAR USUARIOS DE AD EN DB.....	119
FIGURA 2.96	SQLDATASOURCE.....	120
FIGURA 2.97	EDITAR CAMPO DE USUARIO.....	120
FIGURA 2.98	PREGUNTAS EXISTENTES.....	121
FIGURA 2.99	FLUJO GRAMA CAMBIO DE CONTRASEÑA	123
FIGURA 2.100	FUNCIÓN SERVER TRANSFER	124
FIGURA 2.101	DISEÑO WEB CAMBIO DE CONTRASEÑA.....	125
FIGURA 2.102	PANTALLA CAMBIO DE CONTRASEÑA	125
FIGURA 2.103	ENVIO DE CÓDIGO DE VALIDACIÓN	127
FIGURA 2.104	SELECCIÓN PARA ENVIO DE SMS O CORREO	129
FIGURA 2.105	CÓDIGO DE VERIFICACIÓN	129
FIGURA 2.106	ESCOGER PREGUNTA DE VALIDACIÓN	130
FIGURA 2.107	SELECCIONAR PREGUNTA.....	131
FIGURA 2.108	CONTESTAR LA PREGUNTA.....	131

FIGURA 2.109	CAMBIO DE CONTRASEÑA	132
FIGURA 3.1	CREACIÓN DE USUARIO EN EL AD	133
FIGURA 3.2	CREACIÓN DE USUARIO PARA PRUEBAS DE FUNCIONAMIENTO	134
FIGURA 3.3	ACTUALIZAR USUARIOS DE AD	134
FIGURA 3.4	AUTORIZACIÓN DE CAMBIO DE CONTRASEÑA.....	135
FIGURA 3.5	EDITAR INFORMACIÓN DEL USUARIO.....	135
FIGURA 3.6	ACTUALIZAR DATOS	136
FIGURA 3.7	AÑADIR PREGUNTA AL BANCO DE PREGUNTAS	136
FIGURA 3.8	PREGUNTA AÑADIDA	137
FIGURA 3.9	INFORMACIÓN DEL USUARIO CORREO Y CELULAR.	137
FIGURA 3.10	CAMBIO DE CORREO Y CELULAR.....	138
FIGURA 3.11	AÑADIR PREGUNTA MÓDULO USUARIO	138
FIGURA 3.12	EDITAR RESPUESTA	139
FIGURA 3.13	BUSCA EL USUARIO EN EL AD	140
FIGURA 3.14	USUARIO INCORRECTO	140
FIGURA 3.15	BLOQUEO POR INTENTOS FALLIDOS.....	141
FIGURA 3.16	ENVIO DE CÓDIGO POR MENSAJE DE TEXTO	141
FIGURA 3.17	ENVIO DE SMS	142
FIGURA 3.18	ENVIO DE CÓDIGO POR CORREO ELECTRÓNICO	142
FIGURA 3.19	VERIFICACIÓN EN EL CORREO.....	143
FIGURA 3.20	COMPROBACIÓN DEL CÓDIGO	143
FIGURA 3.21	PANTALLA PARA ESCOGER PREGUNTA DE VALIDACIÓN	144
FIGURA 3.22	CONTESTAR PREGUNTA DE VALIDACIÓN	144
FIGURA 3.23	CAMBIAR CONTRASEÑA	145
FIGURA 3.24	MENSAJE DE CONFIRMACIÓN	145

LISTA DE TABLAS

TABLA 1.1	CLASIFICACIÓN DE LOS COMPONENTES DE UN SERVICIO DE DIRECTOR ACTIVO	13
TABLA 1.2	PLANES Y FAMILIAS DE SERVICIOS DE OFFICE 365	20
TABLA 1.3	SERVICIOS DE OFFICE 365	21
TABLA 1.4	DISPONIBILIDAD DE LOS SERVICIOS EN CADA PLAN	22
TABLA 1.5	CUADRO COMPARATIVO DE TÉCNICAS BIOMÉTRICAS	31
TABLA 1.6	CUADRO COMPARATIVO DE TIPOS DE MECANISMOS DE AUTENTICACIÓN	32
TABLA 2.1	HISTORIA DE USUARIO # 1	39
TABLA 2.2	HISTORIA DE USUARIO # 2	39
TABLA 2.3	HISTORIA DE USUARIO # 3	40
TABLA 2.4	HISTORIA DE USUARIO # 4	40
TABLA 2.5	HISTORIA DE USUARIO # 5	41
TABLA 2.6	PRIORIDAD HISTORIA DE USUARIO.....	41
TABLA 2.7	DESARROLLO DE ITERACIONES.....	42
TABLA 2.8	ENTREGABLES.....	43
TABLA 2.9	POSIBLES OBJETOS DETECTADOS EN LAS HISTORIAS DE USUARIOS	44
TABLA 2.10	OBJETOS IDENTIFICADOS DE LAS HISTORIAS DE USUARIO.....	44
TABLA 2.11	POSIBLES OBJETOS DETECTADOS EN LOS MECANISMOS DE AUTENTICACIÓN.....	44
TABLA 2.12	OBJETOS DETECTADOS	44

LISTA DE CÓDIGOS

CÓDIGO 2.1	CONEXIÓN CON EL DIRECTORIO ACTIVO	85
CÓDIGO 2.2	BÚSQUEDA DE USUARIO EN EL AD	85
CÓDIGO 2.3	LLAMADO DE LA FUNCIÓN GETUSER.	86
CÓDIGO 2.4	VALIDACIÓN CAMBIO DE CONTRASEÑA.....	88
CÓDIGO 2.5	CAMBIO DE CONTRASEÑA	88
CÓDIGO 2.6	CÓDIGO ALEATORIO	89
CÓDIGO 2.7	CÓDIGO ENVIO DE MENSAJE TEXTO A CELULAR	91
CÓDIGO 2.8	LLAMADA A FUNCIÓN SMS (ENVIO DE SMS)	92
CÓDIGO 2.9	LIBRERÍAS PARA PODER ENVIAR CORREO ELECTRÓNICO	93
CÓDIGO 2.10	FUNCIÓN PARA ENVIO DE CORREO (PARTE 1).....	93
CÓDIGO 2.11	FUNCIÓN PARA ENVIO DE CORREO (PARTE 2).....	94
CÓDIGO 2.12	LIBRERÍA PARA INTERACTUAR CON LA BASE DE DATOS DE SQL.....	96
CÓDIGO 2.13	CARGAR BASE DE DATOS DE SQL EN DOWNLIST	96
CÓDIGO 2.14	VALIDAR PREGUNTA.....	98
CÓDIGO 2.15	LIBRERÍAS ITERACIÓN 4	103
CÓDIGO 2.16	OBTENER EL ALIAS DEL USUARIO	103
CÓDIGO 2.17	OBTENCIÓN DE LA UBICACIÓN EN EL DIRECTORIO ACTIVO	104
CÓDIGO 2.18	CANONICAL NAME DEL USUARIO	105
CÓDIGO 2.19	CREACIÓN DE USUARIO	105
CÓDIGO 2.20	CARGA DATOS DE CORREO Y CELULAR DE LA BASE DE DATOS.....	105
CÓDIGO 2.21	CARGAR PREGUNTAS DE LA BASE DE DATOS.....	106
CÓDIGO 2.22	BIBLIOTECA DE PREGUNTAS EXISTENTES	107
CÓDIGO 2.23	PREGUNTAS DISPONIBLES	108
CÓDIGO 2.24	CARGAR DATOS DE USUARIO EN FORMULARIO (PARTE 1)	108
CÓDIGO 2.25	CARGAR DATOS DE USUARIO EN FORMULARIO (PARTE 2)	109
CÓDIGO 2.26	BOTÓN CARGAR USUARIO.....	110
CÓDIGO 2.27	ACTUALIZACIÓN DE DATOS DEL USUARIO	114
CÓDIGO 2.28	ACTUALIZAR LA BASE DE DATOS (PARTE 1).....	114
CÓDIGO 2.29	ACTUALIZAR LA BASE DE DATOS (PARTE 2).....	115
CÓDIGO 2.30	AÑADIR PREGUNTA DE UN USUARIO EN LA BASE DE DATOS	115
CÓDIGO 2.31	VALIDAR USUARIO EN LA BASE DE DATOS.....	117
CÓDIGO 2.32	INSERTAR NUEVO USUARIO EN LA BASE DE DATOS	118
CÓDIGO 2.33	CARGAR USUARIOS	118
CÓDIGO 2.34	OBJETO DE BIBLIOTECA DE PREGUNTAS	121
CÓDIGO 2.35	AÑADIR PREGUNTAS A LA TABLA DE PREGUNTAS DE LA BASE DE DATOS.....	122
CÓDIGO 2.36	VERIFICACIÓN DE USUARIO EN EL AD	126
CÓDIGO 2.37	BOTÓN RECUPERAR CONTRASEÑA	126
CÓDIGO 2.38	INTENTOS FALLIDOS.....	127

CÓDIGO 2.39	CARGAR USUARIO	128
CÓDIGO 2.40	VERIFICA EL CÓDIGO ENVIADO	130
CÓDIGO 2.41	CAMBIO DE CONTRASEÑA	132

LISTA DE ANEXOS

ANEXO A BIBLIOTECA DE CLASES A-**ERROR! BOOKMARK NOT DEFINED.**

ANEXO B CÓDIGO CUARTA ITERACIÓNB-**ERROR! BOOKMARK NOT DEFINED.**

ANEXO C QUINTA ITERACIÓNC-**ERROR! BOOKMARK NOT DEFINED.**

ANEXO D CAMBIO DE CONTRASEÑA D-**ERROR! BOOKMARK NOT DEFINED.**

RESUMEN

El presente trabajo, tiene como objetivo diseñar e implementar un sistema prototipo para el cambio de contraseña de una cuenta de Directorio Activo desde fuera de la organización.

En el capítulo 1, estudia las tecnologías involucradas en el desarrollo del prototipo como: .net *Framework*, Active Directory, Office 365, Dirsync, así como el estado del arte de la autenticación para la selección de los mecanismos a utilizar.

En el capítulo 2, presenta el diseño e implementación del sistema prototipo utilizando una adaptación a la metodología de desarrollo XP(*eXtreme Programming*), dentro de este desarrollo se involucra una fase de exploración donde se hace uso de historias de usuarios, para recolectar información que es usada para el diseño de objetos, base de Datos, módulos de sistemas, también se divide el trabajo en iteraciones donde cada iteración abarca un segmento de código que se presenta a manera de pruebas de concepto, estas pruebas de concepto se integran para formar el prototipo.

En el capítulo 3, pruebas de funcionamiento realizadas a los módulos de usuarios, módulos de administración y módulo de cambio de contraseña.

En el capítulo 4, en base a la experiencia adquirida durante el desarrollo del proyecto se presentan las conclusiones y recomendaciones, mismas que pueden ser guías para el desarrollo de futuros proyectos de titulación.

PRESENTACIÓN

En la actualidad las empresas usan al directorio activo de Microsoft como principal mecanismo de autenticación para el ingreso de sus equipos, aplicaciones o servicios en general. Las políticas de seguridad de los directorios activos cambian cada cierto tiempo la contraseña del usuario, así como el bloqueo de las cuentas si la contraseña es ingresada incorrectamente. Si el usuario bloquea su cuenta no tiene acceso en los sistemas, esta acción produce un gran número de tiquetes de soporte para los usuarios de TI.

Con la llegada del internet cada día más servicios son publicados hacia fuera de la corporación o migradas a plataformas externas, para permitir el acceso a los usuarios, cuando se encuentran fuera de la organización, las aplicaciones usan al directorio activo como mecanismos de autenticación, esto limita que el usuario solo pueda cambiar la contraseña cuando se encuentra dentro de la organización.

Para solventar este inconveniente, se ha visto la necesidad de desarrollar un sistema que permita cambiar la contraseña de manera segura desde fuera de la organización.

El sistema propuesto soluciona los siguientes inconvenientes:

- Cambiar la contraseña del usuario desde cualquier parte del mundo.
- Mantener la privacidad de la clave del usuario, al no tener que solicitar el cambio o desbloqueo de esta al administrador de sistemas.
- Disminuir los tiques de soporte generado por bloqueos u olvidos de contraseña.
- El usuario tiene la disponibilidad de cambiar la contraseña a cualquier hora.
- Permitir a los sistemas publicados que utilizan la autenticación del directorio activo una movilidad completa.

El sistema presenta las siguientes características:

- Un módulo de usuario que permite el ingreso de información, para validar la identidad de la persona previo al cambio de contraseña.

- Un módulo de cambio de contraseña, mediante el envío de un código a un celular o correo y a través de preguntas frecuentes permite identificar al usuario que desea cambiar la contraseña.
- Un módulo de administración, que permite autorizar a los usuarios que pueden cambiar la contraseña, editar información de número de celulares y correos de los usuarios, además de la capacidad de incrementar preguntas de validación al sistema.

Capítulo 1

1 FUNDAMENTO TEÓRICO

En este capítulo se describirá el fundamento teórico que sirve de base para la realización del prototipo a implementar. Entre ellas, se incluirá una descripción de las tecnologías ASP.NET, *Framework* .NET, Directorio Activo de Microsoft, DirSync y OFFICE 365.

Adicionalmente, en este capítulo se mostrarán los resultados del estudio del estado del arte de las técnicas de autenticación y control de identidad digital.

1.1 TECNOLOGÍAS UTILIZADAS

1.1.1 ASP.NET

Es una tecnología de desarrollo Web de Microsoft [1], está direccionada como un modelo de desarrollo para aplicativa web unificando los servicios necesarios para crear las mismas, empleando el menor código posible. ASP.NET es un segmento de la tecnología conocida como *.NET Framework*, la cual permite que los códigos de aplicaciones desarrolladas en ASP.NET accedan directamente a las clases pre-desarrolladas de *.NET Framework*. Además ASP.NET maneja una capa llamada CLR (*Common Language Runtime*) que permite compilar los códigos escritos en varios lenguajes de programación:

- C#
- Visual Basic
- J#
- JScript .NET

ASP.NET hace uso de componentes que facilitan el desarrollo de las aplicaciones web, a continuación, se procede a describir cada una de ellas.

1.1.1.1 Marco de trabajo de página y controles

Para hacer uso de las páginas y controles de ASP.NET, se requiere utilizar una infraestructura donde se ejecuten las mismas. Microsoft utiliza el servicio de Internet Information Service (IIS) como plataforma para ejecutar las páginas y controles web de ASP.NET.

Las aplicaciones web creadas en ASP.NET son solicitadas por el cliente, desde cualquier explorador (Internet Explorer, Firefox, Chrome, Safari, etc.) ya que utiliza como base HTML. Si se desea personalizar el desarrollo para un navegador específico, ASP.NET permite hacerlo y aprovechar todas las características que pueda brindar ese explorador. También se puede utilizar controles específicos para dispositivos inteligentes:

- Teléfonos Inteligentes.
- Ordenadores.
- Portátiles.
- Tablet.

Las páginas Web ASP.NET trabajan haciendo uso de objetos, así como elementos propios de HTML. ASP.NET oculta los datos que se colocaron en la implementación; ejemplo los contenidos que se programan dentro de una clase no son revelados en los exploradores, creando una separación entre lo que se almacena en el navegador del cliente y lo que contiene el sitio web en el servidor.

La infraestructura digital también mantiene el estado de la página y controles durante un tiempo determinado (Sesión).

ASP.NET encapsula la funcionalidad de la interfaz de usuario con controles fáciles de manejar y pueden ser reutilizados en otros módulos, también presenta funciones que permiten controlar la apariencia de los sitios Web, utilizando temas y máscaras, las cuales se aplican a las páginas o controles de manera rápida y efectiva.

Así mismo, es posible definir páginas maestras que contengan las características de diseño que serán heredadas a las demás páginas, de esa manera conseguir un diseño coherente. También se pueden crear páginas de contenido, es decir estas pasan a ser parte de un segmento de otra, así cuando los usuarios solicitan las páginas de contenido, los exploradores solo cambian un segmento de la página principal con el fin de generar un resultado que combine el diseño de una con el contenido de la otra.

1.1.1.2 Compilador de ASP.NET

Todo código de ASP.NET es compilado para generar un tipado robusto, una optimización de los recursos, accesos directos, entre otros beneficios, es decir, la compilación usando la capa (CLR) más las bondades de utilizar el código ASP.NET como código nativo, permiten un alto rendimiento.

ASP.NET compilará todos los componentes de la aplicación, incluidas páginas y controles web en un código ensamblador, estos se alojan en un servidor web.

1.1.1.3 Infraestructura de seguridad

ASP.NET permite asignar seguridad al acceso de los usuarios, codificar página mediante el uso de certificados, utilizar diferentes tipos de autenticaciones como: autenticación de Windows, anónima, Kerberos o puede administrar la autenticación con su propia base de datos de usuario y un formulario ASP.NET. Además, resulta fácil quitar, agregar o reemplazar estos esquemas dependiendo de las necesidades de la aplicación.

ASP.NET puede hacer uso de las características de Windows y asignar las mismas a las aplicaciones, por ejemplo, una ACL (*access list control*) del servicio NTFS que suele ser utilizado para dar acceso a carpetas compartidas en una red, permisos de la base de datos, etc.

1.1.1.4 Facilidades de administración de estado

ASP.NET proporciona una funcionalidad que permite administrar información entre los diferentes estados que puede estar una página, esta información suele ser solicitada al inicio de esta.

La información se puede almacenar en los diferentes tipos de variables:

- Aplicación: generadas por la aplicación
- Sesión: que se pueden usar mientras la sesión dure y puede ser usadas en otras páginas que se encuentren creadas en el mismo módulo.
- Página: exclusivas de la página, no pueden ser transferidas a otras paginas
- Usuario: Contiene información del usuario que accede al sitio web.
- Definidas por el desarrollador: creadas al momento de su desarrollo y se encuentra incluidos en el código.

Esta información puede ser independiente de cualquier control de la página y permite mitigar el problema de *flashback* que sucede al momento, que el usuario refresca la página del explorador.

ASP.NET ofrece funciones de estado distribuidas que le permite administrar información de estado en múltiples instancias de la misma aplicación, en un equipo o en varios.

1.1.1.5 Configuración de ASP.NET

Las aplicaciones web desarrolladas con ASP.NET separan su configuración mediante un archivo llamado Webconfig [2], en este archivo en la parte superior se declaran los controladores de sesión que pueden anidar en los formularios, ver Figura 1.1, y dentro de la estructura del archivo colocan los valores de configuración, ver Figura 1.2, para un sitio Web o para aplicaciones individuales, como puede ser el tipo de autenticación que utilice los formularios de la aplicación.


```

<sectionGroup name="system.web"
  type="System.Web.Configuration.SystemWebSectionGroup, System.Web, Version=2.0.0.0, Culture=neutral, Public
  <!-- <section /> elements. -->
</sectionGroup>

```

Figura 1.1 Declaración de variables archivo de configuración

```

<pages
  buffer="true"
  enableSessionState="true"
  asyncTimeout="45"
  <!-- Other attributes. -->
>
  <namespaces>
    <add namespace="System" />
    <add namespace="System.Collections" />
  </namespaces>
</pages>

```

Figura 1.2 Variables de Configuración

Puede crear valores de configuración, agregar o revisar los valores en cualquier momento, con un impacto mínimo en aplicaciones y servidores Web en el que se está ejecutando. Los valores de configuración de ASP.NET, se almacenan en archivos basados en la tecnología XML y utilizan codificación ASCII, lo que permite que sean fácilmente reemplazables.

1.1.1.6 Supervisión de estado y características de rendimiento

ASP.NET maneja estados que permiten supervisar el estado del aplicativo, así como almacenar detalles de estos, en registros, de esa manera se puede obtener un mecanismo que permite obtener un diagnóstico, monitoreo de la salud y rendimiento de los programas. A estos registros se los conoce como contadores, ASP.NET presenta dos grupos de contadores a los que pueden obtener acceso las aplicaciones:

- El grupo de contadores de rendimiento del sistema ASP.NET, estos analizan el estado general que brindan el servicio de ASP.NET.

- El grupo de contadores de rendimiento de la aplicación ASP.NET, estos presentan más detalles y se enfocan directamente en el aplicativo como puede ser un *website*, formulario o aplicación de ASP.NET.

1.1.1.7 Capacidad de depuración

Una de las ventajas de ASP.NET es que permite depuración en tiempo de ejecución, la mayoría de errores de sintaxis se detectan al momento de compilar la aplicación, para los demás errores ASP.NET utiliza el SDK (*Software Development Kit*) [3], que se incluye en la herramienta conocida como *Visual Debugger*, esta herramienta es la que permite colocar puntos de interrupción al momento de ejecutar toda la aplicativo o ejecutar línea por línea, permitiendo visualizar los valores que se almacenan durante la ejecución del programa. Esta herramienta permite hacer la depuración en la capa de CRL y en la creación de *Script* que se incluyan.

Además, el *Framework* de páginas ASP.NET proporciona un modo de seguimiento que permite implantar mensajes de instrumentalización en las páginas Web ASP.NET es decir, puede almacenar el error y luego podríamos desplegar este error en modo de un mensaje al momento que este se genere.

1.1.1.8 Framework de servicios Web XML

ASP.NET permite la utilización de servicios Web XML, estos incluyen la funcionalidad que permiten a las aplicaciones intercambiar información utilizando estándares como los servicios de mensajería HTTP y XML.

Los servicios Web XML no están atados a una tecnología de componentes ni con ninguna convención de llamada a objetos en concreto. Gracias a esto, los objetos pueden obtener acceso a los servicios Web XML, sin importar si estos fueron escritos en un determinado lenguaje, usan cualquier componente o si son ejecutados en algún sistema operativo en concreto.

1.1.1.9 Entorno de host extensible y administración del ciclo de vida de las aplicaciones

ASP.NET utiliza al servidor como una extensión que permite controlar el ciclo de vida de una aplicación [4].

El ciclo de Vida empieza cuando un usuario pide la primera solicitud al servidor, una vez que ASP.NET recibe esta solicitud genera un objeto de tipo *applicationManager* y crea un dominio de aplicación, de esta manera se crea un aislamiento de las variables globales, lo que permite que se ejecuten las aplicaciones de manera independientes dentro de un dominio de aplicación, ver Figura 1.3.

Una vez que se generan los dominios de aplicación se instancia un objeto de tipo *HostingEnvironment* y dentro del mismo se crean los siguientes objetos:

- *HttpContext*: Contiene datos sobre la solicitud actual.
- *HttpRequest*: Contiene cookies e información del explorado.
- *HttpResponse*: Contiene respuestas que se le envía al cliente.

Se tiene instanciados estos objetos, ASP.NET crea un objeto *httpApplication* y si la aplicación maneja un archivo *Global.asax* crea una clase con el mismo nombre, esta clase es derivada del objeto *httpApplication*, ver Figura 1.4.

El Objeto *httpApplication* maneja los eventos necesarios para controlar la aplicación.

1.1.1.10 Entorno de diseñador extensible

ASP.NET contiene la compatibilidad mejorada, para crear diseñadores de controles de servidor Web, para utilizarlos con una herramienta de diseño visual como Visual Studio.

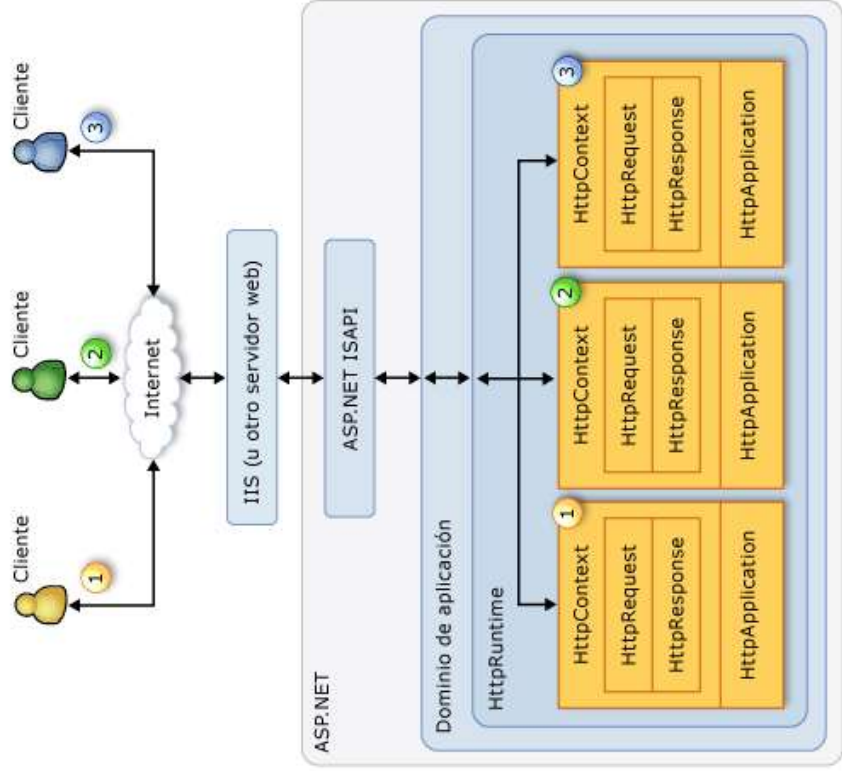


Figura 1.4 Generación del objeto httpApplication

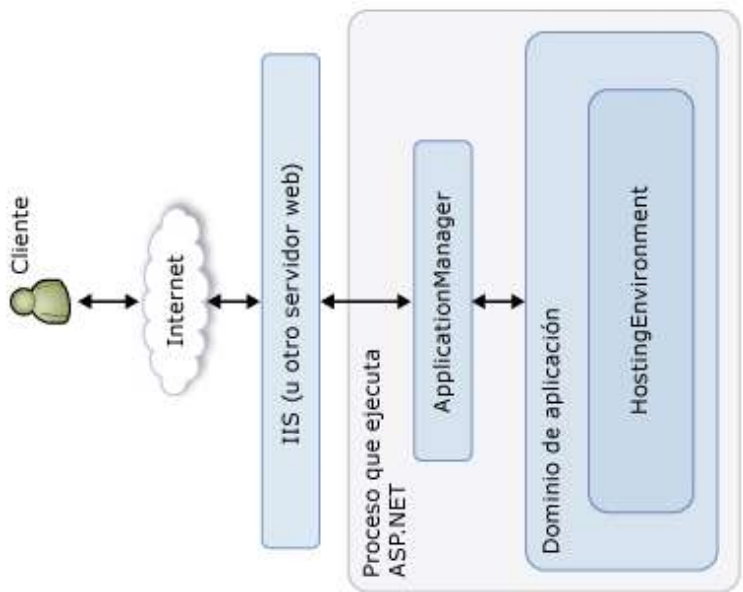


Figura 1.3 Primera solicitud para la aplicación [4]

Los diseñadores permiten crear una interfaz de usuario en tiempo de diseño, de este modo los desarrolladores pueden configurar las propiedades y el contenido de un control a través de una herramienta de diseño visual [5], Asp.net permite editar en determinada región, usar listas de acciones, enlazar orígenes de datos, usar plantillas, incorpora un modelo simplificado de HTML para añadir objetos y manejo de suscripciones.

1.1.1.11 Aplicación Web de ASP.net

Microsoft [6] indica los retos que surgen al momento de programar aplicaciones web a diferencia de la programación tradicional de aplicaciones. En el desarrollo web se presenta los siguientes retos:

- **Implementación de una interfaz de usuario Web compleja:** Puede ser difícil y tedioso el diseñar e implementar una interfaz de usuario utilizando las funciones básicas de HTML, ya que usa únicamente código, especialmente si la página tiene un diseño complejo, una gran cantidad de contenido dinámico y objetos con muchas funciones y que requieren interacción con el usuario. Pero mediante ASP.NET específicamente con los productos de Visual Studio pueden hacer uso de plantillas que cumplan este propósito las cuales pueden ser modificadas con facilidad.
- **Separación de cliente y servidor:** En una aplicación Web que corre en un servidor, este separa la información del código programado de lo que visualiza el usuario final, el cliente puede utilizar uno o varios exploradores, ejecutarlos en equipo distintos (e incluso en distintos sistemas operativos). Pero las dos mitades de la aplicación comparten muy poca información, normalmente se comunican intercambiando pequeñas porciones de información.
- **Ejecución sin estado:** Cuando un servidor Web recibe una solicitud de una página, la busca, la procesa y la envía al explorador, y a continuación, descarta toda la información de dicha página. Si el usuario solicita la página de nuevo, el servidor repite la secuencia completa, volviendo a procesar la página desde el principio. En otras palabras, los servidores no tienen memoria de las páginas que han procesado, no tienen estado. Por

consiguiente, si una aplicación necesita mantener información sobre una página, su naturaleza sin estado podría ser un problema.

- **Funciones desconocidas del cliente:** En muchos casos, las aplicaciones Web resultan accesibles a muchos usuarios que usan exploradores diferentes, los exploradores ofrecen distintas funcionalidades, lo que hace muy difícil crear una aplicación que se ejecute con la misma calidad en todos ellos, por ejemplo, en el explorador Chrome puede mostrar funciones como validar texto para un correo, pero en un explorador de internet de Microsoft, esta funcionalidad no está disponible.
- **Complicaciones con el acceso a datos:** La lectura de los datos de un origen de datos y la escritura en el mismo puede resultar complicada con las aplicaciones Web tradicionales, y requiere la utilización de varios recursos, como pueden ser un Objeto *DataSource*, o la lectura constante de los datos cada vez que se necesite la información.
- **Complicaciones con la escalabilidad:** En muchos casos, las aplicaciones Web diseñadas con los métodos existentes no cumplen los objetivos de escalabilidad, debido a la falta de compatibilidad que existen entre distintos componentes web. Este reto a menudo es más común al manejar un ciclo de crecimiento intensivo.

Vencer estos retos de las aplicaciones web puede requerir un tiempo y esfuerzo importante. Las páginas web ASP.NET y su marco de trabajo de páginas tratan de solucionar estos temas de los siguientes modos:

- **Aplicaciones independientes del explorador:** El marco de trabajo de páginas ASP.NET permite crear toda la lógica de la aplicación en el servidor, lo que elimina la necesidad de confeccionar explícitamente código para los diferentes exploradores. Sin embargo, todavía le permite aprovechar las funciones específicas de cada explorador personalizando el código para la parte cliente, con el fin de mejorar el rendimiento y de proporcionar una experiencia más enriquecedora de un explorador en particular.
- **Compatibilidad de *Common Language Runtime (CLR)* de .NET Framework:** Sus aplicaciones se pueden escribir en cualquier lenguaje que sea compatible con CRL. Además, el acceso a datos se ha simplificado

mediante la infraestructura utilizando conexiones a los datos, los cuales son provistos por *.NET Framework* y *ADO.NET*.

- **Rendimiento del servidor escalable de *.NET Framework*:** el código de páginas *ASP.NET* permite escalar las aplicaciones web de un equipo con un único procesador a varios servidores web y sin realizar cambios complicados en la lógica de la aplicación.

1.1.2 *.NET FRAMEWORK*

Microsoft [7] presenta a *.NET Framework* como la tecnología que permite compilar y ejecutar las siguientes generaciones de aplicaciones y servicios de web y su diseño está enfocado en los siguientes puntos:

- Proporcionar un entorno orientado a objetos, donde su código puede almacenar y ejecutar objetos de forma local, pero también de forma distribuida en Internet o remota.
- Al momento de ejecución de código analiza los errores que podrían ser ocasionados por la implementación de nuevos segmentos de software y conflictos de versiones.
- Dentro de un entorno de ejecución de código, *.Net Framework* fomenta la ejecución de estos de forma segura, incluso de los códigos creados por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que minimiza los problemas de rendimiento, en entornos que se utilice secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
- La comunicación que utiliza *.NET Framework* lo realiza utilizando estándares para asegurar que el código de *.NET Framework*, se puede integrar con otros tipos de código.

1.2 SERVICIO DE DIRECTORIO ACTIVO DE MICROSOFT

El servicio de Directorio Activo de Microsoft [8] es una base de datos distribuida que permite almacenar información relativa a los recursos de una red con el fin de facilitar su localización y administración.

El Directorio Activo es un servicio que puede correr en un servidor o varios al mismo tiempo. A un servidor que ejecute el servicio de Directorio Activo se lo conoce como un controlador de dominio [9], cada controlador de dominio almacena una copia de la base de datos que se sincroniza constantemente.

Este servicio de directorio activo se utiliza para administrar un conjunto de computadores en un ambiente corporativo que utilicen sistemas operativos Microsoft, el servicio de directorio activo maneja una base de datos de los objetos del dominio como usuarios, computadores y grupos como se representa en la Figura 1.5.

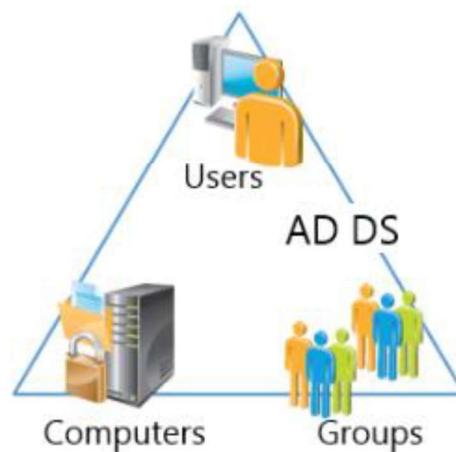


Figura 1.5 Active Directory Domain Service (AD DS) [9]

El servicio de Directorio Activo está compuesto por componentes físicos y lógicos, ambos componentes forman el ambiente de un directorio activo.

1.2.1 COMPONENTES FÍSICOS

La información del ADDS (servicio de dominio del Directorio Activo) se encuentra almacenado en una base de datos en el disco duro de cada controlador de dominio.

En la Tabla 1.1, se enlistan los componentes físicos y lógicos del Directorio Activo.

Tabla 1.1. Clasificación de los componentes de un servicio de Directorio Activo

Componentes Físicos	Componentes Lógicos
Almacenamiento de Datos	Particiones
Controladores de Dominio	Esquema
Catálogos Globales	Dominios
Controladores de Dominio de lectura únicamente RODCs	Árbol de Dominios
	Bosques
	Sitios
	Unidades Organizaciones (OUs)

1.2.1.1 Almacenamiento de Datos

Los archivos en cada controlador de dominio que almacena la información ADDS, reside en una base de datos como se observa en la Figura 1.6 llamada "ntds.dit", los archivos de log EDB (*Express Data Base*) y la carpeta compartida SYSVOL suelen tener las políticas del Dominio como se observa en la Figura 1.7.

En la base de datos NTDS, ver Figura 1.6, almacena las configuraciones y objetos que conforman al directorio activo.

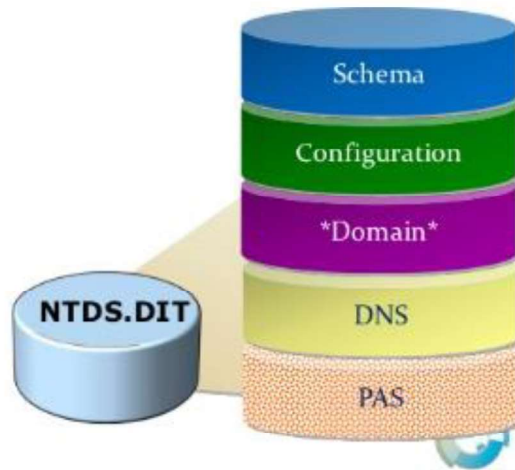


Figura 1.6 Base de datos del AD [9]

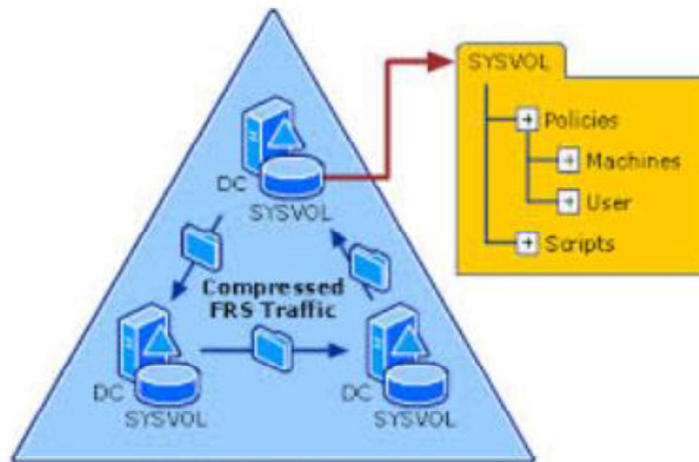


Figura 1.7 Carpeta SYSVOL [9]

Mientras que en la carpeta SYSVOL como se observa en la Figura 1.7 esta se replica en todos los controladores de dominio, dentro de esta carpeta compartida se almacenan las políticas y *Script* que se ejecutan en cada una de las máquinas del dominio.

1.2.1.2 Controladores de dominio

Un controlador de dominio contiene una réplica de la base de datos ADDS y la carpeta compartida SYSVOL, la información específica de un dominio puede ser actualizada desde cualquier controlador de dominio que sea miembro del mismo dominio.

1.2.1.3 Catálogo Global

El anfitrión con el *roll* de catálogo global ver Figura 1.8, contiene parcialmente una copia de sólo lectura de todos los objetos de un dominio de un bosque. Un catálogo global acelera las búsquedas de objetos que se almacenan en los controladores ubicados en un dominio diferente del mismo bosque.

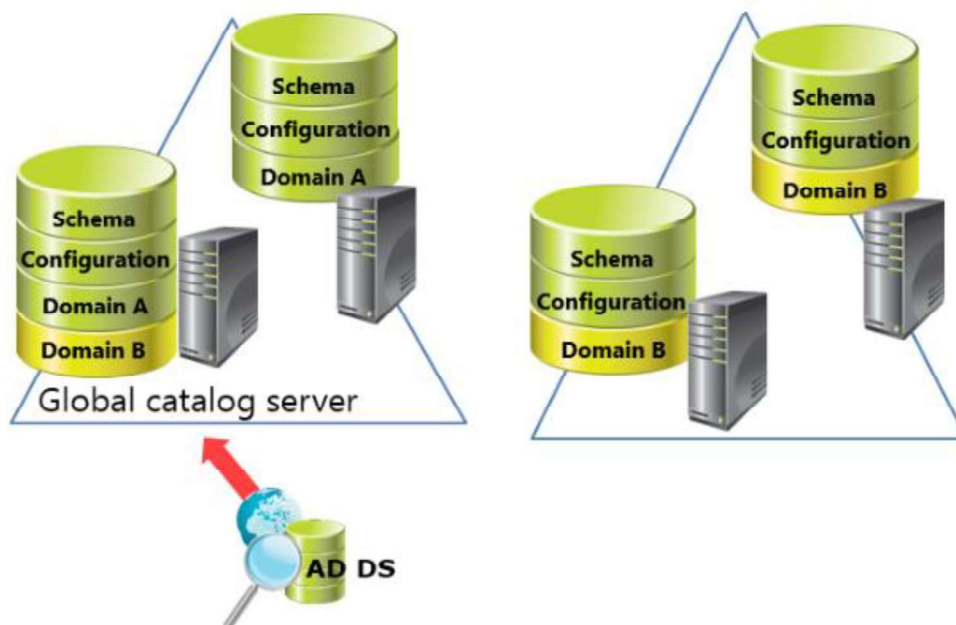


Figura 1.8 Representación de un controlador de dominio con el rol de catálogo global [9]

1.2.1.4 RODCs

Un controlador de dominio de solo lectura (RODC) es una instalación de un servidor en donde almacena una copia de solo lectura, como se observa en la Figura 1.9, los procesos de cambio de contraseña son redirigidos hacia un controlador de dominio de escritura.

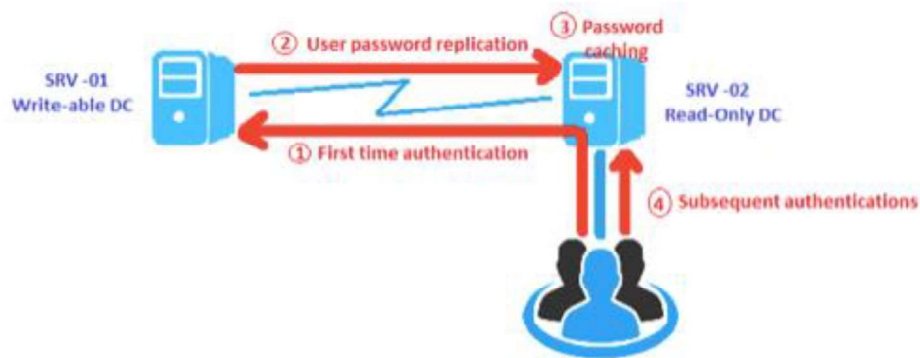


Figura 1.9 Representación de un controlador de domino de solo lectura [9]

1.2.2 COMPONENTES LÓGICOS

Los componentes lógicos del ADDS son estructuras que se utilizan para implementar un diseño de directorio activo, que es apropiado para una organización.

1.2.2.1 Particiones

Es una sección de la base de datos del AD-DS, cada controlador de dominio contiene las siguientes particiones, estas se almacenan en la base de datos como se observa en la Figura 1.6:

- **Configuración:** almacena objetos de configuración para el bosque, estos recopilan información acerca de sitios, servicios y particiones de directorio, para visualizarlos se utiliza la consola llamada ADSIEDIT.

- **Esquema:** Es el contenedor de esquema, que almacena la clase y definiciones de atributo para todos los objetos de Active Directory, esta se la puede observar con la consola del ADSIEDIT.
- **Dominio:** Es el contenedor de dominio, que almacena usuarios, equipos, grupos y otros objetos de un dominio específico, esta se puede observar con la consola de usuarios y computadores del Directorio Activo.
- **DNS de Dominio:** Los controladores de dominio utilizan el servicio de DNS para integrar el espacio de nombre del dominio, está por defecto se encuentra integrada al Active Directory, es decir los registros DNS se almacenan en estas particiones y utiliza la replicación del directorio activo para distribuirse a los demás controladores de dominio.
- **DNS de Forest:** En estas particiones se almacenan las zonas DNS que se necesitan replicarse a los demás dominios que se encuentren en el mismo bosque.
- **Aplicación:** Esta es una partición opcional que se utiliza para replicar la información de aplicaciones,

1.2.2.2 Dominios

Representa lógicamente al conjunto de equipos o usuarios que permiten compartir recursos de una manera segura y controlada, como se observa en la Figura 1.10 existen dos dominios adamtum.com y fabrikam.com, así como un subdominio alt.adamtum.com.

1.2.2.3 Árbol de Dominios

Un árbol de dominios conoce al esquema como se observa en la Figura 1.10 con el dominio Frabrican.com cuando este está constituido por un solo dominio, en este caso el nombre del árbol es igual al de dominio.

1.2.2.4 Bosques de dominio

Una colección de uno o más dominios que comparten el mismo nombre raíz en común, por ejemplo, como se observa en la Figura 1.10, existe un dominio raíz llamado Adantum.com y este puede ser parte de muchos dominios de un mismo bosque de dominio, como es el sub-dominios atl.adantum.com, que comparten el mismo dominio raíz de adamantum.com.

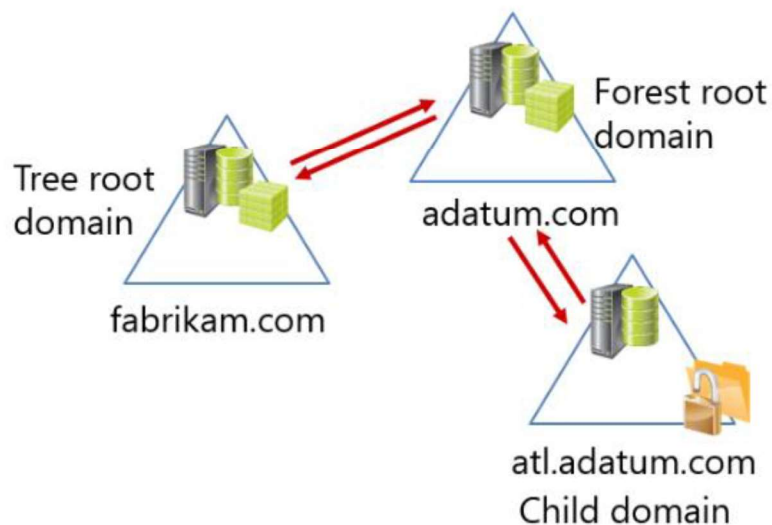


Figura 1.10 Representación de una bosque, árbol, dominio y subdominios.

1.2.2.5 Sitios

El Directorio Activo utiliza a los objetos llamados sitios para representar un lugar físico como podría ser una ciudad, un distrito o un edificio; dentro de este objeto se almacena una o varias subredes que contenga ese sitio en específico, cada sitio normalmente suele estar conectado mediante un enlace WAN y siempre debe de contener un controlador de domino dentro del mismo.

Los controladores que se encuentran en cada sitio, estos replican su información tanto de la base de datos NTDS como la carpeta compartida SYSVOL cada 15 min

o más, con esto se garantiza que los enlaces WAN no sean saturados por las réplicas de los controladores de dominio.

A estas réplicas se las conoce como réplicas *inter-site*, si se tiene más de un controlador de dominio en el mismo sitio, uno de ellos toma el *roll* de cabecera de sitio y es el encargado de replicar la información con las otras cabeceras del sitio. Cuando se tiene varios controladores en el mismo sitio la réplica entre ellos es inmediata.

Una de las ventajas de tener varios sitios, es que se puede especificar políticas que se apliquen a los equipos que se encuentren en las subredes de ese sitio.

1.2.2.6 Unidades organizativas OU

Las unidades organizativas (OU) como se observa en la Figura 1.11 son contenedores de ADDS que ofrecen la opción de agrupar objetos, computadores y usuarios lógicamente dentro de un dominio. OU también proporcionan un marco para delegar derechos administrativos y para vincular objetos de directiva de grupo (GPO).

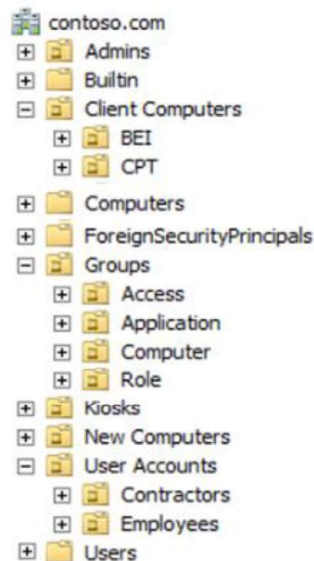


Figura 1.11 Estructura de OUs

1.3 OFFICE 365

Es la plataforma de Microsoft en la nube [10] que brinda los servicios de Exchange Online como servicio de correo, Skype For Bussines como servicio de comunicación unificada, One Drive como servicio de almacenaje en la nube, SharePoint como herramienta de colaboración, Yammer como red social corporativa, Paquete de Ofimática (Word, Excel, PowerPoint, Outlook, Publisher, Access y One note), Office en Tablet y Smartphone (Word, Excel, PowerPoint) y versiones en línea de Office como Word, Excel y PowerPoint.

Microsoft maneja una serie de licencias que combinan el uso de estos servicios a los cuales los clasifica en los siguientes planes y familias como se muestra en la Tabla 1.2.

Tabla 1.2. Planes y Familias de servicios de Office 365 [11]

Familia de servicios de Office 365	Planes
Empresa (máximo: 300 usuarios)	Office 365 Empresa Office 365 Empresa Esencial Office 365 Empresa Premium
Enterprise (número ilimitado de usuarios)	Office 365 Enterprise E1 Office 365 Enterprise E3 Office 365 Enterprise E4 Office 365 Enterprise K1
Educación (número ilimitado de usuarios)	Office 365 Educación
Administración Pública (número ilimitado de usuarios)	Office 365 Administración Pública E1 Office 365 Administración Pública E3 Office 365 Administración Pública E4 Office 365 Administración Pública K1

En la Tabla 1.4, describe los servicios y el tipo de plan general que pueden tener.

1.3.1 SERVICIOS DE OFFICE 365

Dentro de los servicios de Microsoft se tiene subcategorías como se muestra en la Tabla 1.3.

Tabla 1.3. Servicios de Office 365 [11]

Servicio	Planes
Exchange Online	Exchange Online plan 1 Exchange Online plan 2 Protección en línea de Exchange Archivado de Exchange Online Quiosco de Exchange Online
SharePoint Online	SharePoint Online plan 1 SharePoint Online plan 2
OneDrive para la Empresa	OneDrive para la Empresa con Office Online
Skype Empresarial Online	Skype Empresarial Online (Plan 1) Skype Empresarial Online (Plan 2)
Aplicaciones de Office	Office 365 ProPlus
Administración de cartera de proyectos	Project Online Project Pro para Office 365
Yammer	Yammer Basic Yammer Enterprise
Servicio de inteligencia empresarial	Power BI para Office 365
Software de diagrama en línea	Visio Pro para Office 365
Information Rights Management	Azure Rights Management (RMS)

1.3.1.1 Exchange Online

Exchange Online es el servicio de correo que proporciona Microsoft, dentro del cual se puede obtener gran capacidad en el buzón. Cada usuario tiene un almacenamiento de 50 GB.

1.3.1.2 SharePoint Online

Sharepoint Online es el servicio que proporciona Microsoft como herramienta de colaboración.

Tabla 1.4. Disponibilidad de los servicios en cada plan [11]

Servicio	Office 365 Empresa Essentials	Office 365 Empresa Premium	Office 365 Educación	Office 365 Enterprise E1 Administración Pública E1	Office 365 Enterprise E3 Administración Pública E3	Office 365 Enterprise E4 Administración Pública E4	Office 365 Enterprise K1 Office 365 Administración Pública K1
Plataforma Office 365	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Exchange Online	Sí	Sí	Sí	Sí	Sí	Sí	Sí
SharePoint Online	Sí	Sí	Sí	Sí	Sí	Sí	Sí
OneDrive para la Empresa	Sí	Sí	Sí	Sí	Sí	Sí	No
Skype Empresarial Online	Sí	Sí	Sí	Sí	Sí	Sí	No
Office Online	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Office 365 ProPlus	No	No	No	No	Sí	Sí	No
Office 365 Empresa	No	Sí	No	No	No	No	No
Project Online	No ¹	No ¹	No ¹	No ¹	No ¹	No ¹	No ¹
Yammer Enterprise	Sí	Sí	Sí	Sí ₂	Sí ₂	Sí ₂	Sí ₂
Azure Rights Management (RMS) ⁴	No ³	No ³	No ³	No ³	Sí	Sí	No ³

¹ Project Online no se incluye, pero puede adquirirse como un servicio de complemento independiente o agregarse de forma gratuita al plan de Office 365 Educación.

² Yammer Enterprise no es un componente de Office 365 Administración Pública, pero puede ser adquirido gratuitamente como una oferta independiente para cada usuario con licencia para Office 365 Administración Pública plan E1, E3 y E4. Esta oferta actualmente está limitada a los clientes que compran Office 365 Administración Pública bajo contrato Enterprise y los contratos de suscripción Enterprise.

³ Azure RMS no se incluye, pero puede adquirirse como un servicio de complemento independiente o agregarse de forma gratuita al plan de Office 365 Educación.

1.3.1.3 Skype Empresarial Online

Es la herramienta de comunicación unificada proporcionando: servicio de chat, video conferencia, compartición de recursos (programas, escritorio, pizarra, presentaciones) y telefonía.

1.3.1.4 Office online

Office Online son aplicativos Web que permiten de una manera limitada desde un navegador de Internet ver o editar archivos de Word, Excel y PowerPoint.

1.3.1.5 Dirsync

Es una herramienta que permite sincronizar el directorio activo de la Empresa con el Directorio Activo de Azure que se utiliza en los servicios de Office 365.

1.4 ESTADO DEL ARTE DE LAS TÉCNICAS DE AUTENTICACIÓN Y CONTROL DE IDENTIDAD DIGITAL

Para entender el estado del arte de la autenticación, se situará algunos conceptos de lo que es autenticar en la actualidad.

Como lo menciona la comunidad de seguridad de ESET [12], la autenticación es la forma de demostrar quien dice ser. Además, esta describe las categorías de los métodos de autenticación en el mundo offline y Online.

En el mundo offline se hace referencia a mecanismos físicos como son una cedula de Identidad, un pasaporte, una credencial, en general algún documento que pueda probar la identidad de una persona. Cuando se habla del mundo de las computadoras no existen estos documentos por lo que se debe revisar algunas

técnicas que permitan identificar a la persona quien dice ser, para esto [12] divide en tres grandes grupos, los tipos de autenticación que son: algo que solo el usuario sepa, posea y buscar un rasgo único y notable del mismo.

IBM en su base de conocimiento [13] pone en vigencia lo que es la identificación y autenticación, estos dos conceptos lo definen de la siguiente manera. La Identificación es la capacidad de reconocer de manera exclusiva a un usuario en un sistema. Mientras que la autenticación es la capacidad de demostrar que el usuario es quien dice ser.

Así mismo ESET [14] hace referencia al concepto de las cuatro A: Autenticación, Autorización, Acceso y Auditoría. Estos son los mecanismos que permiten tener un sistema de administración completo del acceso de los usuarios a un determinado sistema. Al enfocar en el mecanismo de acceso, se está hablando del control para acceder a un sistema, es decir el administrador excluye ciertos grupos o personas de los recursos y servicios de este. Realizando un símil con lo enunciado por IBM en el mundo offline, este sería una nómina de personas que puedan ingresar a una zona restringida.

La autenticación por defecto ofrece un control de acceso, ya que, si las identidades de las personas no se pudieron validar, deniega la entrada al sistema.

Para que se dé un acceso de manera segura se debe hablar de una autenticación robusta como lo menciona el artículo de PANDA-ID [15], donde hace referencia a los tres grupos de técnicas de autenticación:

- Algo que sabe el usuario.
- Algo que posee el usuario.
- Algo que la persona es.

Este artículo menciona que, para que una autenticación sea confiable, la misma debe cumplir dos de las tres técnicas mencionadas.

Así mismo [16], también habla de los tres grupos de autenticación y del equilibrio que se debe tener entre complejidad, disponibilidad y factibilidad. Por ejemplo, uno de los mecanismos de autenticación más conocidos es el uso de contraseñas, si ésta es más compleja, será más segura.

Si la contraseña es muy compleja es posible que no pueda ser recordada por el usuario y el acceso al sistema no estaría disponible, lo más seguro es que el usuario escriba la contraseña en alguna parte, esto comprometería la seguridad del sistema, por lo tanto, una contraseña compleja no sería factible.

John Shier experto *senior* en seguridad de Naked Security [17] menciona la autenticación de dos factores que en realidad tiene el mismo concepto de una autenticación robusta referenciada en PANDA-ID [15], son actualmente muy utilizadas en medios sociales como Facebook, Twitter y LinkedIn.

Todas las documentaciones mencionadas enuncian que los mecanismos de autenticación se clasifican en 3 grandes grupos, ver Figura 1.12.



Figura 1.12 Clasificación de la autenticación

En el libro de Tipton [18] menciona esta clasificación con una pequeña diferencia y divide a la categoría “de algo que es” en dos:

- Sistemas basados en una característica física del usuario o un acto involuntario del mismo, ejemplo: verificación de voz, escritura, huellas dactilares, patrones oculares.
- Sistemas basados en algo que el usuario pueda hacer de forma única. Ejemplos: Forma de andar o patrón de escritura.

Pero en contexto general es un solo grupo de “algo que es el usuario”.

Otro concepto que viene relacionado con la autenticación es la identidad digital, cuando se refiere a la palabra identidad, son una serie de rasgos, atributos o características propias de una persona como se observa en la Figura 1.13.



Figura 1.13 Identidad

Cada persona tiene una serie de características y rasgos que hacen identidades únicas y suelen ser representadas por una credencial, cédula o pasaporte, las cuales están validadas por alguna entidad.

La identidad digital es creada por la necesidad del uso del internet, cada día éstas incrementa su información como lo menciona [19], a más de la información

presentada por la persona, se suma la añadida por terceros, es decir, ésta se conforma por varias fuentes de información, como se representa en la Figura 1.14.



Figura 1.14 Identidad digital

A continuación, un análisis de los mecanismos de autenticación.

1.4.1 ANÁLISIS DE LOS MECANISMOS DE AUTENTICACIÓN

1.4.1.1 Sistemas basados en conocimiento

Los sistemas basados en conocimiento o mejor identificados como “algo que conoce el usuario”, dependen únicamente del usuario y no está atado a un dispositivo adicional, por lo que el usuario podría autenticarse desde cualquier computador en el mundo.

Dentro de esta categoría se identifica los siguientes sistemas de autenticación:

- Contraseña → Es el método de autenticación más común de identificación y autorización, como lo menciona el Ing. Tito Armas [20], este presenta como ventajas una fácil implementación, un bajo costo y es fácil de usar a excepción de que la contraseña sea muy compleja. También presentan las siguientes

desventajas: son susceptibles de adivinar, pueden ser infiltrados con fuerza bruta y fáciles de olvidar por estos suelen ser anotados en adhesivos pegados a los monitores.

- Preguntas conocidas por el usuario → También llamadas preguntas frecuentes como lo menciona Sangita Pakala [21] precursora de una lista de preguntas frecuentes que hasta el día de hoy se mantiene, estas son pistas que el usuario pueda conocer, por ejemplo: ¿La marca de tu primer celular?, ¿Cuál es tu pasatiempo favorito?, ¿Cuál es el nombre de tu primera mascota?, este tipo de autenticación es muy útil ya que delimita a que solo el usuario y pocas personas de confianza pudieran contestar estas preguntas. Se debe discriminar que tipos de preguntas deben incorporar, tratando de que el usuario pueda contestar, no se debe confundir con preguntas que sepa el usuario pero que puedan fácilmente obtener como por ejemplos: tu número telefónico, tu dirección, el nombre de tu abuelo, etc.
- Reconocimiento de imagen conocidas → este mecanismo tiene una serie de imágenes de la cual hay que elegir una. Para el cerebro humano es más fácil recordar una imagen que una contraseña, sin embargo, esta no es muy fuerte ya que fácilmente con el tiempo se puede deducir la imagen seleccionada al realizar varios intentos de acceso.
- Imagen Inteligente → este método consiste en almacenar un patrón sobre una imagen que solo el usuario conozca, esta tecnología es utilizada en los sistemas operativos de Windows 8 o superior, como lo menciona [22]. Este método el usuario realiza una serie de líneas, triángulos, cuadrados o círculos sobre la imagen de referencia, este es muy útil cuando se utilizan Tablet o dispositivos móviles.

1.4.1.2 Sistemas basados en algo poseído por usuarios

Esta categoría utiliza algo que tenga el usuario, como una llave USB, un celular, un correo personal, tarjeta de coordenadas, etc. Este método es bastante seguro ya que sin el objeto que permite el acceso se deniega el mismo.

- Llave USB → este mecanismo almacena un certificado digital en dispositivo USB, este certificado garantiza el acceso, pero implica que el usuario debe llevar la USB a todas partes, este mecanismo se utiliza para la autenticación de VPN, también se lo está incorporando para autenticar el correo como lo indica PANDA-ID en [23] al activar el factor de doble autenticación, este limita el poder acceder al sistema usando únicamente la llave USB. Por supuesto la limitación de este mecanismo es depender siempre del dispositivo USB.
- Un SMS de Celular → El envío de un mensaje de texto a un celular es considerado uno de los mecanismos de autenticación más populares al hablar de un factor de doble autenticación y es considerado un mecanismo seguro como lo menciona Denise Giust [24], donde indica que los SMS son seguros a partir de GSM, ya que estos son enviados de manera cifrada con protocolos (A5/3 o A5/1) dependiendo del país, además este artículo hace aducción de ser un buen mecanismo para ingresar a un sistema, ya que si una persona maliciosa intenta acceder al sistema, este envía un mensaje de texto alertando de que alguien está tratando de ingresar con su credencial.
- Envío de correos → El poder enviar un *token* a un correo es considerado un mecanismo de seguridad ya que el usuario es la única persona que tiene acceso a su correo personal y se puede hacer un símil de seguridad al que utiliza el enviar un SMS de celular.
- Tarjeta de coordenadas → El usuario tiene una tabla de columnas y filas, las cuales generan coordenadas que contienen una clave normalmente de dos

dígitos, la cual se la conoce como *token*, el usuario utiliza los mismos para el ingreso al sistema, este es útil, pero implica que el usuario debe de tener la tabla de *tokens*, la cual puede ser fácilmente vulnerable en el mundo actual, bastaría que cuando se esté usando la tarjeta, una cámara cercana capture la misma.

1.4.1.3 Sistemas basados en lo que es el usuario

Son los mecanismos más seguros que identifican características únicas de un usuario, tales como: iris del ojo, huella digital, voz, escritura. Lamentablemente para poder utilizar estos mecanismos es necesario de un dispositivo adicional que permitan realizar el reconocimiento de voz, reconocimiento de escritura, lecturas biométrías, etc.; limitando el ingreso a un sistema desde un computador cualquiera.

Actualmente estos dispositivos ya se están desarrollando para que sean incorporados en dispositivos móviles siendo el caso de un lector de huellas digitales o el reconocimiento facial a través de la cámara como lo menciona [25], que pretende utilizar este mecanismo para autorizaciones de pagos, pero siendo realistas estas características son consideradas en dispositivos móviles de gama alta y la gran mayoría de personas utilizan los celulares considerados de gama media.

En [26] presenta un resumen del mecanismo biométrico más conocidos, se observa en la Tabla 1.5, una comparación de los mecanismos de su fiabilidad, facilidad de uso, prevención de ataques, estabilidad, estándares, posibles interferencias y su ubicación habitual.

No se puede afirmar que una tecnología es mejor que otra, depende de las necesidades del sistema, por ejemplo, si se habla del ingreso a una planta nuclear, con seguridad se usaría un lector de reconocimiento de iris, ya que este mecanismo tiene una mayor fiabilidad, que, al hablar de una oficina comercial, donde bastaría con un lector de huellas digitales.

Lo que se puede afirmar es que entre mayor fiabilidad se necesite, el mecanismo de seguridad tiene un funcionamiento más complejo, es decir que la implementación del sistema para reconocer un iris es más compleja, que la de un lector de huellas digitales.

Dependiendo del recurso que se desea resguardar o controlar el ingreso al mismo, se puede utilizar una serie de mecanismos que pueden tener distintos niveles de seguridad.

Tabla 1.5. Cuadro Comparativo de Técnicas Biométricas [26]

	Ojo – Iris	Ojo - Retina	Huellas dactilares	Geometría de la mano	Escritura – Firma	Voz
Fiabilidad	Muy alta	Muy alta	Alta	Alta	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta	Alta
Prevención de ataques	Muy Alta	Muy alta	Alta	Alta	Media	Media
Aceptación	Media	Media	Media	Alta	Muy alta	Alta
Estabilidad	Alta	Alta	Alta	Media	Media	Media
Identificación y autenticación	Ambas	Ambas	Ambas	Autenticación	Ambas	Autenticación
Estándares	-	-	ANSI/NIST, FBI	-	-	SVAPI
Interferencias	Gafas	Irritaciones	Suciedad, heridas, asperezas	Artritis, reumatismo	Firmas fáciles o cambiantes	Ruido, resfriados ...
Utilización	Instalaciones nucleares, servicios médicos, centros penitenciarios	Instalaciones nucleares, servicios médicos, centros penitenciarios	Policía, industrial	General	Industrial	Accesos remotos en bancos o bases de datos

Otro factor que se debe de tomar en cuenta es la disponibilidad que se tenga al momento de realizar la autenticación, ya sea esta una contraseña, un lector de huellas, etc.

Se puede hacer una comparación entre los tipos de autenticación como se puede ver en la Tabla 1.6, donde se colocan a los mecanismos más conocido de cada categoría y se los compara con los parámetros de disponibilidad y el nivel de seguridad.

Se observa que entre más simple el mecanismo de autenticación, como los de conocimiento, tienen un nivel menor de seguridad, pero su disponibilidad es mayor, ya que no requieren de dispositivos adicionales, pero si se usa mecanismo de algo que posee el usuario o se utiliza un dispositivo para leer características del usuario como un lector de huellas, es más segura ya que depende de un dispositivo adicional para la autenticación, pero su disponibilidad es menor.

Tabla 1.6. Cuadro Comparativo de tipos de mecanismos de autenticación

Mecanismo	Tipo Basado en	Disponibilidad	Nivel de Seguridad
Contraseña	Conocimiento	Depende de la memoria del usuario	Seguridad media
Preguntas conocidas por el usuario	Conocimiento	Fáciles de Recordar	Seguridad media
Reconocimiento de imagen	Conocimiento	Fácil de recordar	Seguridad media
Imagen Inteligente	Conocimiento	Fácil de recordar	Seguridad media
Un celular	Algo que posee	Hoy en día prácticamente todo el mundo dispone de un celular	En conjunto de tokens temporales se considera altamente segura
Un correo personal	Algo que posee	Inmediata	En conjunto de tokens temporales se considera altamente segura
Tarjeta de Coordenadas	Algo que posee	Los usuarios tendrían que llevarla consigo, si el usuario se olvidó la tarjeta no se tiene la disponibilidad de este	Segura
Lector de Huella digital	Características únicas del usuario	El usuario necesitaría disponer de un dispositivo que permita leer la huella dactilar	Altamente segura

1.4.2 ANÁLISIS DE LOS MECANISMOS DE AUTENTICACIÓN PARA EL RESETEO DE CONTRASEÑA.

En la Figura 1.15 se muestra algunos tipos de autenticación que se podrían seleccionar para el prototipo, pero se debe considerar los siguientes puntos al momento de resetear la contraseña:

- El usuario dispondrá de una computadora básica para realizar este proceso.
- El usuario no lleva consigo dispositivos especiales para realizar este proceso.

En resumen, se debe revisar la accesibilidad de la tecnología o mecanismos necesarios para lograr la autenticación deseada.

Para esto se debe de analizar la complejidad que pueda tener un mecanismo de autenticación, por ejemplo, un dispositivo biométrico es más seguro y complejo, pero también es más costoso. ver Figura 1.16.



Figura 1.15 Mecanismo de autenticación [27]

La accesibilidad tiene una relación inversa a la complejidad, mientras menos complejo sea el mecanismo de autenticación la accesibilidad al sistema aumenta como se observa en la Figura 1.17, por ejemplo si el mecanismo solo fuera una simple contraseña no se necesitaría de ningún dispositivo adicional, para poder realizar el proceso de autenticación, si la autenticación se lo hace a través de un dispositivo móvil, en ese caso para poder realizar el mecanismo de autenticación es necesario del mismo, disminuyendo la accesibilidad al sistema y si fuera el mecanismo un medio más complejo como un lector de huellas dactilares disminuirá aún más la disponibilidad, ya que se tendría que disponer del lector de huellas dactilares para acceder al sistema.

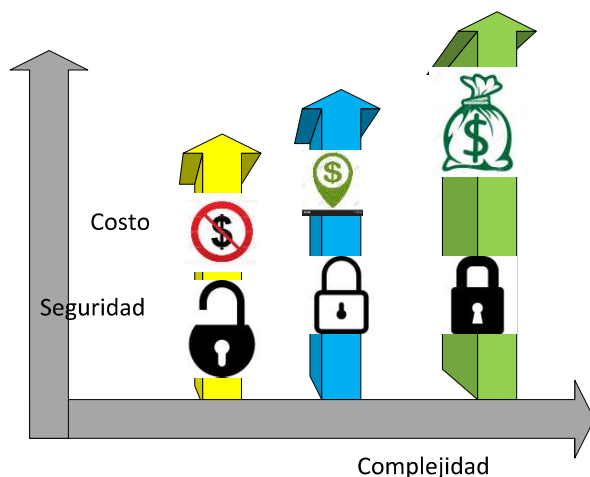


Figura 1.16 Relación costo y seguridad vs. Complejidad

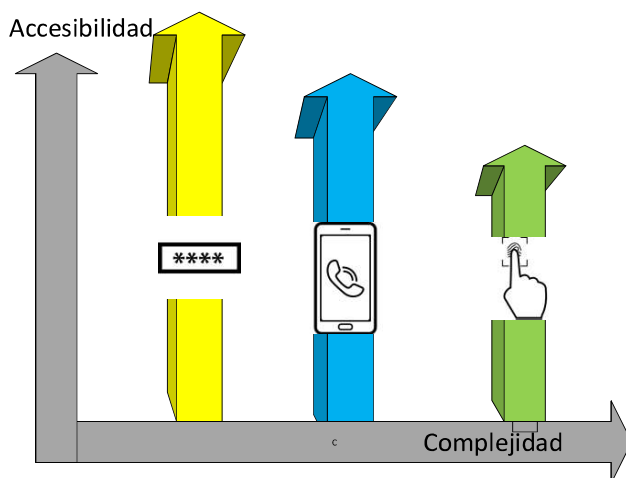


Figura 1.17 Relación Accesibilidad vs. Complejidad Mecanismo

Si bien la tecnología cada día avanza y se tiene ya dentro de un dispositivo móvil algunas tecnologías integradas como lector de huellas o reconocimiento facial, estas no se encuentran en todos los dispositivos móviles, limitándolos a que solo un pequeño grupo de personas tenga la accesibilidad a las mismas y si eso se suma a que estos mecanismos no son seguros como lo menciona [28] como por ejemplo el lector de huellas de los teléfonos iPhone son fáciles de vulnerar.

En otras palabras, si se habla de accesibilidad los mecanismos de autenticación biométricos no serían la mejor opción ya que son los menos accesibles, por eso para este proyecto se los tiene que descartar.

Si se analizan los mecanismos de algo que sepa el usuario, se verá que estos son los mecanismos de mayor accesibilidad pues no dependen de ningún complemento tecnológico que no sea un computador básico conectado al internet y la memoria del usuario. Si bien estos presentan una buena disponibilidad, no presentan una gran seguridad ya que estos mecanismos pueden ser vulnerables o fácilmente olvidados y en particular este proyecto trata de mitigar justamente el problema si es que los usuarios llegasen a olvidar la contraseña.

Si se analizan los mecanismos de algo que tenga el usuario, estos son muy seguros ya que solo pueden ser identificados con algo que tiene un usuario, como un correo electrónico o enviando un mensaje a un usuario.

Si se considera que el usuario para poder resetear la contraseña necesita de solo un computador, el mecanismo de enviar un correo electrónico personal podría ser la opción más adecuada para recuperar la contraseña siempre y cuando el usuario no se olvide la contraseña de su correo personal. Sin embargo, no es el único mecanismo que se debe considerar dentro de este grupo, si tomamos en cuenta que Ecuador en el 2013 superó el número de líneas celulares con respecto a la población como lo menciona [29] cada persona prácticamente tiene un celular a la mano, por lo que es factible enviar un mensaje de texto como mecanismo para resetear la contraseña.

No sería suficiente, con solo estos mecanismos de autenticación para que sea una autenticación fuerte, lo más recomendado es usar un doble factor de autenticación como lo recomienda la comunidad de ESET [30].

Se concluye que para este trabajo de titulación se utilizará un factor de doble autenticación, donde el primer factor de autenticación empleará mecanismos que pertenecen al grupo “Algo que tiene” por su alta confiabilidad. Para darle una mayor disponibilidad de servicio, se escogen dos mecanismos; envío de correo y SMS.

Como se descartó los mecanismos de grupo “algo que es” por su poca accesibilidad, como segundo factor de autenticación se selecciona un mecanismo del grupo “algo que sepa”, se escoge “preguntas frecuentes”, ya que brinda pistas conocidas por el usuario.

1.4.2.1 Procedimientos para cambiar la contraseña

Como primer mecanismo para cambiar la contraseña se enviará un *token* a través del sistema, este podrá ser enviado a:

- Un correo
- Un dispositivo celular

Una vez que el usuario se valide con el *token* en el sistema, el mismo le presentará una serie de preguntas que el usuario podrá escoger para contestar. Una vez que el usuario conteste la pregunta podrá cambiar la contraseña.

Capítulo 2

2 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PROTOTIPO

Este capítulo se revisará el diseño e implementación del programa tomando como referencia a la metodología de desarrollo XP (*eXtreme Programming*), dividiendo el trabajo en pequeños entregables hasta llegar a un programa definitivo.

Se adapta la metodología de XP en los siguientes puntos:

- Solo se usó una persona para el desarrollo.
- Se simulará las historias de Usuario.

XP utiliza las siguientes fases como se observa en la Figura 2.1.

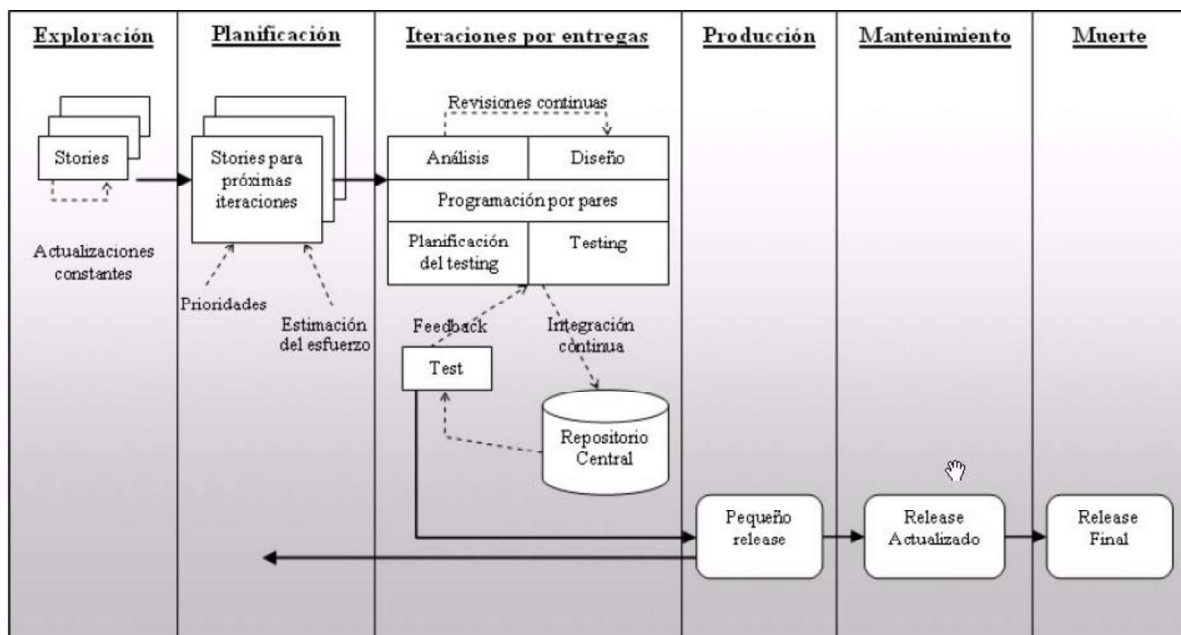


Figura 2.1 Ciclo de Vida de XP [31]

El desarrollo empieza con la primera etapa llamada exploración, donde se obtiene datos necesarios para el diseño y desarrollo del prototipo, en esta etapa se recolecta la información utilizando historias de usuarios, ver Figura 2.2.

En una segunda etapa se priorizan las historias de usuario, se desarrolla en base a estas las iteraciones necesarias y se estima el esfuerzo en horas de cada una de ellas, también se presenta cuáles son los entregables de cada una de las iteraciones.

En la tercera etapa se empieza con el desarrollo de cada una de las iteraciones, la primera iteración es la encargada de presentar el diseño del sistema, y desde la segunda iteración empieza el desarrollo del programa.

Historia de Usuario		
Número:	Usuario:	
Nombre Historia:		
Prioridad en Negocio:	Riesgo en Desarrollo:	
Puntos Estimados:	Iteración Asignada:	
Programador Responsable:		
Descripción:		
Observaciones:		

Figura 2.2 Formato de historia de usuario

2.1 DISEÑO

2.1.1 EXPLORACIÓN

2.1.1.1 HISTORIAS DE USUARIO

Se simula la iteración con un cliente, como si este solicitara los requerimientos del prototipo y se procede a llenar las tarjetas de usuarios como se observa desde la Tabla 2.1 hasta la Tabla 2.5.

Tabla 2.1. Historia de usuario # 1

Historia de Usuario	
Numero: 1	Usuario: Gerente de Sistemas
Nombre de la Historia: Cambio de Contraseña desde fuera de la empresa	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 1-6
Programador Responsable: David Abad	
<p>Descripción:</p> <p>Los gerentes cuando se encuentran fuera de la empresa y han olvidado su contraseña de correo o simplemente desea cambiarla, pero ellos no desean informar al personal de sistemas su nueva contraseña.</p>	
<p>Observaciones:</p> <p>El usuario debe de ser capaz de cambiar la contraseña desde fuera de la empresa de una manera segura sin intervención del área de sistemas.</p>	

Tabla 2.2. Historia de usuario # 2

Historia de Usuario	
Numero: 2	Usuario: Gerente de Sistemas
Nombre de la Historia: Métodos de cambiar la contraseña	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 1-4
Programador Responsable: David Abad	
<p>Descripción:</p> <p>Se debe de buscar el mejor método para que el usuario pueda cambiar la contraseña de una manera segura, con lo que pudiera tener o conocer el usuario considerando que este podría estar de vacaciones.</p> <p>Las configuraciones de seguridad que permitan realizar los cambios de contraseña deberán de poderse configurar únicamente cuando se encuentren los usuarios dentro de la empresa.</p>	
<p>Observaciones:</p> <p>Se debe de buscar mecanismos de autenticación con cosas que normalmente podría tener a la mano o con cosas que solo el usuario solo deba de conocer.</p>	

Tabla 2.3. Historia de usuario # 3

Historia de Usuario	
Numero: 3	Usuario: Gerente de Sistemas
Nombre de la Historia: Control del cambio de contraseña	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 4-6
Programador Responsable: David Abad	
<p>Descripción:</p> <p>El personal de TI debe de poder controlar quien, si puede cambiar la contraseña y quién no, el personal de TI debe de poder controlar esto desde cualquier parte del mundo ya que cuando un usuario deja la empresa esta función debe de ser deshabilitada inmediatamente.</p>	
<p>Observaciones:</p> <p>El administrador debe tener acceso a este control desde cualquier parte en el mundo.</p>	

Tabla 2.4. Historia de usuario # 4

Historia de Usuario	
Numero: 4	Usuario: Gerente de Sistemas
Nombre de la Historia: Seguridad al cambio de contraseña	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 4-6
Programador Responsable: David Abad	
<p>Descripción:</p> <p>Se debe de desarrollar al menos dos anillos de seguridad para que el usuario pueda cambiar la contraseña</p>	
<p>Observaciones:</p> <p>Se debe de buscar al menos 2 métodos de autenticación</p>	

Tabla 2.5. Historia de usuario # 5

Historia de Usuario	
Numero: 5	Usuario: Gerente de Sistemas
Nombre de la Historia: Cambiar los datos de seguridad	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados:	Iteración Asignada: 4-6
Programador Responsable: David Abad	
Descripción: El usuario debe de ser capaz de editar su información de seguridad, esta debe ser editada únicamente desde la empresa.	
Observaciones:	

2.1.2 PLANIFICACIÓN

2.1.2.1 PRIORIDADES DE HISTORIA

Se procede a considerar el orden de prioridad de cada historia, para establecer cual entregar primero, como se muestra en la Tabla 2.6.

Tabla 2.6. Prioridad Historia de Usuario

Numero de Historia	Nombre de Historia	Motivo y descripción
1	Cambio de Contraseña desde fuera de la empresa.	Esta abarca el principal objetivo del proyecto, en esta etapa se presentará pruebas de concepto de cómo se puede cambiar la contraseña en el directorio activo.
2	Métodos para cambiar la contraseña.	Esta etapa se basa en los mecanismos de validación de la persona previo a el cambio de contraseña se estima tomar unas 2 iteraciones.
3	Control del cambio de contraseña.	Esta etapa debe poder especificar el control que el usuario va a tener previo al cambio de contraseña.
4	Seguridad en el cambio de contraseña	Esta etapa debe desarrollar un conjunto de métodos para cambiar la contraseña
5	Cambiar los datos de seguridad	El usuario debe de tener la capacidad de cambiar los datos de seguridad.

2.1.3 ITERACIÓN

2.1.3.1 Análisis de iteraciones

Se crea las iteraciones en base a las historias de los usuarios, en la Tabla 2.7 muestra el número de iteración, una descripción, y que historia se encuentra relacionada.

Tabla 2.7. Desarrollo de Iteraciones

Número	Descripción	Historias involucradas
1ra Iteración	Diseño del programa en base a las historias de usuarios.	Todas
2da Iteración	Búsqueda de un usuario en el directorio activo y cambio de contraseña de este.	1
3ra Iteración	Desarrollo de los mecanismos de autenticación	2,4
4ta Iteración	Ingreso de datos del usuario necesarios para los mecanismos de autenticación. (Módulo para usuario)	5
5ta Iteración	Administración del mecanismo de autenticación (Módulo para administrador)	3
6ta Iteración	Integración de las pruebas de concepto (Módulos Cambio de contraseña).	1,2,3,4,5

2.1.3.2 Entregables por Iteración y tiempos estimados

En la Tabla 2.8 se muestran los entregables de cada iteración, así como una estimación del esfuerzo en horas que podría llevar cada iteración.

2.1.4 ITERACIÓN 1

2.1.4.1 Identificación de Objetos

Se procede a identificar los posibles objetos a partir de las historias de usuario recolectadas como se muestra en la Tabla 2.9.

Del listado identificado se procede a revisar cuales son realmente objetos como se muestra en la Tabla 2.10.

Tabla 2.8. Entregables

Entregables	Iteración	# Historias	Prioridad	Estimación de Horas
Exploración Diseño Diagrama de Caso de Uso Diagrama de Clases Diagrama de Bases de Datos Diseño de Interfaces	1	1,2,3,4,5	1	80
Prueba de concepto Búsqueda de usuario en Directorio Activo y Cambie la contraseña	2	2,4	1	20
Prueba de concepto Funciones: <ul style="list-style-type: none"> • Código aleatorio • Envío de mensaje texto • Envío de correo • Preguntas de Validación 	3	5	1	60
Prueba de Concepto: <ul style="list-style-type: none"> • Biblioteca de Clases • Asignación de Certificado digital • Autenticación de una cuanta de AD de Windows • Obtención de Datos de Usuario • Presentar datos de usuario • Consultar preguntas de Usuario • Cargar preguntas no seleccionadas por el usuario • Actualización de datos 	4	5	1	80
Prueba de Concepto <ul style="list-style-type: none"> • Acceso a solo un grupo del AD • Actualiza Base de Datos • Editar Información de usuario Añadir Preguntas	5	3	1	40
Release Módulos: <ul style="list-style-type: none"> • Cambio de Contraseña • Usuario Administración	6	1,2,3,4,5	1	80

Total de Horas estimadas		300		
--------------------------	--	-----	--	--

Tabla 2.9. Posibles objetos detectados en las historias de usuarios

N°	Posible Objeto
1	Usuario
2	Administrador
3	Contraseña
4	Método de Autenticación

Tabla 2.10. Objetos identificados de las historias de usuario

Nombre	Descripción	Objeto
Usuario	Representa el usuario con el que se va a interactuar.	SI
Administrador	Es una extensión de usuario ya que el administrador también es un usuario.	SI
Contraseña	Es una característica del usuario.	NO
Método de autenticación	No es un objeto, sin embargo, del análisis del método de autenticación pueden salir más objetos.	NO

Se identifican otros objetos como se ve en la Tabla 2.11, a partir de los mecanismos de autenticación que se revisó en la sección 1.4.2.

Tabla 2.11. Posibles Objetos detectados en los mecanismos de autenticación

N°	Posible Objeto
1	Usuario
2	Preguntas
3	Correo personal
4	Celular

Se identifica cuáles son realmente objetos con se muestra en la Tabla 2.12.

Tabla 2.12. Objetos detectados

Nombre	Descripción	Objeto
Usuario	Que representa el usuario con el que se va a interactuar.	SI

Preguntas:	Se pueden considerar como un objeto que al mismo tiempo serán características del usuario.	SI
Respuestas	Son características de la pregunta.	NO
Correo personal	Es una característica del usuario.	NO
número de celular	Es una característica del usuario.	NO

2.1.4.2 Diagrama de Clases

Una vez identificados los objetos se diseña el diagrama de objetos utilizando UML como se muestra en la Figura 2.3.

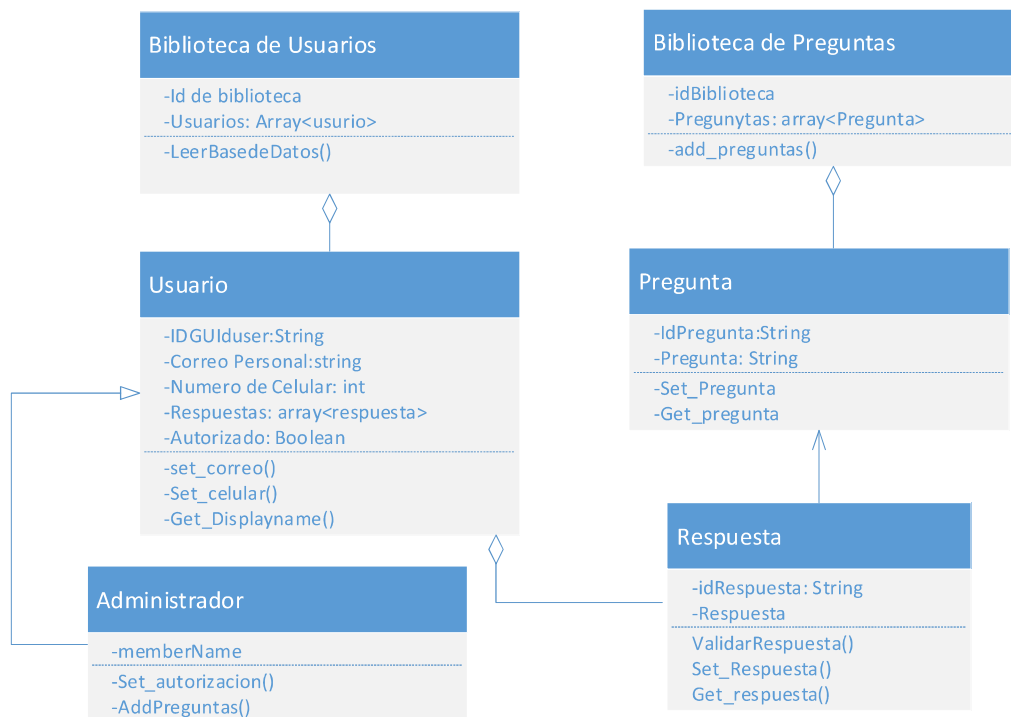


Figura 2.3 Diseño UML de acuerdo con los objetos identificados

2.1.4.3 Diseño de Base de Datos

A partir del diseño de clases se genera el diseño de la base de datos como se muestra en la Figura 2.4.

2.1.4.4 Casos de Uso

Se procede a crear un diagrama de uso para indicar para representar de manera general la interacción del usuario con el sistema como se observa en la Figura 2.5.

2.1.4.5 Módulo Ingreso de Datos de Verificación

Este módulo se utilizará para ingresar la información necesaria para validar la identidad del usuario al momento de cambiar la contraseña, la Figura 2.6 representa el proceso de ingreso del usuario al módulo.

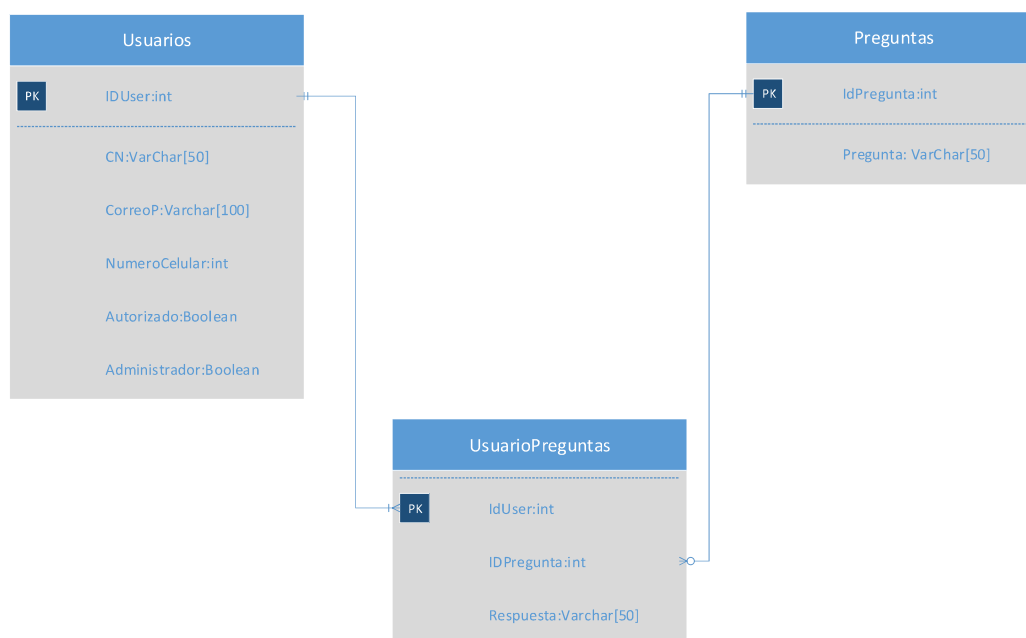


Figura 2.4 Diseño de la Base de Datos

Una vez que el usuario ingresa en el módulo, ver Figura 2.7, este procede a cargar los datos del usuario, luego podrá seleccionar las siguientes opciones:

- Editar información de usuario.
- Añadir pregunta.
- Editar respuesta.

En la Figura 2.8 se representa las opciones que puede seleccionar el usuario:

- Información de Usuario (B), este podrá editar su información de correo y número de celular.
- Añadir pregunta(C), podrá añadir preguntas que servirán para la validación en el módulo de cambio de contraseña.
- Actualizar repuestas(D), podrá actualizar sus respuestas que ha añadido.

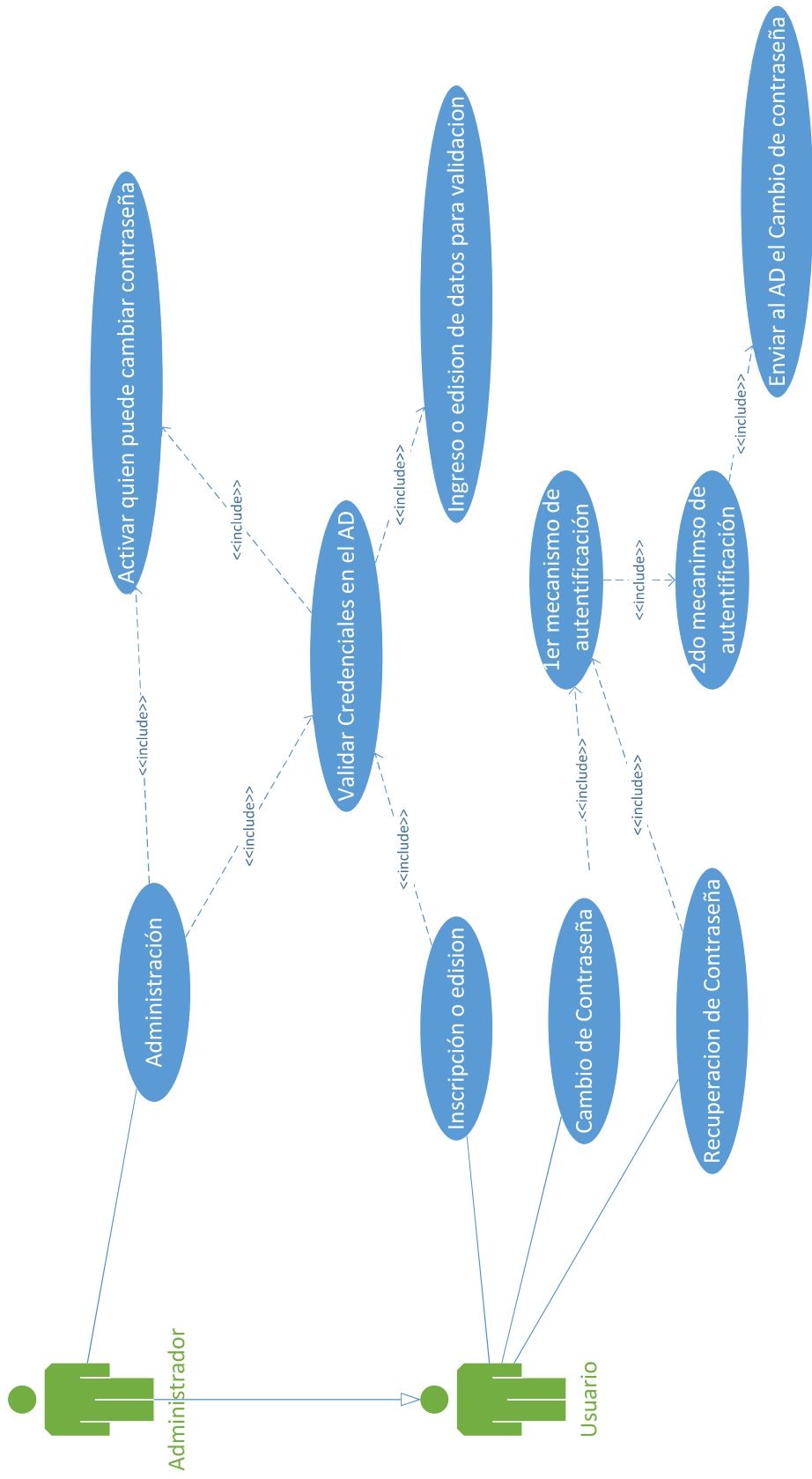


Figura 2.5 Diagrama casos de uso

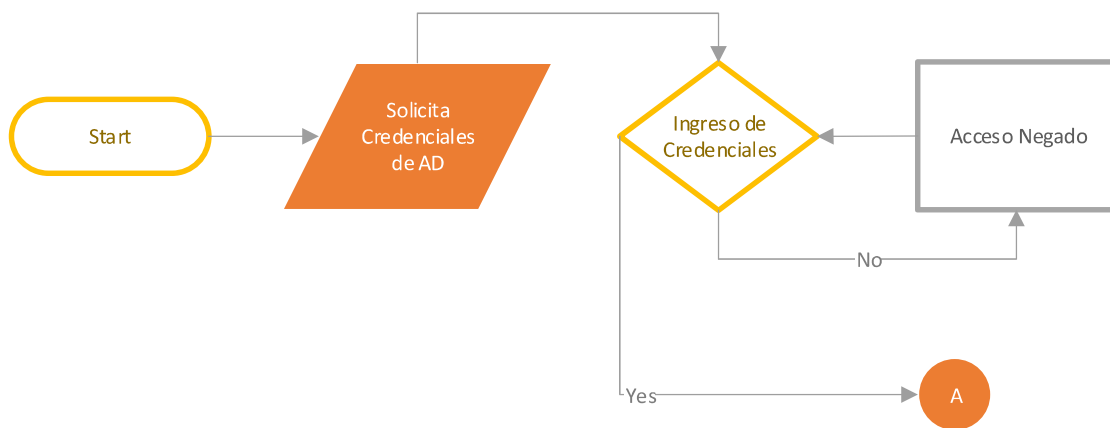


Figura 2.6 Ingreso Módulo Usuario

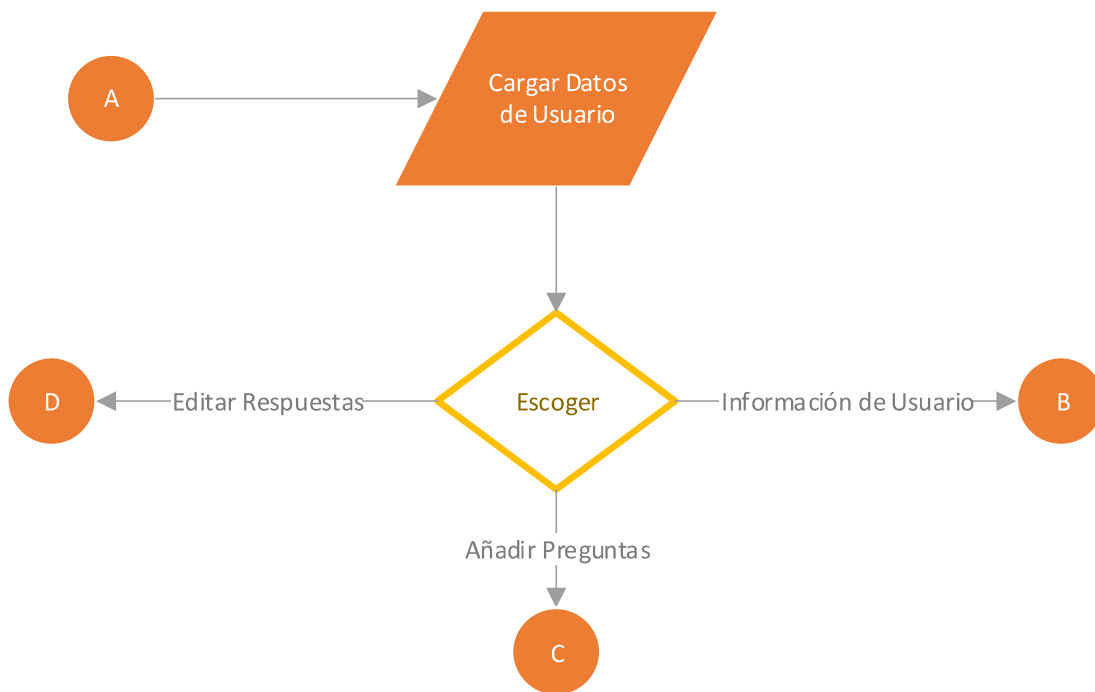


Figura 2.7 Opciones de Módulo Usuarios

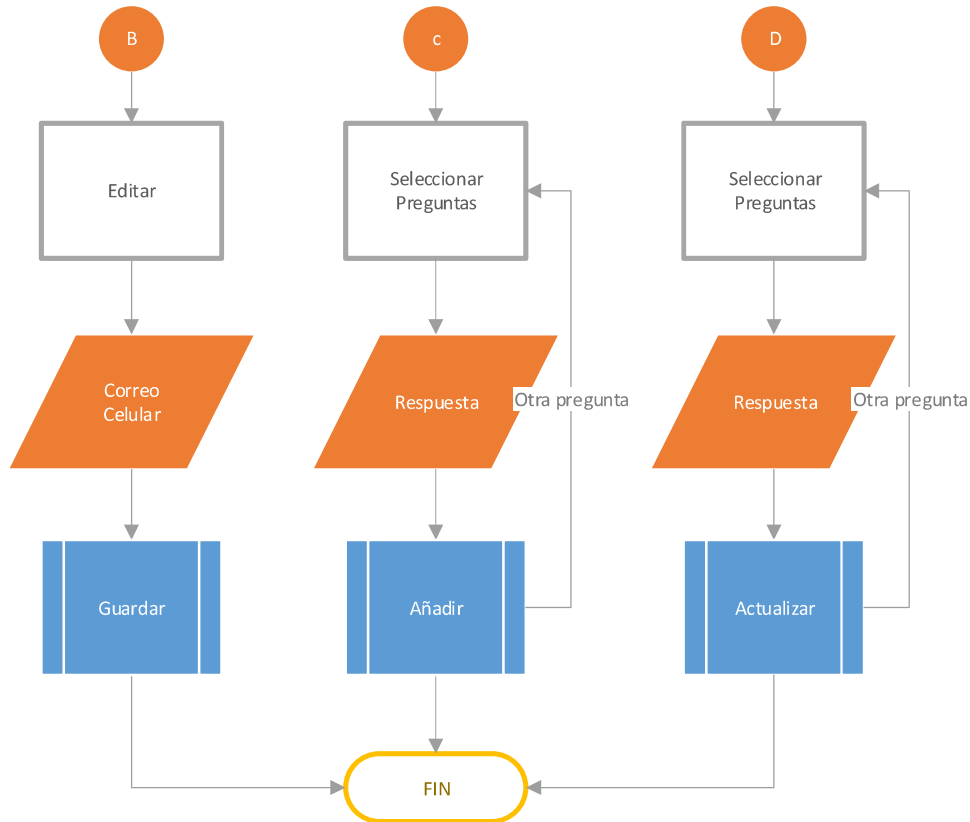


Figura 2.8 Actualización de datos

2.1.4.6 Módulos de Administración

El módulo de Administración tendrá las funciones de:

- Actualizar los usuarios del directorio Activo en la Base de Datos.
- Autorizar a los usuarios que pueden cambiar contraseña.
- Actualizar datos de correo y celular del usuario.
- Añadir preguntas al Banco de Preguntas.

En la Figura 2.9 se representa el ingreso al módulo de administración, solo los usuarios que pertenecen a un grupo del AD estarán autorizados a acceder al sistema.

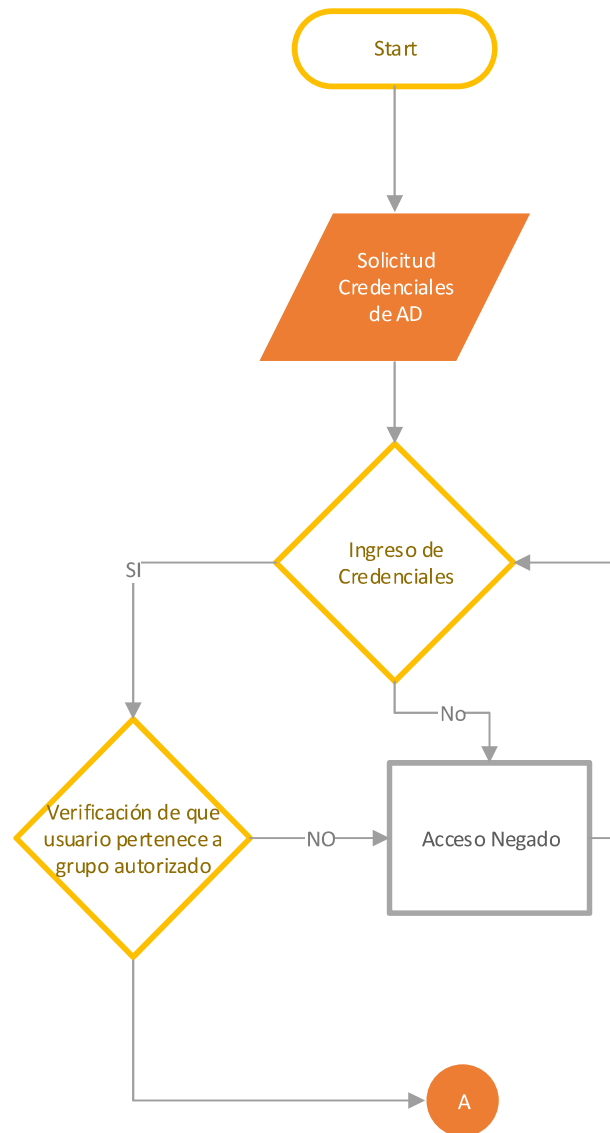


Figura 2.9 Ingreso Módulo de Administración

En la Figura 2.10 se representa los procesos que pueden realizar el módulo de administración, el módulo permite escoger dos opciones:

- Actualizar Usuarios (B), este actualiza los usuarios del AD en la base de datos, así como permite editar el correo, número de celular y si está o no autorizado para cambiar la contraseña.
- Añadir Preguntas (C), permite observar el banco de preguntas para validación de identidad y añadir más preguntas al mismo.

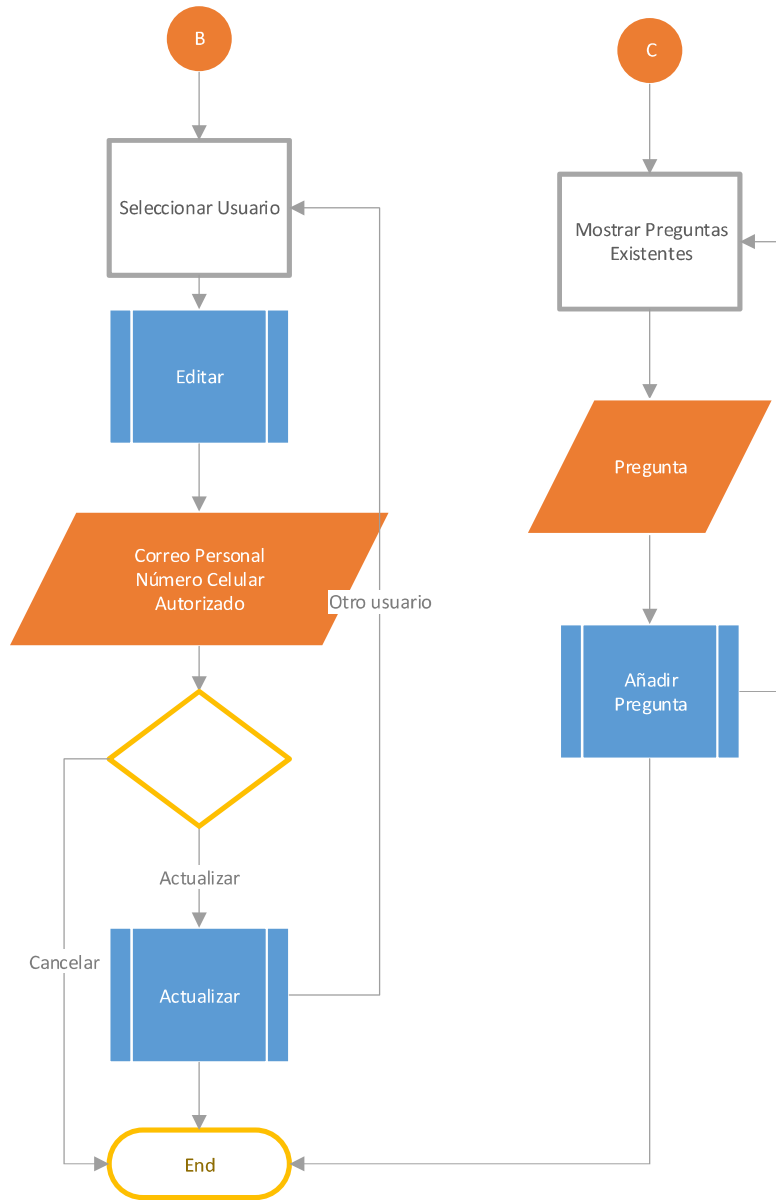


Figura 2.10 Procesos del módulo administración

2.1.4.7 Módulo de Cambio de Contraseña

Este módulo permitirá el cambio de contraseña, previo a que el usuario ingrese a este módulo debe de completar el módulo de usuarios, y el administrador tiene que darle permiso para que pueda cambiar la contraseña, ver Figura 2.11.

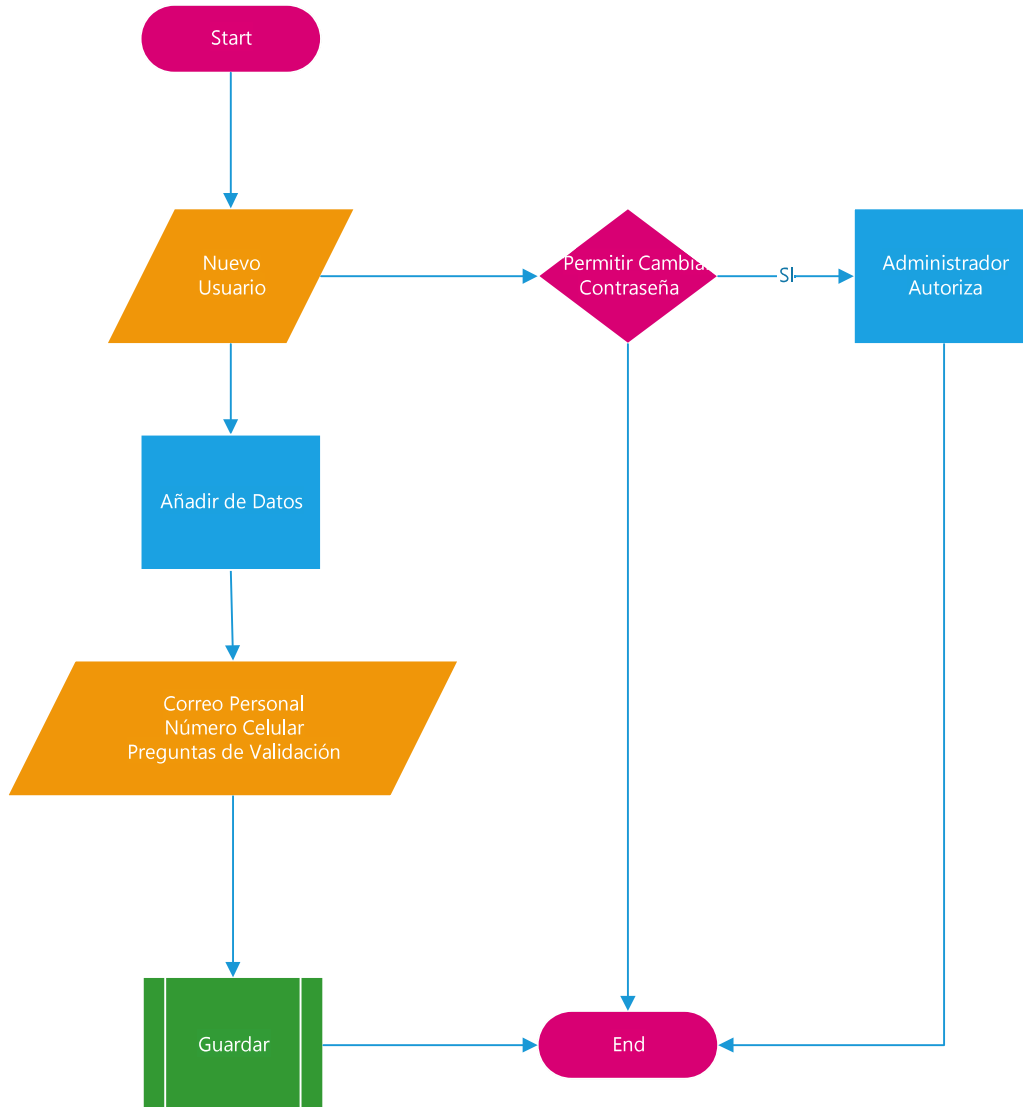


Figura 2.11 Información para validación

La Figura 2.12 representa el módulo que permitirá cambiar la contraseña, el usuario debe de ingresar su cuenta del directorio activo, si no es correcta el sistema le permitirá hasta 3 intentos posteriormente a eso sistema se bloqueará.

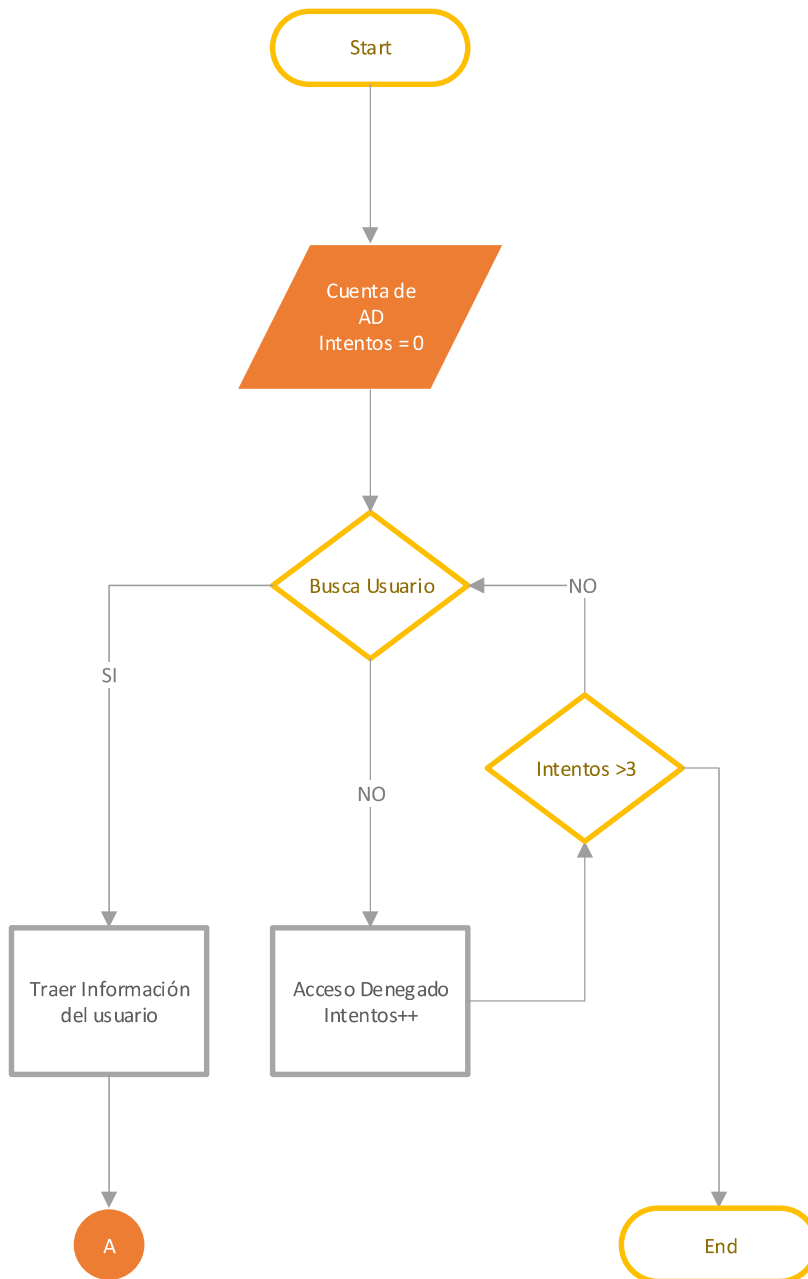


Figura 2.12 Ingreso al cambio de contraseña

Una vez que ingresa el usuario, el sistema deberá tener la información del usuario necesaria para poder validar la identidad del mismo, se procede con la validación del primer anillo, ver Figura 2.13, en el primer anillo de seguridad debe enviar un código aleatorio a través de un mensaje de texto o un correo.

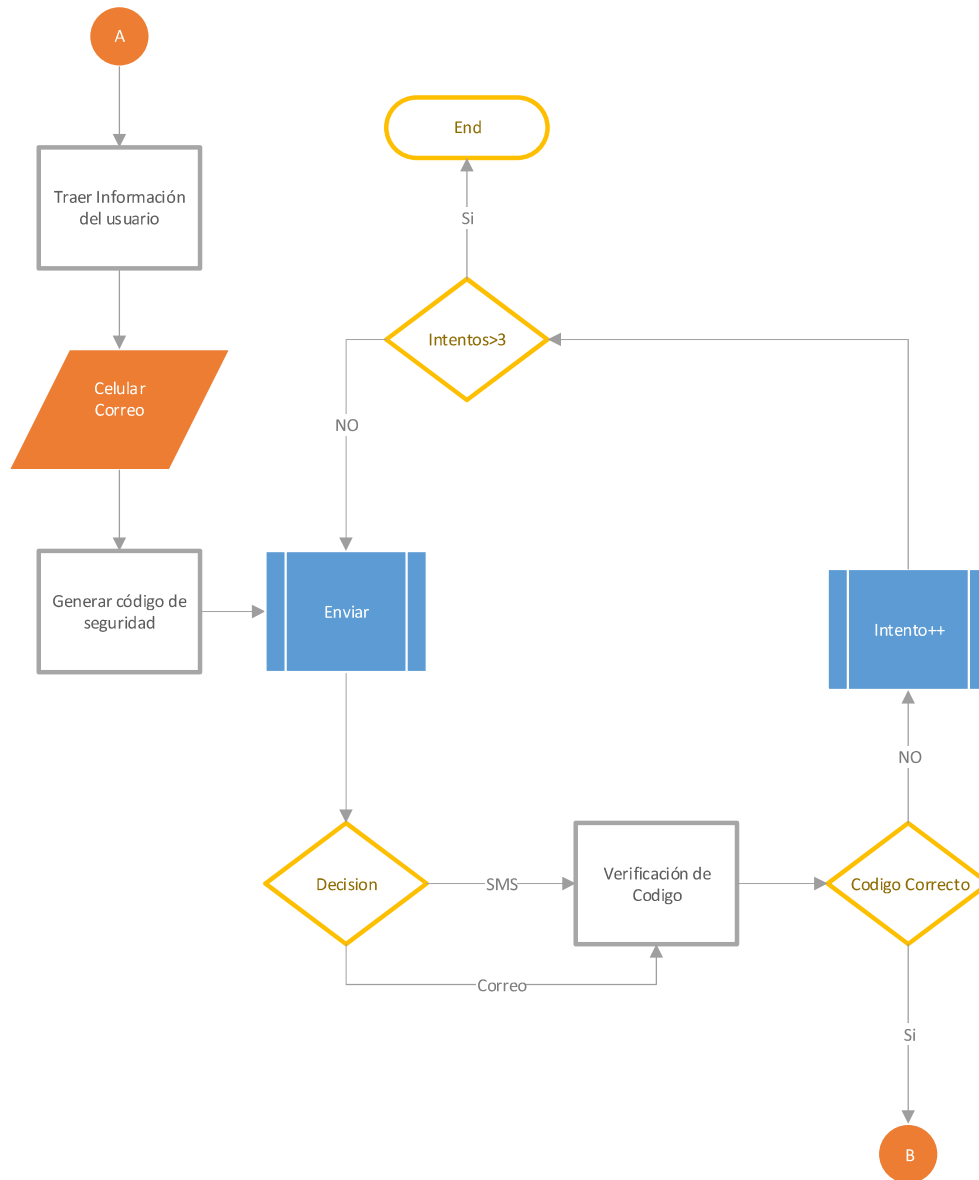


Figura 2.13 Primer anillo de seguridad

En el segundo anillo de seguridad, ver Figura 2.14, se seleccionará una de las preguntas de validación, una vez que se contesta la pregunta permitirá el cambio de la contraseña.

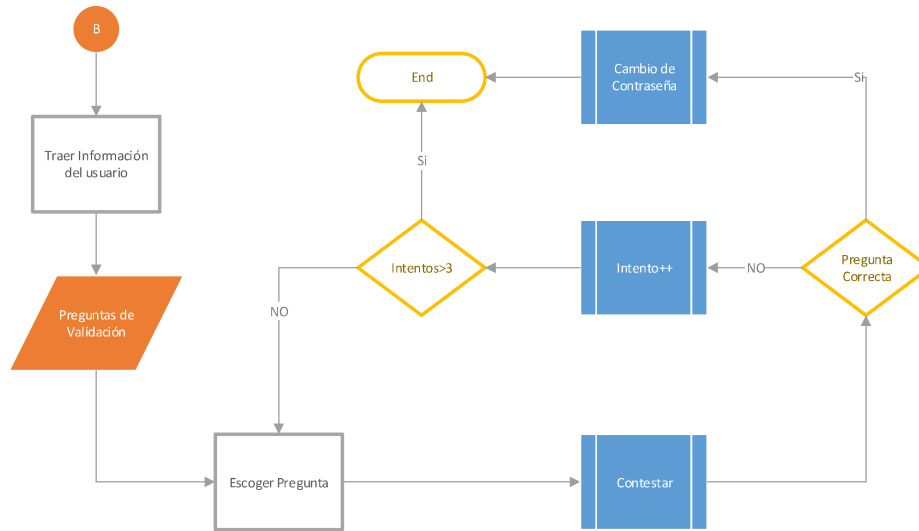


Figura 2.14 Segundo anillo de seguridad

2.2 PREPARACIÓN DE UN AMBIENTE CONTROLADO

2.2.1 MÁQUINAS VIRTUALES

Para la preparación de las máquinas virtuales, se utilizó las bondades que presenta Azure como servicio de Infraestructura IaaS [32].

Se utilizó una suscripción proporcionada del beneficio de MSDN de Microsoft activada con el módulo de Visual Estudio Profesional Enterprise, ver Figura 2.15, esta suscripción implica un bono de 150 dólares de uso mensual del portal de Azure.

SUSCRIPCIÓN	ID...	MI ROL	GASTO ACTUAL	ESTADO
Microsoft Partner Network	32050f8...	Propietario	\$ 32,03	Activo
Visual Studio Enterprise con MSDN	f886711...	Administrador de cuenta	\$ 48,96	Activo

Figura 2.15 Suscripción de MSDN

En el portal de Azure, ver Figura 2.16, se crean 2 máquinas virtuales:

- Un controlador de dominio.
- Un servidor web.

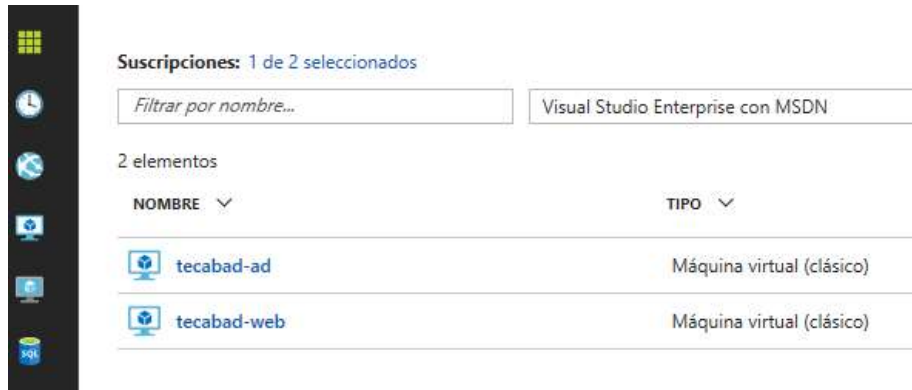


Figura 2.16 Máquinas provistas

2.2.2 CARACTERÍSTICAS DE LOS EQUIPOS

Azure agrupa los recursos de sus máquinas virtuales por categorías dependiendo de la cantidad de procesador, memoria o disco, para el caso de las máquinas virtuales se provee estas dentro de la categoría A1, como se observa en la Figura 2.17.

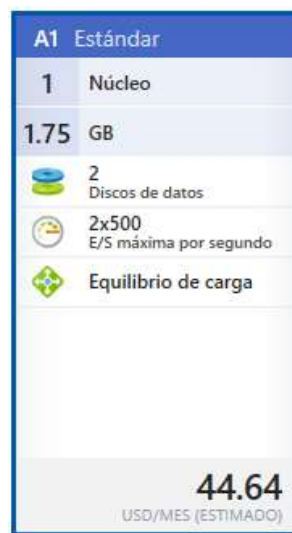


Figura 2.17 Características de las MV

Las máquinas presentan las siguientes características:

- Un procesador de 2.10 GB.
- Permite el uso de hasta 2 discos de datos no incluye disco C:
- Un equilibrio de carga que permite publicar atreves de una dirección pública los servicios.
- Las máquinas son provistas con Windows Datacenter 2012 R2 (el costo de la licencia de Windows de la máquina ya se encuentra incluida en la provisión de esta).

2.2.3 CONFIGURACIÓN DE LA VNET

Se procede a crear y configurar la VNET, se asigna el espacio de direcciones 10.0.0.0 /24 como se observa en la Figura 2.18.

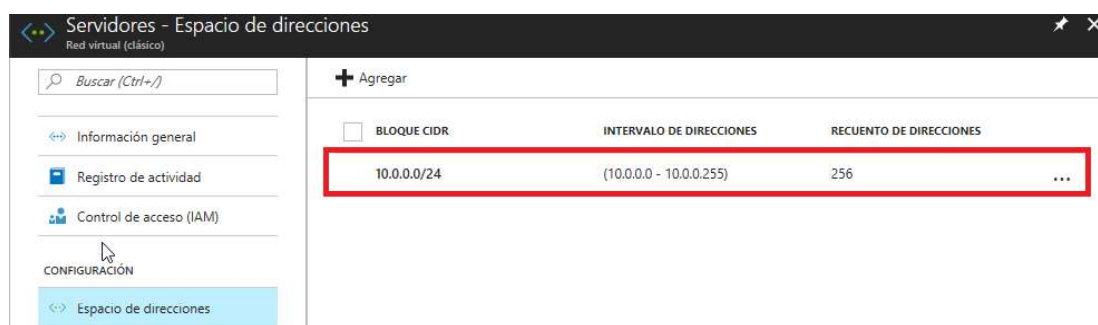


Figura 2.18 Espacio de Direcciones de la VNET

2.2.3.1 Subredes

Una vez seleccionado el espacio de direcciones, se segmenta la misma en las subredes como se muestra en Figura 2.19, se divide en dos, una para servidores y otra para segmento local.

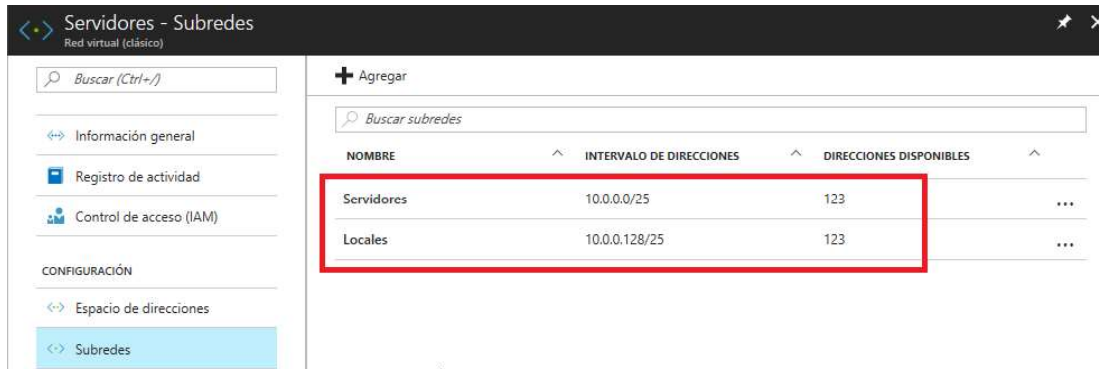


Figura 2.19 Subredes de Azure

2.2.4 CONFIGURACIONES DE LAS VNIC DE LAS MV

Se configura las direcciones IPs, ver Figura 2.20, en las tarjetas de red virtual de Azure, estas configuraciones no se las puede hacer a través del sistema operativo.

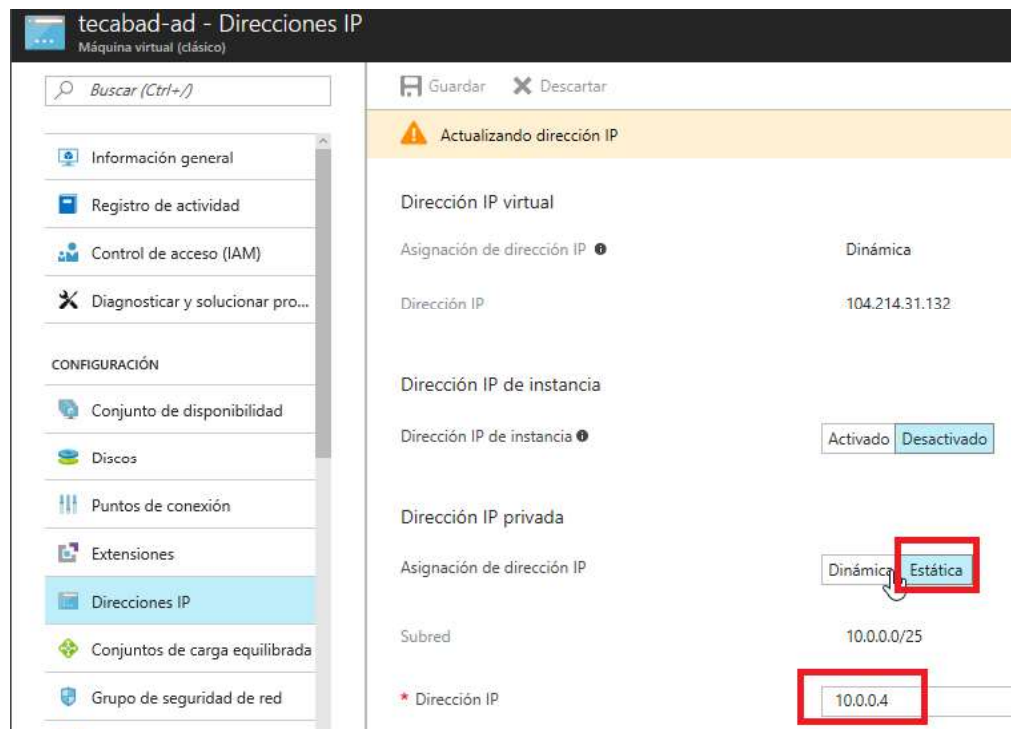


Figura 2.20 Asignación de IP estática

Para el caso de la máquina virtual que contendrá el controlador de dominio, se asigna su IP de manera estática como se observa en la Figura 2.20, este servidor contendrá el servicio de resolución de Nombres de dominio (DNS), Azure reserva siempre las primeras direcciones IP como puerta de enlace que utiliza para la comunicación entre otros segmentos de red, por lo que la primera IP que se puede utilizar para este propósito es 10.0.0.4 que será la IP asignada a la máquina tecabad-ad.

2.2.5 IMPLEMENTACIÓN DE UN CONTROLADOR DE DOMINIO

En la máquina Virtual tecabad-ad, se procede con la provisión de un controlador de dominio.

Desde la consola de server manager, ver Figura 2.21, se selecciona *manage*.

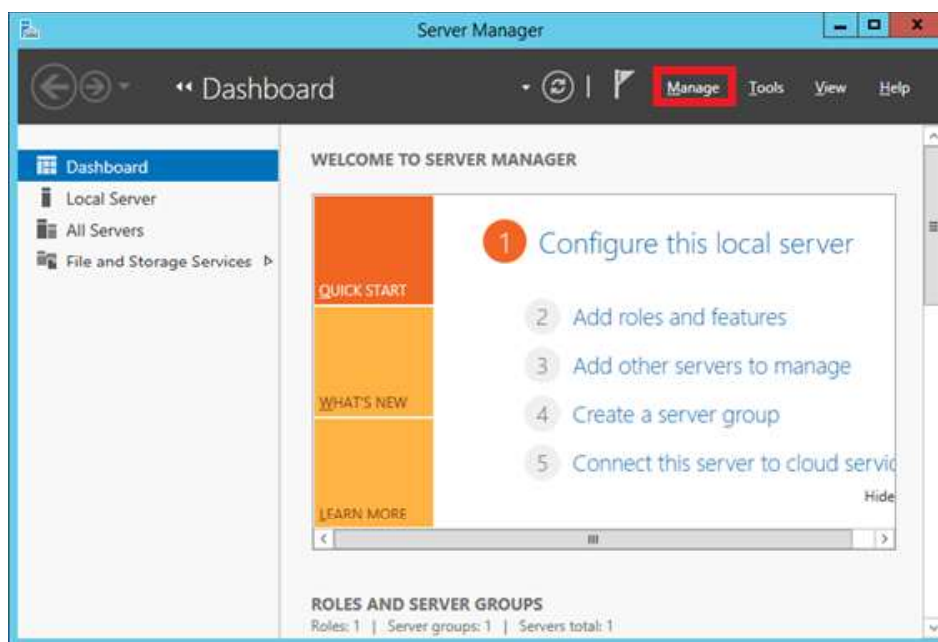


Figura 2.21 Server Manager

En el despliegue del menú se seccionó *Add Roles and Features*, ver Figura 2.22, y aparecerá un asistente de configuración de servidor.

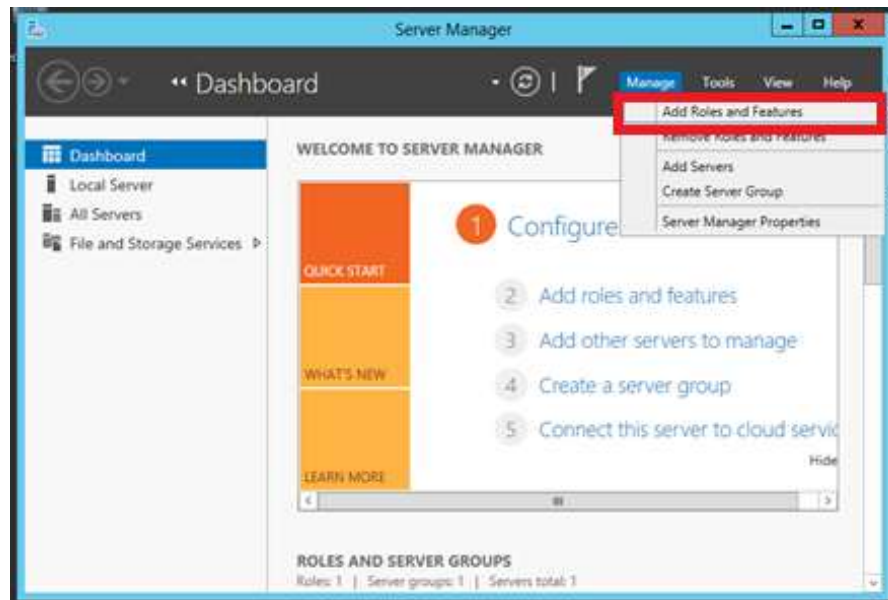


Figura 2.22 Añadir roles y características

Se especifica el servicio de directorio activo, ver Figura 2.23.

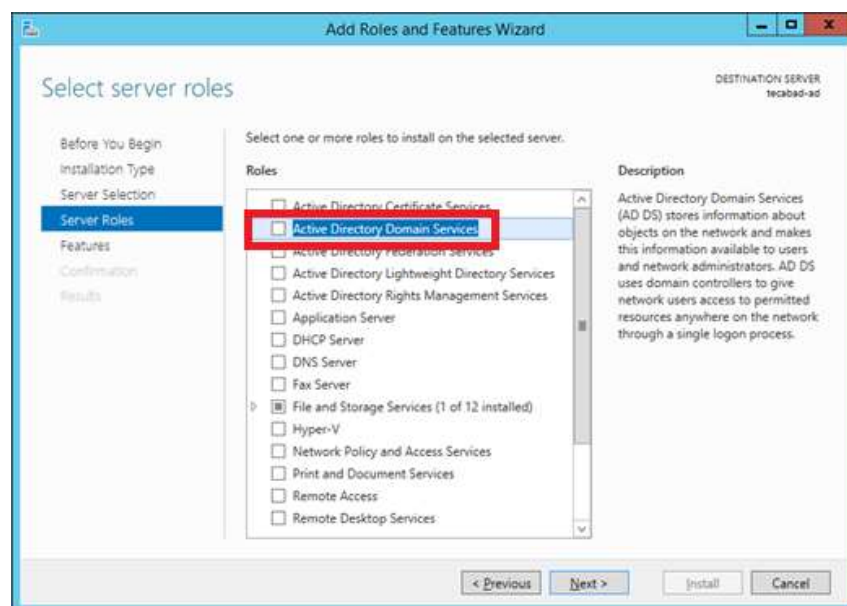


Figura 2.23 Server Roles

Se aceptan las características necesarias para activar el servicio, ver Figura 2.24.

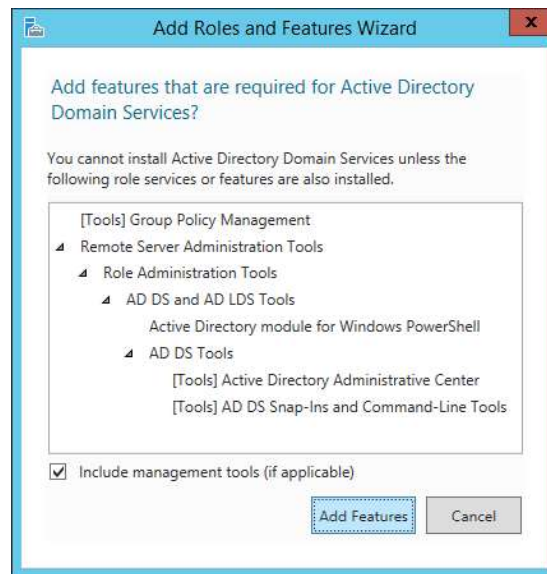


Figura 2.24 Características necesarias para el servicio de dominio.

Se da clic en add features, aparecerá la pantalla descriptiva del servicio de Directorio Activo, ver Figura 2.25, dar clic en siguiente.

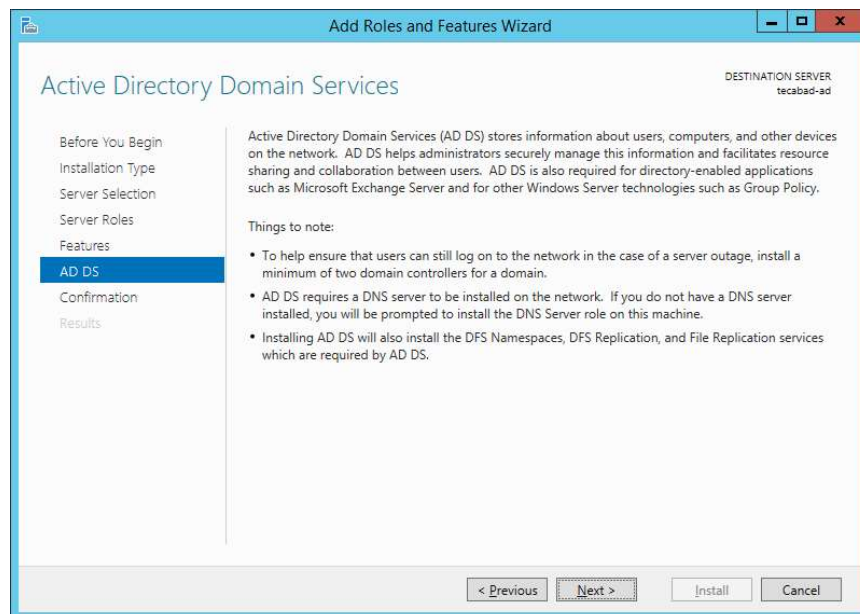


Figura 2.25 Información del servicio a implementar

En la pantalla de confirmación dar clic en el botón instalar, ver Figura 2.26.

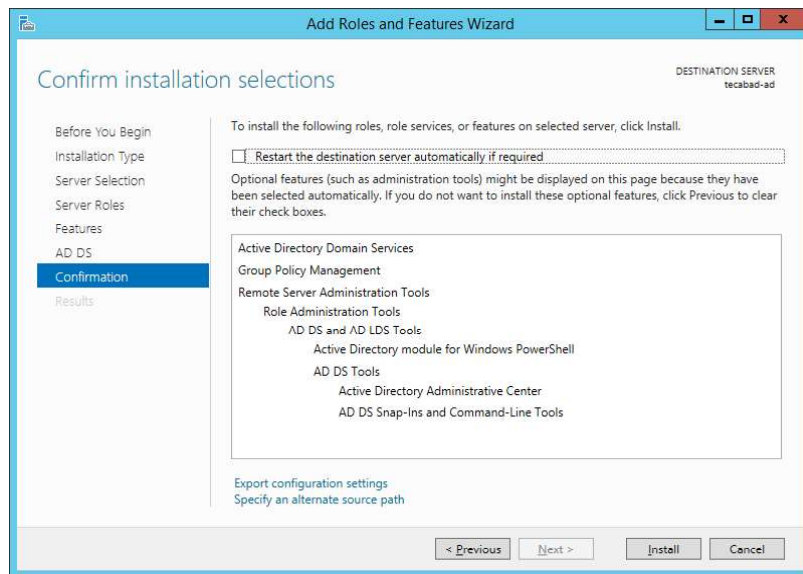


Figura 2.26 Confirmación de servicio a instalar

Una vez que se instala los componentes necesarios para el servicio de directorio activo, se procede a promover el primer controlador de domino, para ello dar clic en promover este servidor a un controlador de domino, ver Figura 2.27.

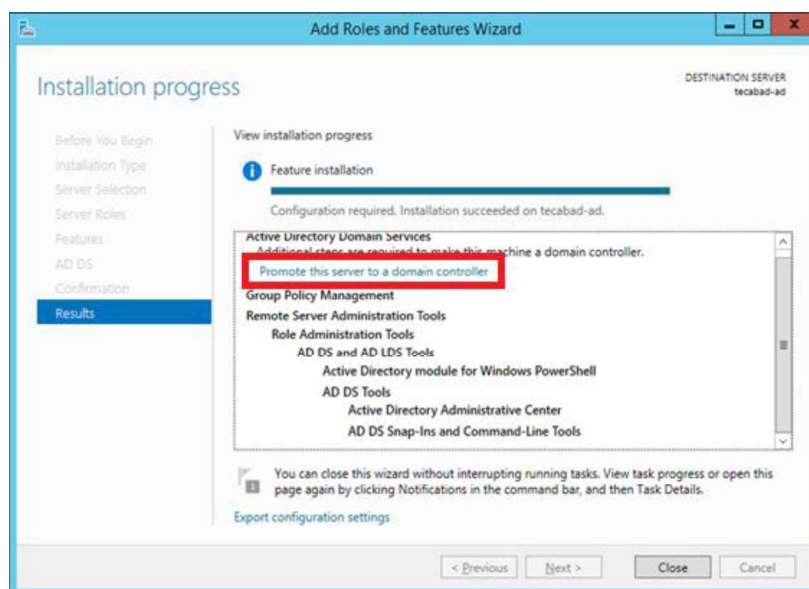


Figura 2.27 Promover al servidor como un controlador de dominio

Aparece el asistente de configuración del directorio activo, se selecciona añadir un nuevo bosque de dominio, y se especifica el nombre de este, ver Figura 2.28.

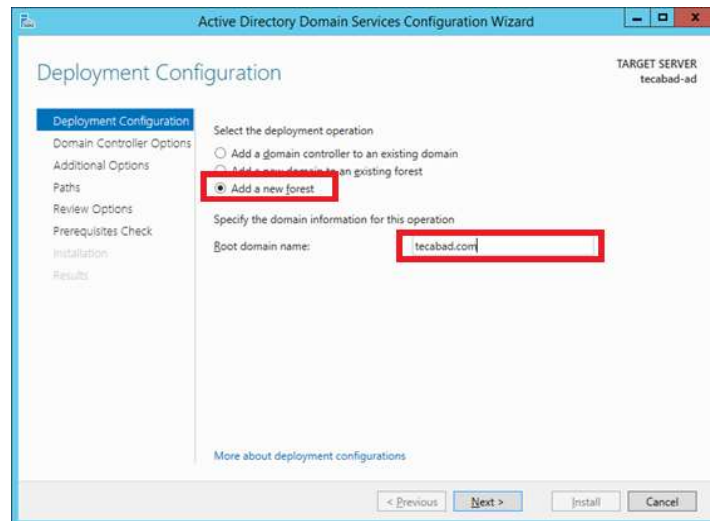


Figura 2.28 Añadir nuevo Bosque de AD

Se especifica el nivel funcional del Bosque y dominio en Windows 2012 r2, al ser el primer controlador de domino, se especifica automáticamente como catálogo global, que tenga el servicio de DNS y la clave de restauración, ver Figura 2.29

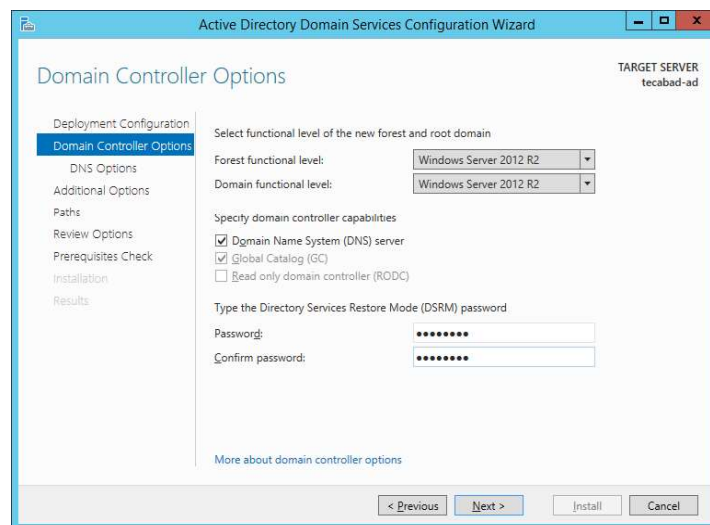


Figura 2.29 Opciones del controlador de dominio

Se asigna el nombre netbios o también conocido como nombre corto del dominio “tecabad”, ver Figura 2.30.

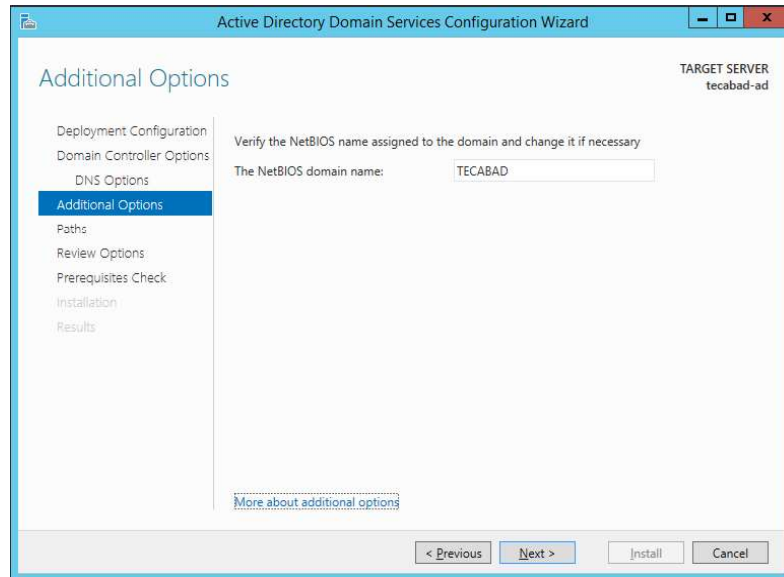


Figura 2.30 Nombre NetBIOS del dominio.

Se especifica la ruta para: base de datos, registros y carpeta compartida SYSVOL, ver Figura 2.31.

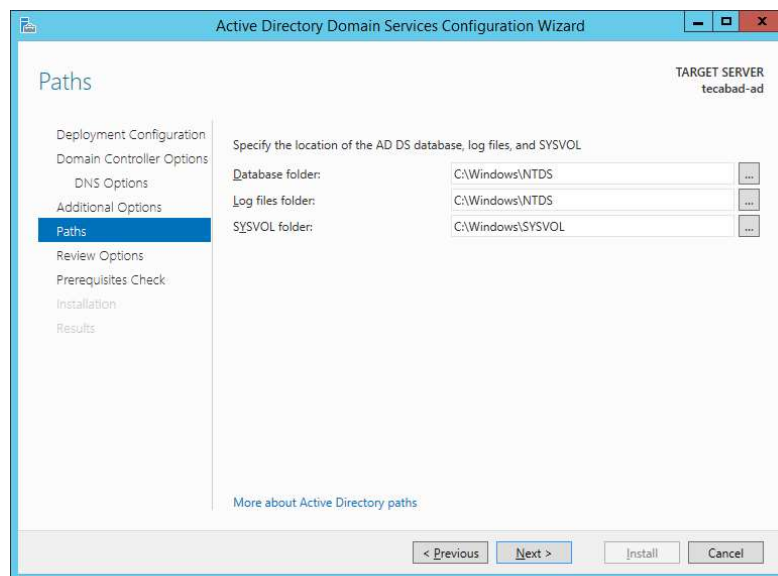


Figura 2.31 Especificación de ubicación base de datos y carpeta compartida

En la pantalla de resumen de configuración, ver Figura 2.31, clic en siguiente, el asistente busca los requerimientos necesarios previa instalación.

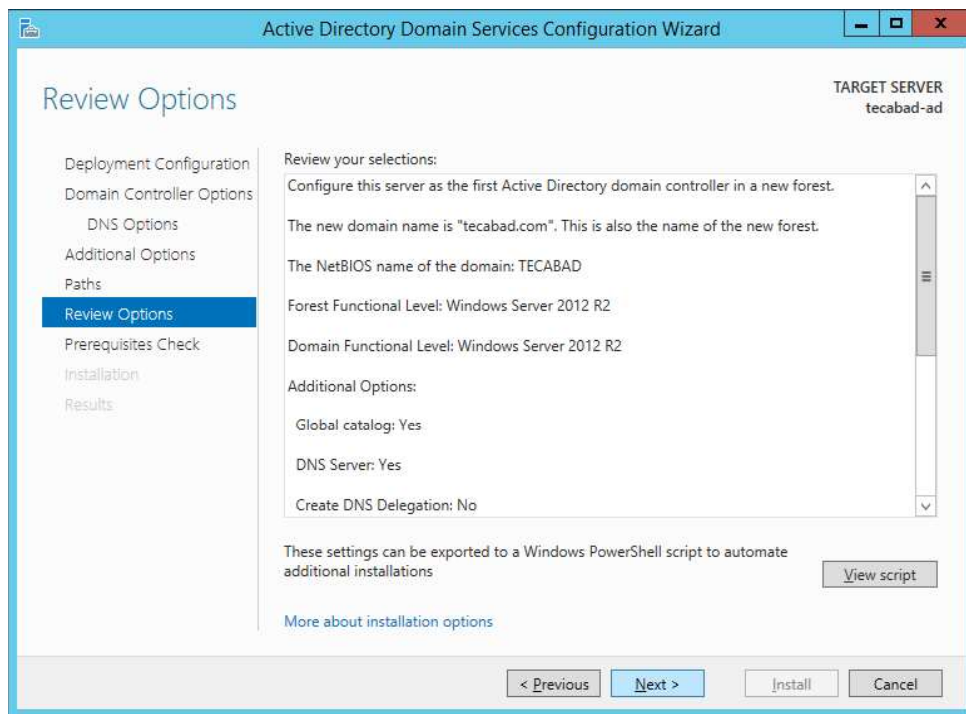


Figura 2.32 Resumen de configuración controlador de dominio

Se revisa el resultado de los prerequisites, ver Figura 2.33, en ellos muestra tres advertencias:

- Los servidores NT 4.0 o inferiores no podrán pertenecer al dominio.
- El servidor no tiene asignado una IP estática.
- El servidor va a reiniciar.

Nota: si bien la advertencia muestra que el servidor esta con una IP dinámica, desde el portal de Azure se realizó la reserva para que la VNET siempre asigne la misma IP a este servidor.

Al no encontrar problemas con las advertencias que muestra el asistente, se continúa con la instalación.

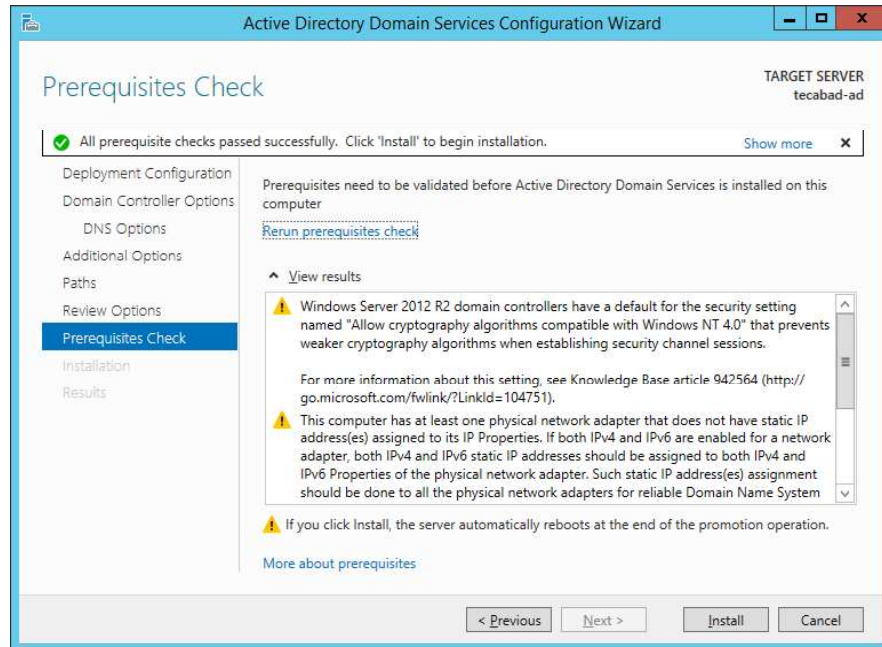


Figura 2.33 Chequeo de Prerrequisitos

Se verifica la instalación de controlador de dominio, para esto se comprueba en la consola de Usuarios y equipos del AD, la aparición del controlador de dominio, ver Figura 2.34.

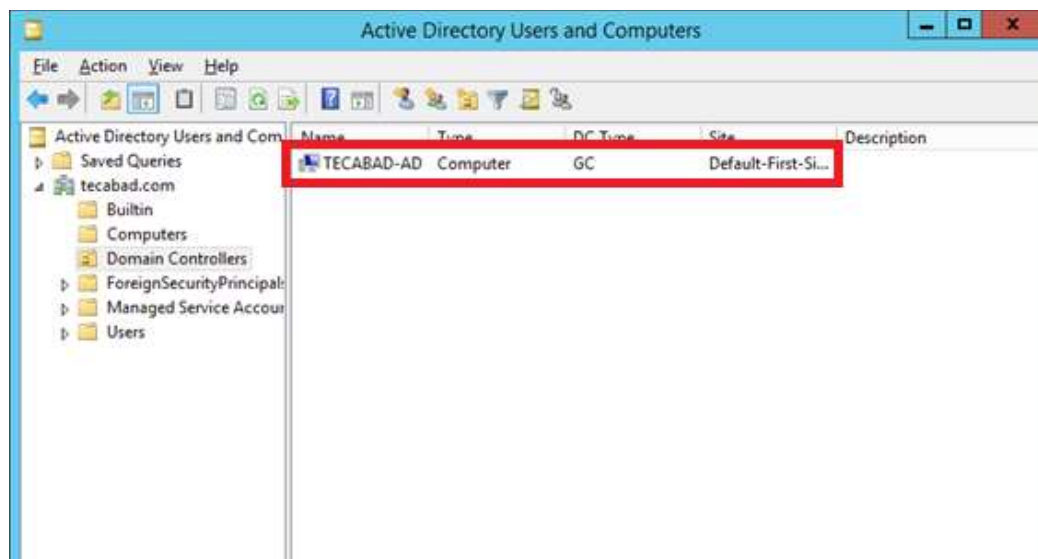


Figura 2.34 Consola de Active Directory Users and Computers

Se procedió a crear la subred en la que se encuentra el servidor, ver Figura 2.35, así como el renombrando del *site* por defecto con el nombre de servidores, también se relaciona la subred 10.0.0.0/24 al sitio.

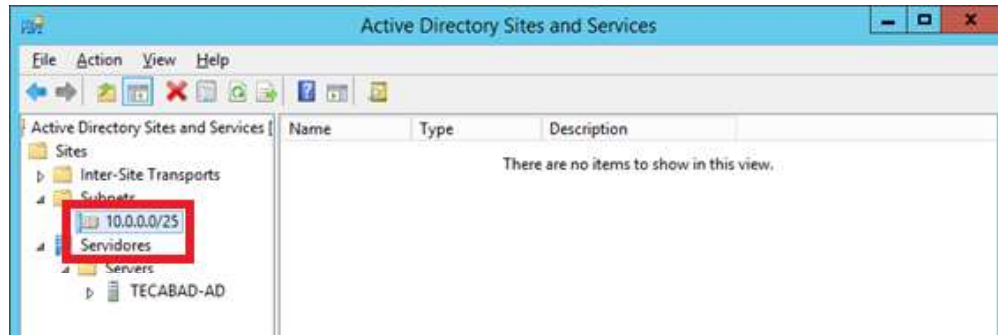


Figura 2.35 Consola de Active Directory Sites and Service

2.2.6 CONFIGURACIÓN DEL DNS EN LA VNET DE AZURE

Una vez configurado el directorio Activo y el servicio de DNS se modifica de las configuraciones de la VNET los DNS, ver Figura 2.36, para que los equipos en esa VNET sean asignados como DNS al controlador de dominio.

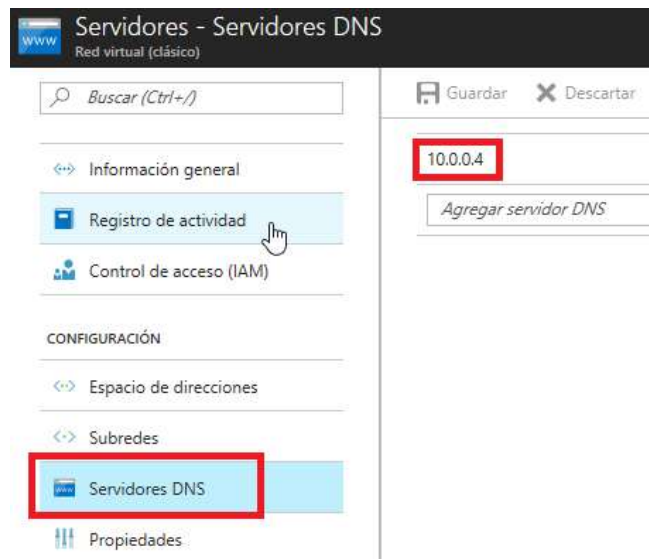


Figura 2.36 Configuración de DNS

2.2.7 CONFIGURACIÓN DE UNA ENTIDAD CERTIFICADORA DE WINDOWS

Se procede a levantar la entidad certificadora, para generar un certificado digital que se utilizará para codificar la comunicación entre el navegador del cliente y el servidor.

Se activa la entidad certificadora en el mismo controlador de dominio, para lo cual se activa el rol de entidad certificadora raíz, ver Figura 2.37.

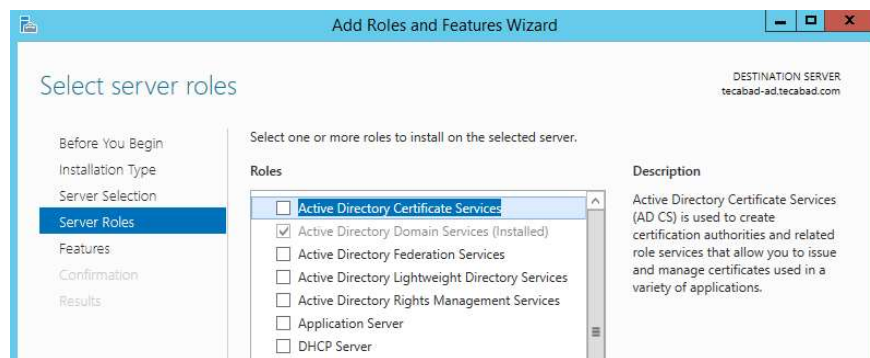


Figura 2.37 Activación del servicio de entidad certificadora

Se añade las características necesarias como se observa en la Figura 2.38.

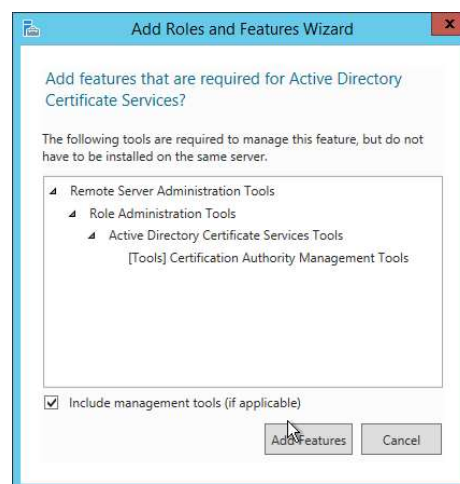


Figura 2.38 Características de entidad certificadora

También se activa el rol de enrolamiento web para poder firmar las solicitudes de certificado, ver Figura 2.39.

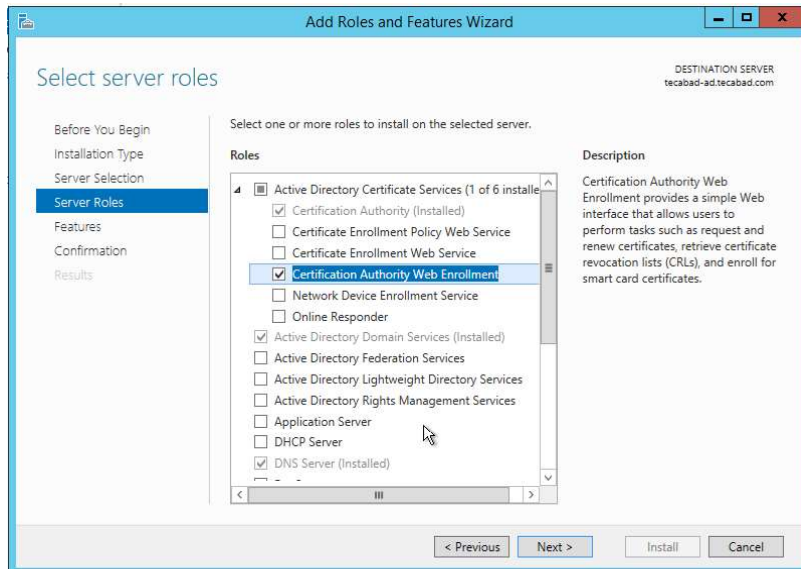


Figura 2.39 Enrolamiento Web de certificados

Se confirma la instalación de la entidad certificadora, ver Figura 2.40.

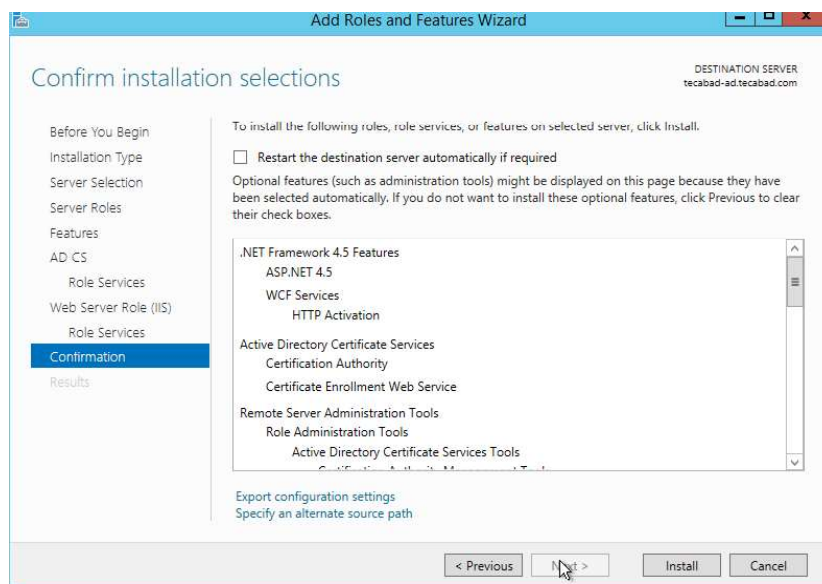


Figura 2.40 Resumen de configuración entidad certificadora Raíz

Una vez que termina de instalar los servicios de entidad certificadora se procede con la configuración. Para acceder a la configuración se da clic en el link como se muestra en la Figura 2.41.

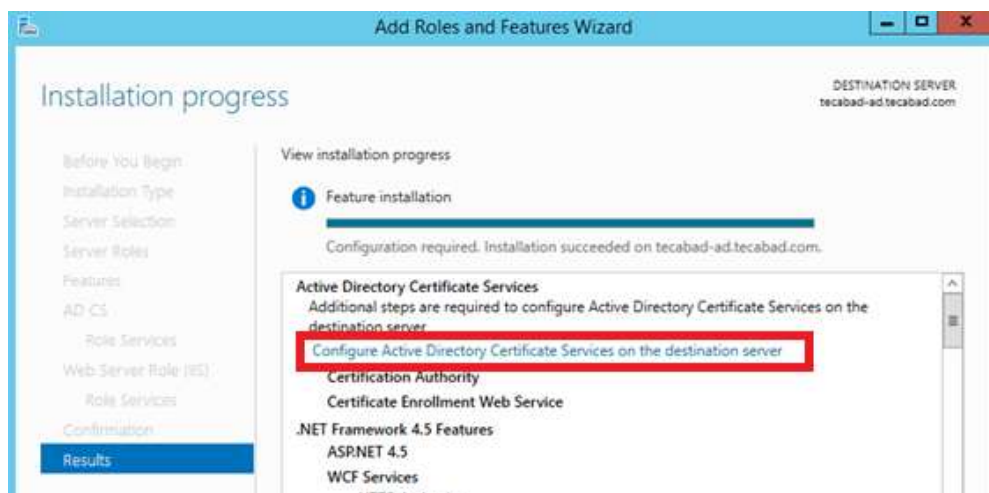


Figura 2.41 Configuración del servicio de entidad certificadora

Aparece el asistente de configuración de los roles de Servidor, donde se escoge el usuario con el cual se configura el servicio, el usuario "adminad" es un usuario que tiene los privilegios de administrador de dominio, ver Figura 2.42, necesarios para esta instalación.

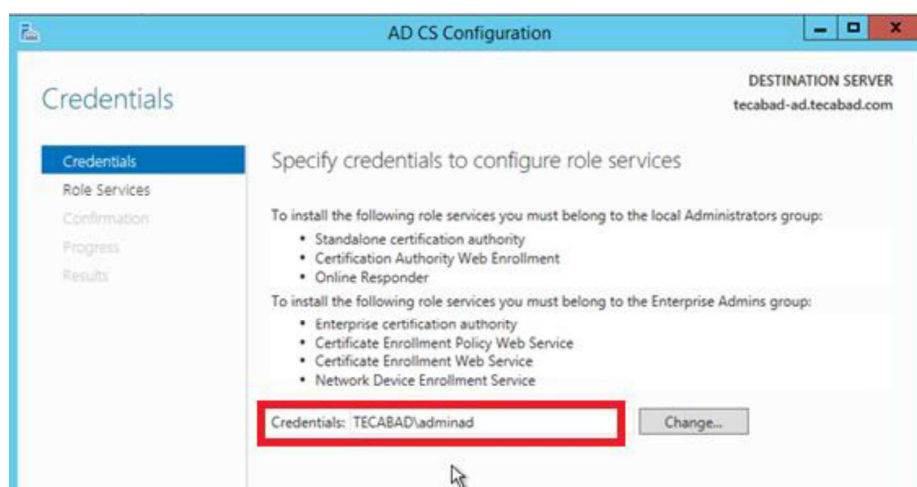


Figura 2.42 Asistente de configuración roles del servidor

Se configura primero la entidad certificadora como se observa en la Figura 2.43, para configurar el servicio de enrolamiento primero se debe configurar la entidad certificadora.

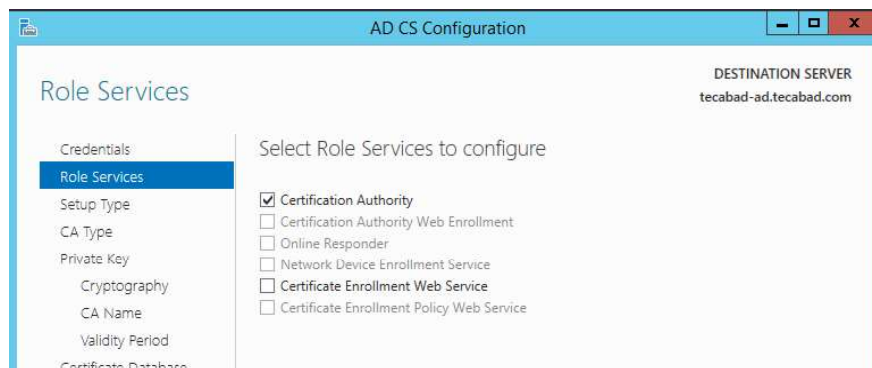


Figura 2.43 Configuración de entidad certificadora

Como el servidor donde se está instalando la entidad certificadora es un controlador de dominio, se especifica que es una entidad certificadora de tipo Enterprise como se observa en la Figura 2.44.

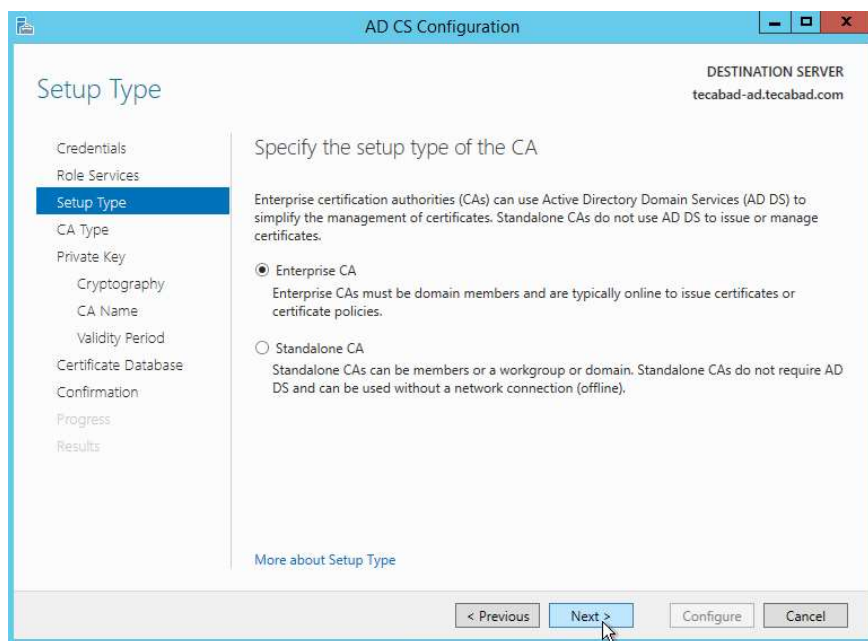


Figura 2.44 Entidad certificadora Enterprise

Se especifica que sea una entidad certificadora raíz, ver Figura 2.45.

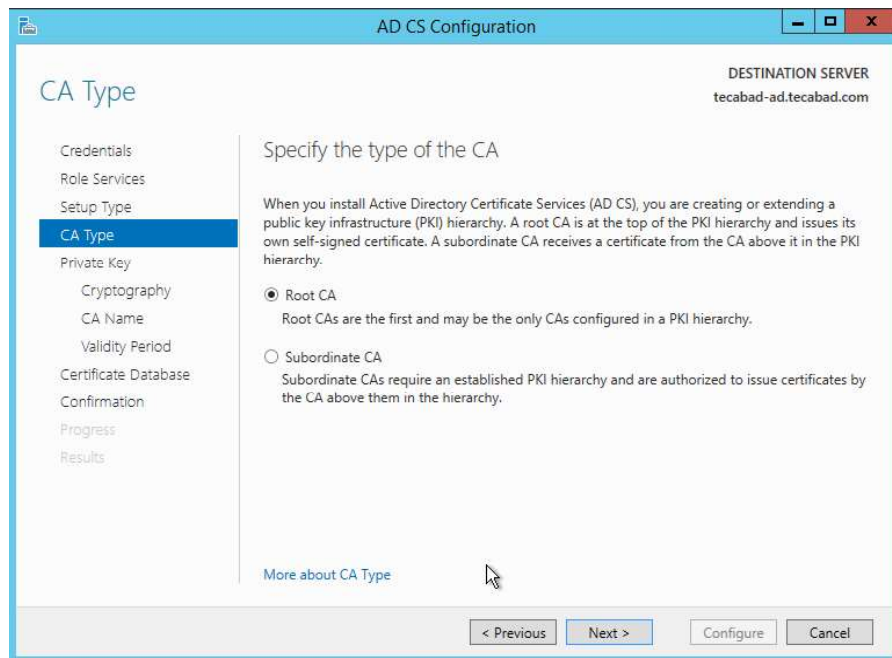


Figura 2.45 Entidad certificadora raíz

Se procede a crear una nueva llave privada para el certificado raíz de la entidad certificadora, ver Figura 2.46.

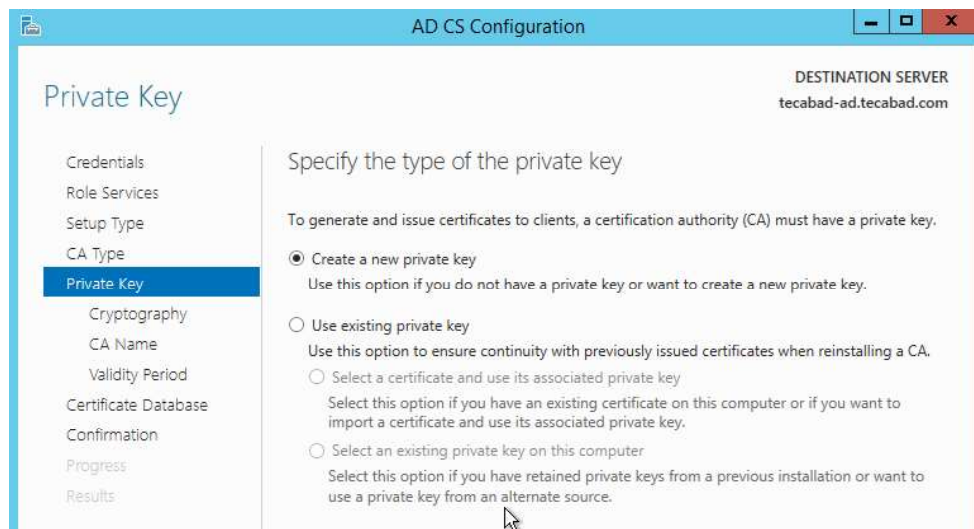


Figura 2.46 Creación de llave privada

Se especifica que el certificado raíz sea creado con un algoritmo SHA256 y una longitud de 2048 Bits, ver Figura 2.47, que son los parámetros por defectos.

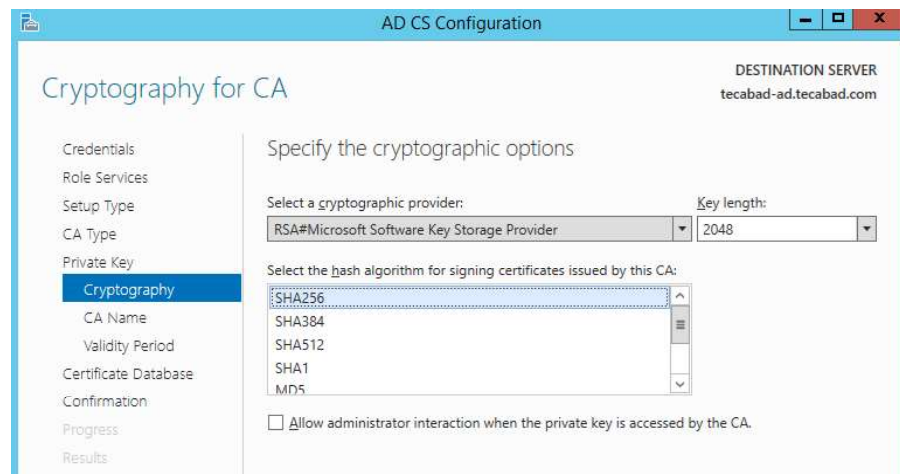


Figura 2.47 Codificación de llave privada

Se especifica el nombre común de la entidad certificadora, ver Figura 2.48.

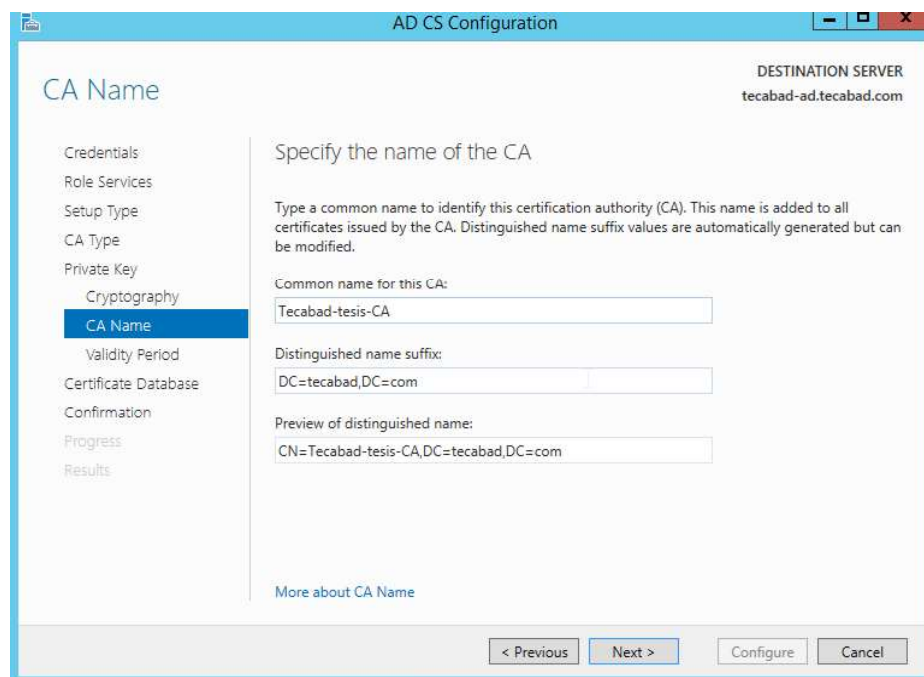


Figura 2.48 Nombre de la entidad certificadora

Se especifica que el certificado raíz con la que se utilizará para firmar los certificados, tenga una vigencia de 5 años, ver Figura 2.49.

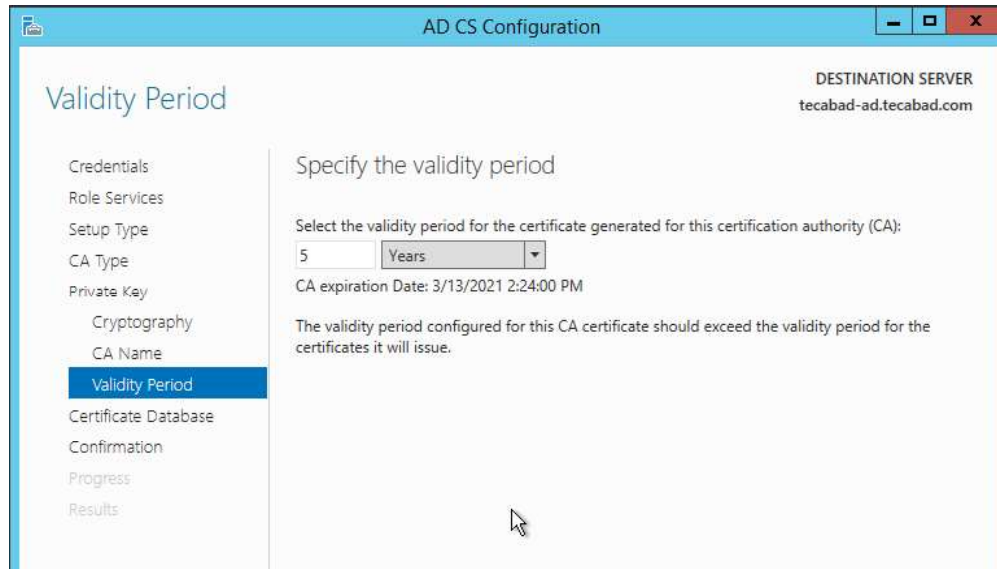


Figura 2.49 Vigencia de clave privada

Se especifica las rutas donde se almacenarán la base de datos y los registros de la entidad certificadora, ver Figura 2.50.

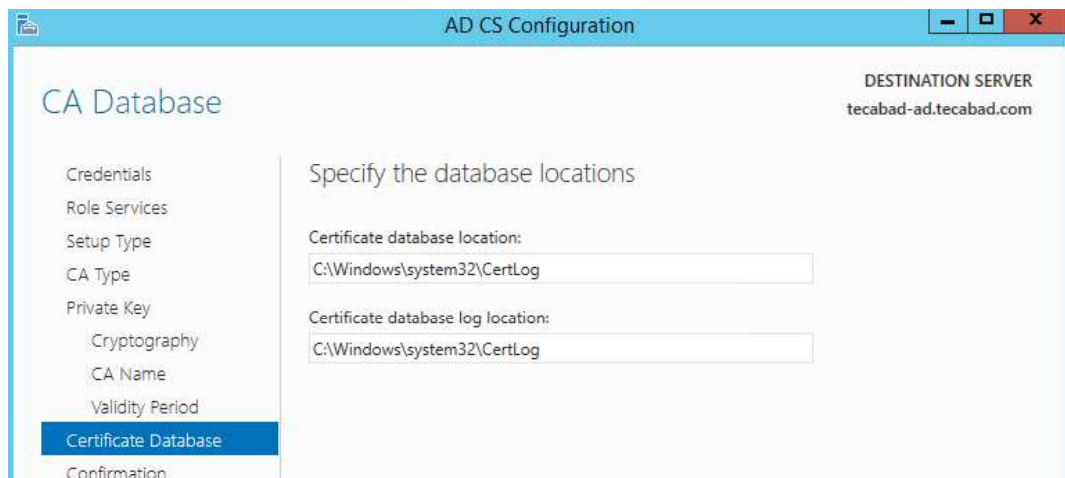


Figura 2.50 Ubicación de la DB de certificados y sus logs

Se revisa las configuraciones y clic en configurar, ver Figura 2.51.

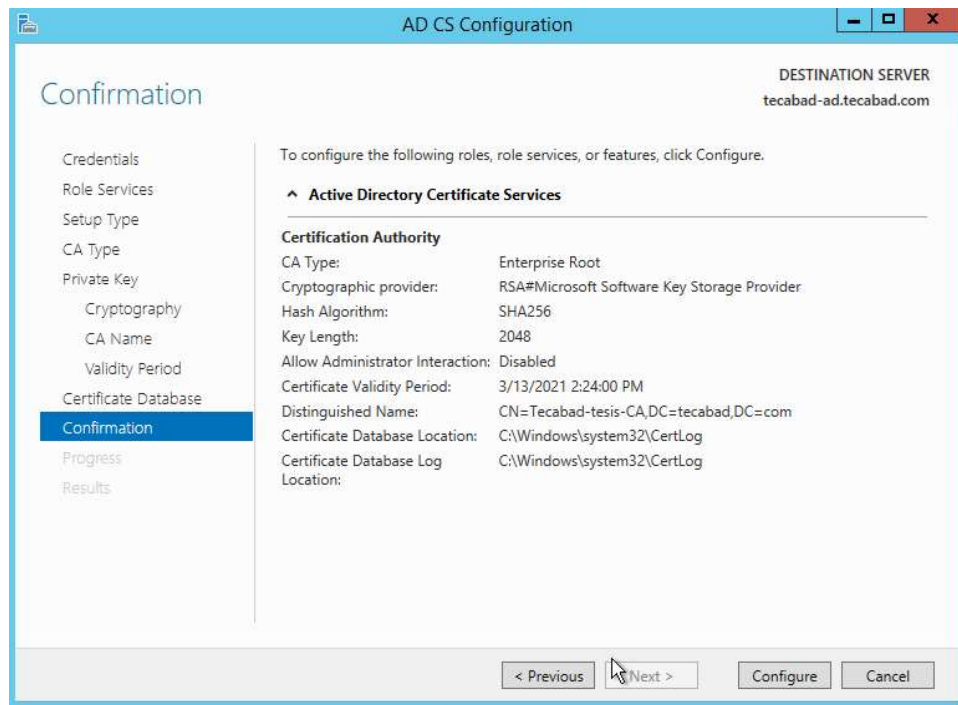


Figura 2.51 Resumen de configuración entidad certificadora

Posterior se procede activar el enrolamiento web de los certificados, esto configurará un sitio web llamado CertSrv, que permitirá firmar los certificados, ver Figura 2.52.

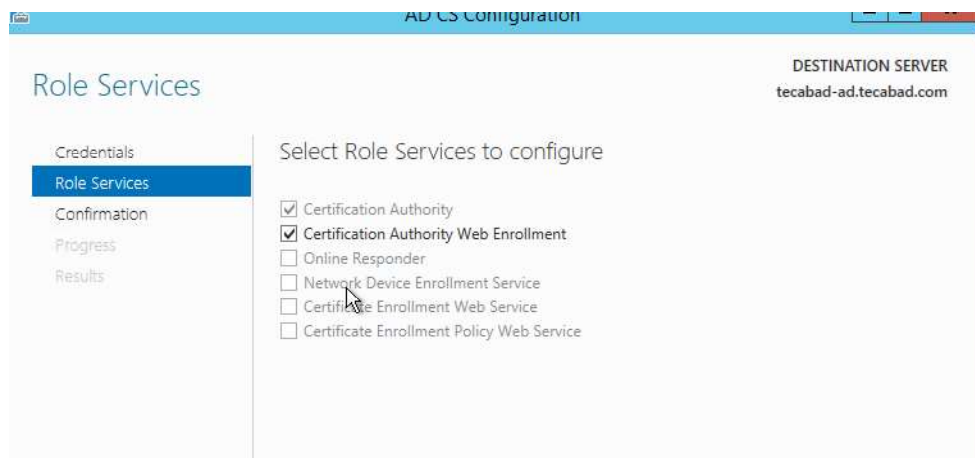


Figura 2.52 Configuración de enrolamiento web

Una vez terminado de configurar el enrolamiento web, ver Figura 2.53, se procede a verificar que en el IIS se creó correctamente servicio web CertSrv, ver Figura 2.54.



Figura 2.53 Activación de enrolamiento web

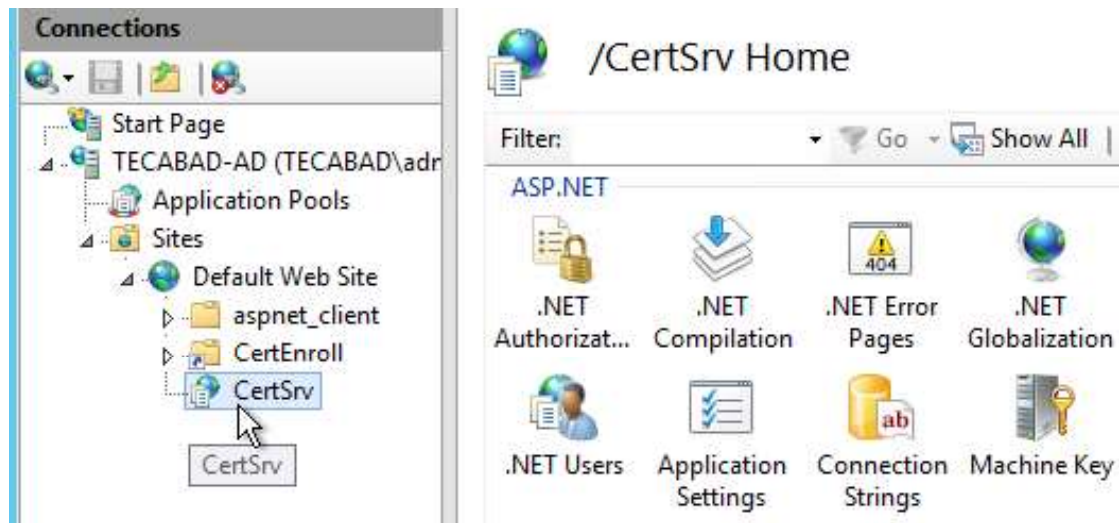


Figura 2.54 Carpeta de IIS CertSrv

Se verifica el servicio de enrolamiento para lo cual se abre <http://localhost/CertSrv> como se observa en la Figura 2.55.

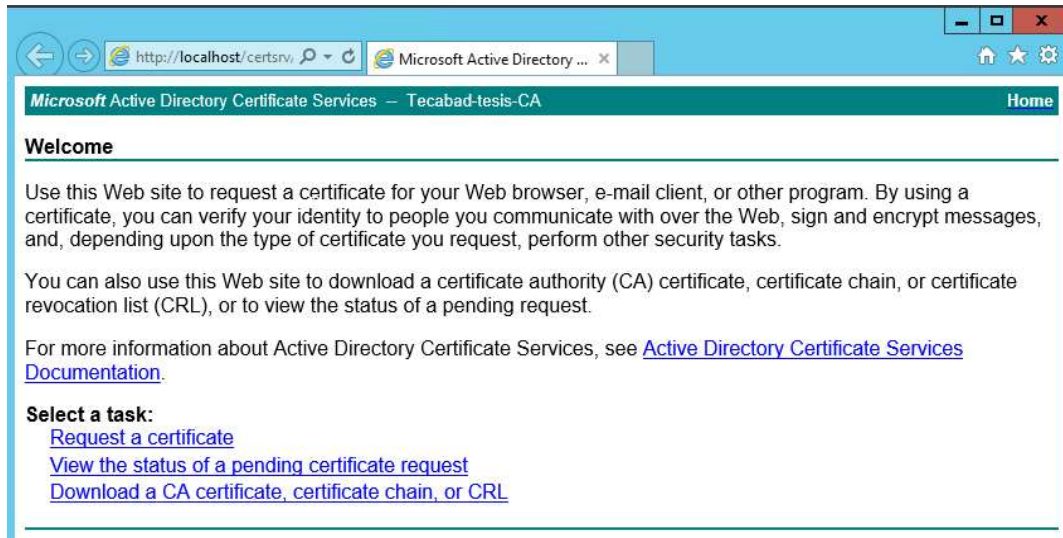


Figura 2.55 Portal de enrolamiento web

Se verifica el almacenaje del certificado en las políticas de AD, para esto se ejecuta el comando, ver Figura 2.56, y aparece el certificado creado, ver Figura 2.57.

```
C:\Users\adminad> certutil -viewstore "ldap:///CN=Tecabad-tesis-CA,CN=Certification Authorities,CN=Public Key Services,CN=Services,CN=Configuration,DC=tecabad,DC=com?cACertificate?base?objectClass=certificationAuthority"
ldap:///CN=Tecabad-tesis-CA,CN=Certification Authorities,CN=Public Key Services,CN=Services,CN=Configuration,DC=tecabad,DC=com?cACertificate?base?objectClass=certificationAuthority
```

Figura 2.56 Certificado raíz en el almacenamiento de certificados



Figura 2.57 Certificado raíz

2.2.8 CONFIGURACIÓN DE UN SERVIDOR DE APLICACIONES WEB

Se debe configurar un servidor en el cual va a ejecutarse las aplicaciones web a desarrollar, para lo cual se procede a levantar al servicio de *Internet Information Service* (IIS).

Desde el asistente para agregar roles y servicios se procede a activar el *roll* de IIS, ver Figura 2.58.

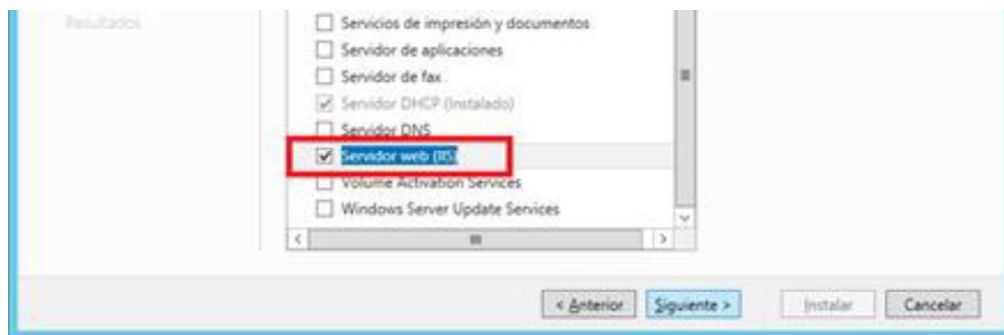


Figura 2.58 Habilitación de IIS

También, se activa la consola de administración de IIS, ver Figura 2.59.

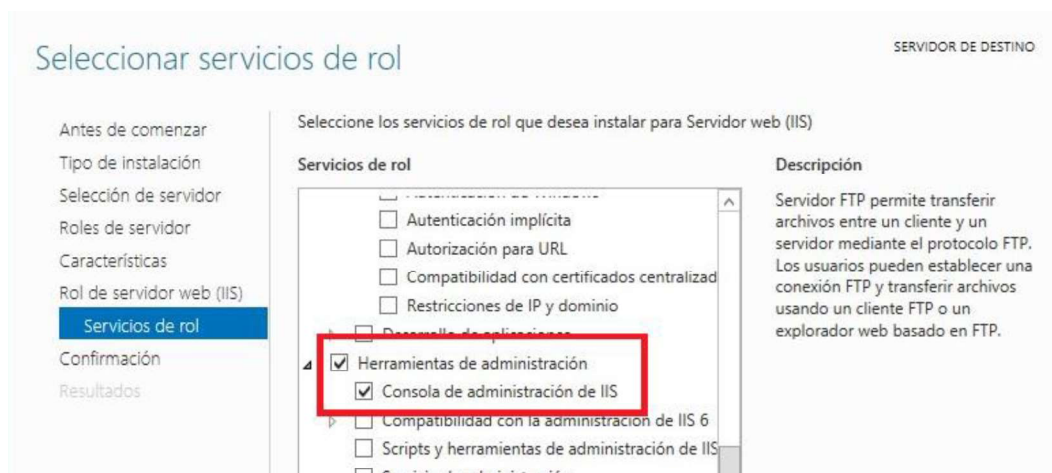


Figura 2.59 Consola de administración de IIS

2.3 IMPLEMENTACIÓN

2.3.1 DESARROLLO DE INTERFACES WEB

A partir de la segunda iteración se procedió con el desarrollo de formularios web que permiten demostrar las pruebas de concepto.

Desde la cuarta iteración, en la codificación de los formularios web, se recurre al uso de divisiones, ver Figura 2.60, que en conjunto con una plantilla creada permite dar personalización en el estilo de fuentes, fondos y tamaño a cada uno de los segmentos de formulario web.

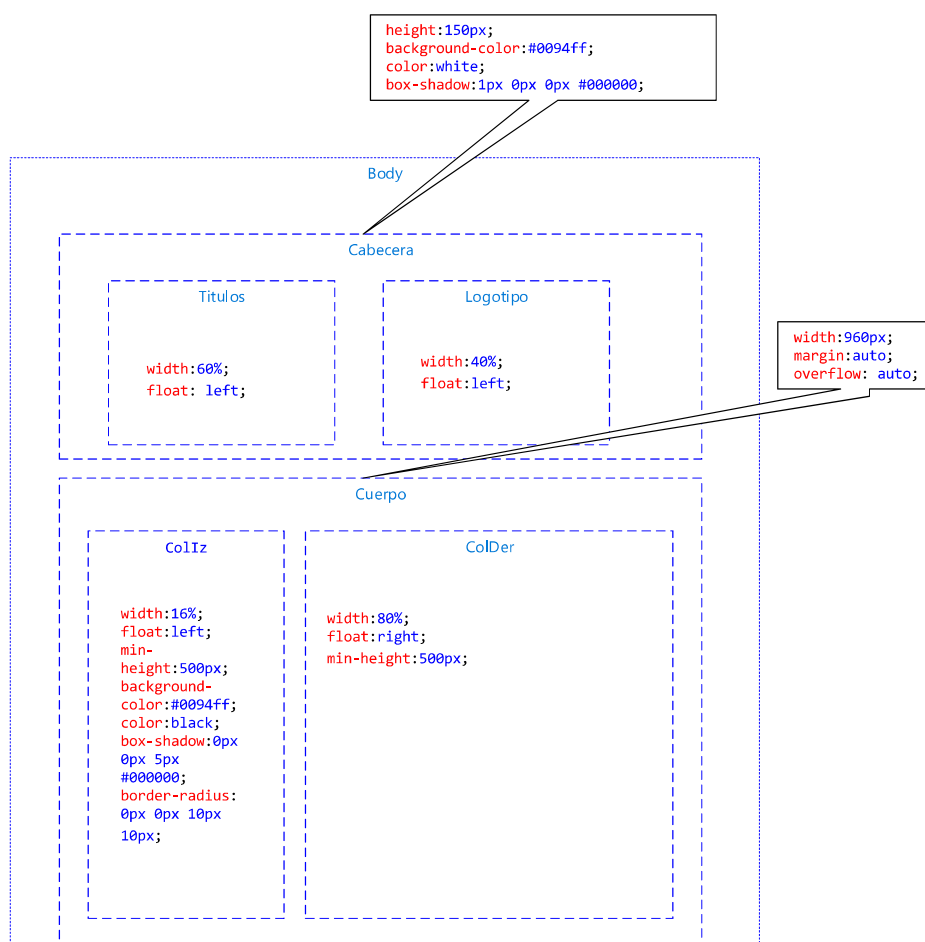


Figura 2.60 División de formulario Web

Se crea una hoja de estilos, ver Figura 2.61, esta generará un archivo de extensión ccs con los estilos que tiene cada división de la página.

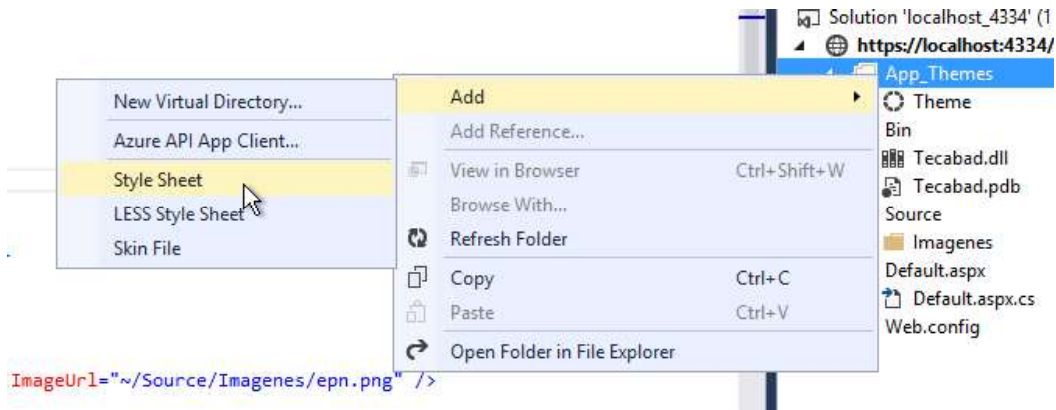


Figura 2.61 Creación de Hoja de Estilo

En el archivo de temas, ver Figura 2.62, se especifica las divisiones a colocaron en todos los formularios web.

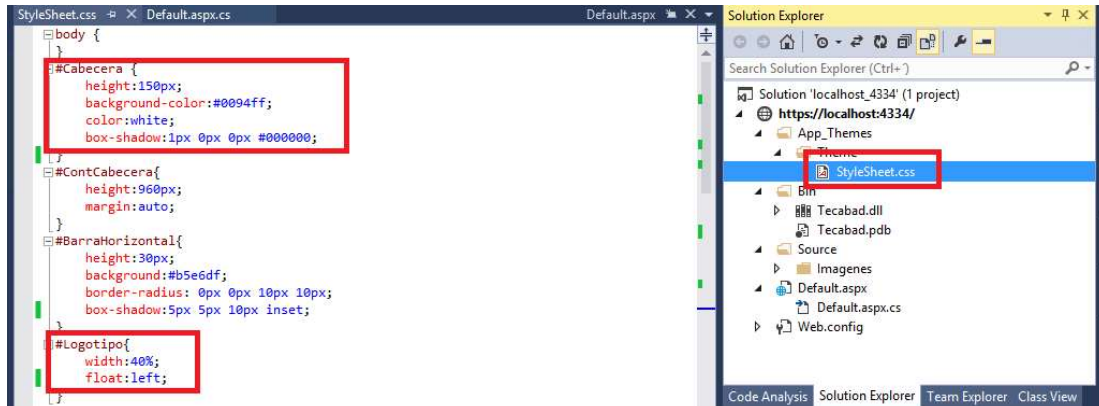


Figura 2.62 Hoja de Estilo CSS

Para relacionar al formulario web, se especifica en la cabecera el archivo de extensión aspx, el nombre del tema que se está utilizando, ver Figura 2.63.

De esta manera cuando se ingresan los elementos a cada división del formulario, estos toman el formato del *template*, ver Figura 2.64.

```

StyleSheet.css  Default.aspx.cs
<%@ Page theme = "Theme" Language="C#" AutoEventWireup="true" CodeFile="De
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>

```

Figura 2.63 Referencia del estilo de la pagina

Figura 2.64 Vista web del formulario

2.3.2 ITERACIÓN 2

En esta iteración busca el mecanismo por el cual se puede cambiar la contraseña de un usuario del Directorio Activo.

Para poder cambiar la contraseña se hace uso de una librería de .NET llamada SystemDirectoryServices, esta librería permite la comunicación con el Directorio Activo.

Para hacer uso de esta librería se requiere los siguientes parámetros del Directorio Activo.

- Dirección o IP del Servidor de AD:
 - IP: 10.0.0.4.
 - Nombre: Tecabad-ad.
- Ubicación de los objetos del Directorio Activo en formato LDAP.
 - Se especifica desde la raíz del directorio.
 - LDAP://tecabad.com/DC=tecabad,DC=com.
- Credenciales de un Usuario administrador.
 - Usuario: Adminweb.

Se procede a crear un formulario web ASP.NET con C# llamado a RestCon.aspx, ver Figura 2.6, este formulario contiene el desarrollo de la primera prueba de concepto.

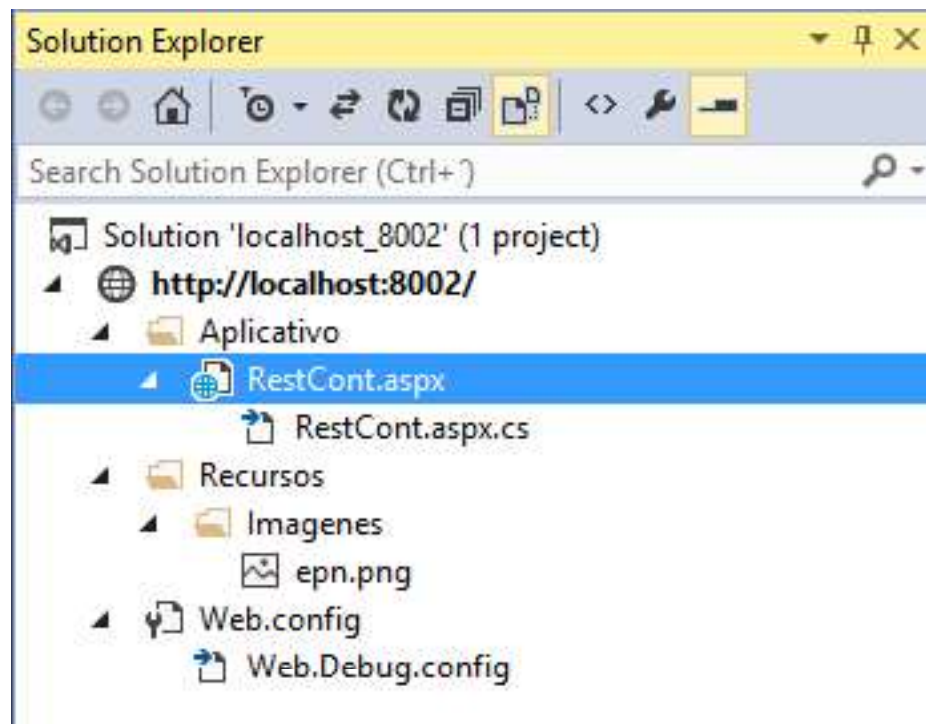


Figura 2.65 Formulario web en ASP.NET

Se coloca en el formulario algunos rótulos, cuadros de texto, botones y personalizaciones como se muestra en la Figura 2.66.



The image shows a web form titled "Prueba de concepto" (Concept Test) with the subtitle "Iteración II". It features the logo of the "ESCUELA POLITÉCNICA NACIONAL" (National Polytechnic School), which includes a shield with a globe, a book, and a gear, and the motto "CERCA DEL FUTURO" (Close to the Future). The form contains the following elements:

- A label "Usuario:" followed by a text input field.
- A "Buscar Usuario" (Search User) button.
- A "Resultados" (Results) label.
- A "Nueva Contraseña" (New Password) label followed by a text input field.
- A "Repetir Contraseña" (Repeat Password) label followed by a text input field.
- A "Cambiar Contraseña" (Change Password) button.

Figura 2.66 Formulario del usuario

Además se generan las siguientes funciones:

2.3.2.1 Conector al directorio Activo

Se crea una función dentro del formulario que permite conectar con el Directorio Activo, esta función especifica la ruta donde se encuentran los objetos del Directorio Activo, ver Código 2.1, se coloca la raíz del directorio para ver todos los objetos del mismo, para esto se especifica el usuario y contraseña de un administrador del dominio.

Esta función devuelve un objeto de tipo DirectoryEntry, que permite crear un conector hacia el Directorio Activo.


```

private DirectoryEntry getDirectoryObject()
{
    DirectoryEntry OdE;
    OdE = new DirectoryEntry("LDAP://tecabad.com/dc=tecabad,dc=com",
"adminweb", "Password");
    return OdE;
}

```

Código 2.1 Conexión con el Directorio Activo

2.3.2.2 Búsqueda en el Directorio Activo

Se crea una función que devuelva la búsqueda de un usuario en el Directorio Activo, ver Código 2.2.

```

private DirectoryEntry GetUser(String UserName) {
    DirectoryEntry de = getDirectoryObject();
    DirectorySearcher desearch = new DirectorySearcher();
    desearch.SearchRoot = de;

    desearch.Filter = "(&(objectClass=user)(SAMAccountName=" + UserName + "))";
    desearch.SearchScope = SearchScope.Subtree;
    SearchResult results = desearch.FindOne();

    if (!(results == null))
    {
        de = new DirectoryEntry(results.Path, "adminweb", "Passw0rd",
AuthenticationTypes.Secure);
        return de;
    }
    else { return null;
    }
}
}

```

Código 2.2 Búsqueda de usuario en el AD

En esta función se crea un objeto DirectorySearch, el cual permite realizar consultas al AD, ha este objeto se filtra otro de tipo usuario con el parámetro SAMAccountName que coincida con el parámetro de entrada de la función llamada nombre, esto devuelve el *path* del usuario en el Directorio Activo. En caso de existir un *path* se instancia un objeto DirectoryEntry con el *path* obtenido como resultado de la función, si no existe el *path* se devuelve un objeto *null*.

2.3.2.3 Búsqueda de objeto Active Directory (AD)

Para la búsqueda del Objeto AD se genera un formulario con un cuadro de texto donde se coloca el alias del usuario como se muestra en la Figura 2.67.

Figura 2.67 Búsqueda del Objeto en el Directorio Activo

En el botón Buscar Usuario se llama a la función GetUser como se observa en el Código 2.3.

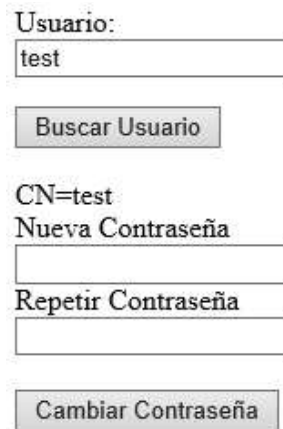
```
protected void Button1_Click(object sender, EventArgs e)
{
    LabelResult.Text = "*****";
    TextCont1.Text = "";
    TextCont2.Text = "";
    DirectoryEntry userAD = GetUser(this.TextadUser.Text);
    if (!(userAD == null))
    {
        LabelResult.Text = userAD.Name;
        LabelCont1.Visible = true;
        LabelCont2.Visible = true;
        TextCont1.Visible = true;
        TextCont2.Visible = true;
        ButtonCambioCont.Visible = true;
    }
    else
    {
        LabelResult.Text = "No encontrado";
        LabelCont1.Visible = false;
        LabelCont2.Visible = false;
        TextCont1.Visible = false;
        TextCont2.Visible = false;
        ButtonCambioCont.Visible = false;
    }
}
```

Código 2.3 Llamado de la función GetUser.

Esta función utiliza como parámetro el valor que tiene el campo `textUser.text`, en el *textbox* se especifica el alias del usuario ejemplo `test`, la función devuelve un objeto de tipo `DirectoryEntry` con la instancia del usuario.

2.3.2.4 Cambio de contraseña

Si la búsqueda del usuario fue exitosa, en el formulario aparecen dos cuadros de texto y botones adicionales, ver Figura 2.68, si no fue exitosa muestra un mensaje de que no encontró el usuario.



Formulario de cambio de contraseña para el usuario "test".

Usuario:
test

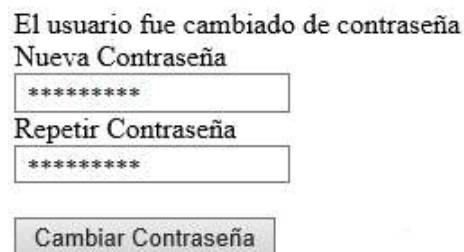
Buscar Usuario

CN=test
Nueva Contraseña
Repetir Contraseña

Cambiar Contraseña

Figura 2.68 Cambio de contraseña

En los *textbox* se coloca la nueva contraseña del usuario y luego clic en cambiar contraseña. Si el cambio de contraseña es exitoso el rótulo cambia de "test" a "el usuario fue cambiado de contraseña".



Formulario de cambio de contraseña después de un éxito exitoso.

El usuario fue cambiado de contraseña

Nueva Contraseña

Repetir Contraseña

Cambiar Contraseña

Figura 2.69 Nueva contraseña

Dentro del botón Cambiar Contraseña se tiene el Código 2.4, el cual realiza una validación de los valores ingresados en los *textbox* coincidan, previo al cambio de contraseña.

```
protected void ButtonCambioCont_Click(object sender, EventArgs e)
{
    try
    {
        if ((TextCont1.Text == TextCont2.Text))
        {
            DirectoryEntry userAD = GetUser(this.TextadUser.Text);
            userAD.Invoke("SetPassword", new object[] { TextCont1.Text });
            LabelResult.Text = "El usuario fue cambiado de contraseña";
        }
        else
        {
            LabelResult.Text = "Las contraseñas no Coinciden";
        }
    }
    catch (Exception excep)
    {
        LabelResult.Text = "Error al cambiar la contraseña: " + excep.Message;
    }
}
}
```

Código 2.4 Validación cambio de Contraseña

Dentro del objeto de DirectoryEntry existe una función llamada *invoke*, ver Código 2.5, que permite llamar a la función para cambiar la contraseña.

```
DirectoryEntry userAD.Invoke("SetPassword", new object[] { TextCont1.Text });
```

Código 2.5 Cambio de contraseña

El objeto llamado userAD, contiene almacenado el objeto del directorio activo y al utilizar el comando *Invoke*, este realiza los cambios en el usuario del AD, se utiliza el comando *Invoke* con los parámetros SetPassword, y la nueva contraseña que se encuentra almacenada en objeto Tesxtcont1.

2.3.3 ITERACIÓN 3 (ANILLOS DE SEGURIDAD)

Se procede a trabajar en los mecanismos de autenticación, como primer mecanismo o anillo de seguridad se utiliza, algo que posee el usuario, previamente se estableció que el usuario tenga acceso a su celular o correo personal, al cual se envía un código de validación.

2.3.3.1 Código Aleatorio

Esta función, ver Código 2.6, genera un número aleatorio de cuatro dígitos que será utilizado para enviar al celular o correo personal del usuario.

```
private int codigo()
{
    Random cod = new Random();

    return cod.Next(1000,9999);
}
```

Código 2.6 Código aleatorio

2.3.3.2 Envío SMS

Para el envío de mensaje de texto se hará uso de un servicio llamado Twilio [33], este servicio permitirá enviar mensajes de texto hacia un celular.

Para poder hacer uso este se crea una cuenta en el portal de Twilio y al momento de contratar el servicio de envío de SMS, se crea una cuenta SID y un *token* de autenticación, ver Figura 2.70, necesarios para la comunicación entre el software y el servidor de SMS de Twilio.

Así mismo se procede a instalar las librerías de Twilio en la herramienta de desarrollo visual estudio, a través de un paquete que se encuentra en nuget.org como se observa en la Figura 2.71.

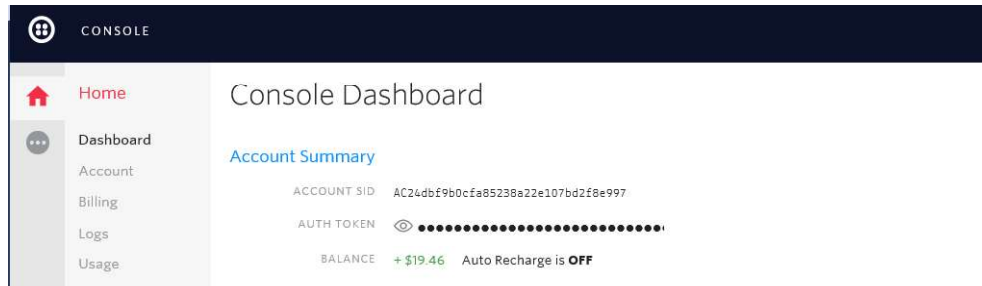


Figura 2.70 Cuenta y Token de Twilio

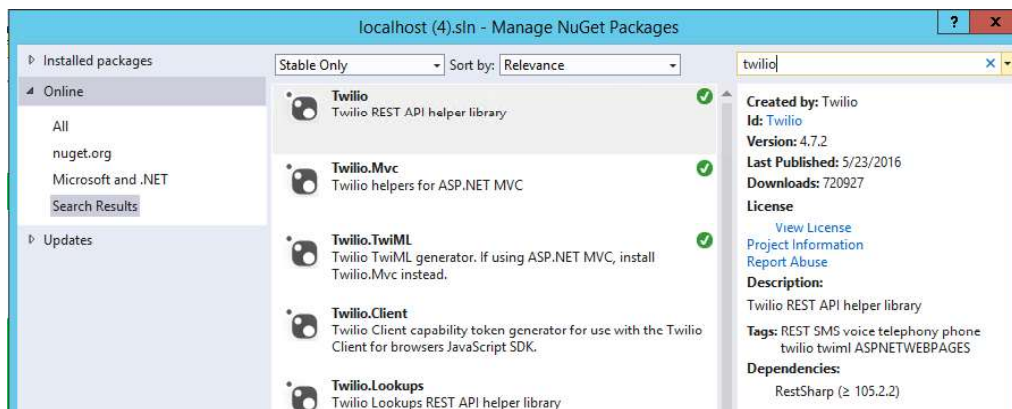


Figura 2.71 Instalación de librería “Twilio” en Visual Studio

Se crea una función, ver Código 2.7, que permita contactar con el servicio de twilio y poder enviar los SMS.

Estas funciones esta dentro de un formulario ASP.NET llamada Envio de SMS, como se observa en la Figura 2.72.

En el botón del formulario llamado SMS Test, ver Código 2.8, se almacenan en variables el número de teléfono y el mensaje que va a enviar.

En la variable que se almacena en el mensaje de texto, incluye un código de cuatro dígitos generada por la función código (), ver Código 2.6.

Para el envío del mensaje de texto, se realiza una llamada a la función SMS (), y se especifica como parámetros el número de teléfono y mensaje de texto, como se observa en la Figura 2.73, el mensaje de texto llegó al celular.

```
using Twilio;
private void sms(String numero, string mensaje)
{
    //var accountSid = "AC2d63382217ad5eba9522909291af5fc7"; //Cuenta de
test
    //var authToken = "1873ec1c80c8ef54c7a5be46612105ba"; // Credenciales de
test
    var accountSid = "ACXXXXXXXXXXXXXXXXXXXXXXXX997"; //Cuenta
    var authToken = "f2XXXXXXXXXXXXXXXXXXXXXXXX18c"; //Token de
autenticación
    var twilio = new TwilioRestClient(accountSid, authToken);
    var message = twilio.SendMessage(
        "+13174973689", //Número proporcionado por la cuenta de twilio
        numero, // Variable con el número de teléfono
        mensaje //Mensaje que contiene el SMS a enviar
    );
    if (message.RestException != null)
    {
        LabelResult.Text = message.RestException.Message;
    }else {
        LabelResult.Text = "mensaje enviado correctamente";
    }
}
}
```

Código 2.7 Código envío de Mensaje texto a celular

Figura 2.72 Formulario ASP.NET envío de SMS

```
protected void ButtonSMS_Click(object sender, EventArgs e)
{
    string num = "+593" +Texttelf.Text;
    string cod = "TECABAD -- Su codigo de verificacion es: " + codigo();
    Labelcod.Text = cod;
    sms(num, cod);
}
```

Código 2.8 Llamada a función SMS (envío de SMS)

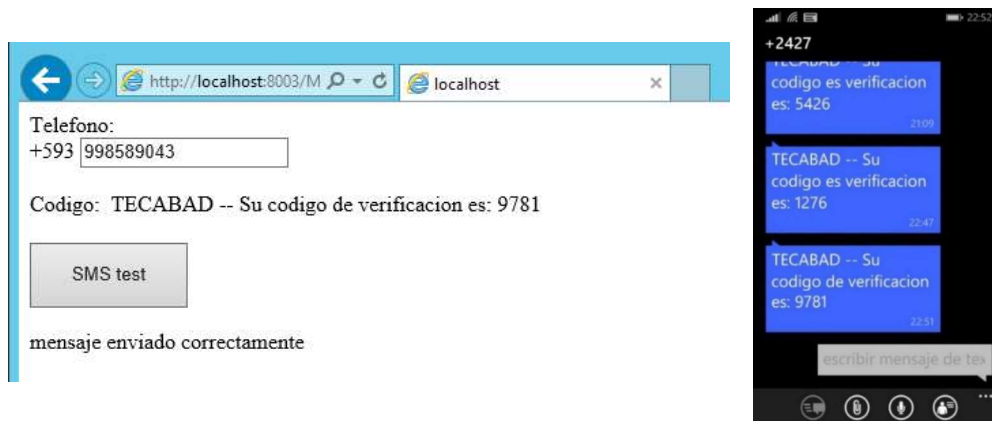


Figura 2.73 Código de validación

2.3.3.3 Envío de correo electrónico

El otro mecanismo de autenticación “de algo que posee un usuario”, es el correo electrónico personal del usuario, este puede ser utilizado en remplazo del envío del mensaje de texto.

Para poder enviar un correo electrónico a un servidor de correo electrónico externo se utiliza un proceso conocido como *Relay* de correo, como el servidor de correo utilizando es Exchange Online, se requiere que en el conector SMTP con autenticación.

Se procede a crear un usuario en Office 365, ver Figura 2.74, para este proceso exclusivamente y se asigna una licencia de Exchange Online para activar el buzón del usuario.

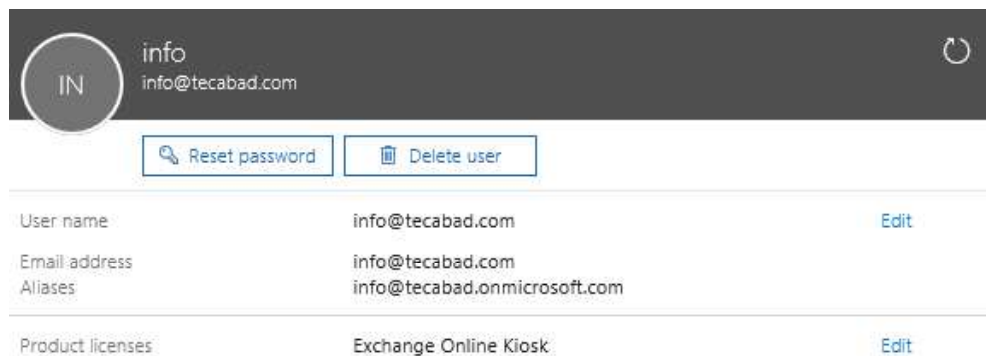


Figura 2.74 Usuario para relay

El usuario de Office 365 a utilizar es `info@tecabad.com`, mediante este usuario autentica el software con el servidor de correo, para permitir el envío de los mismos.

Para poder enviar el correo electrónico en la aplicación se hace uso de las librerías, ver Código 2.9.

```
using System.Net.Mail;
using System.Net;
using System.Net.Mime;
using System.Threading;
using System.ComponentModel;
```

Código 2.9 Librerías para poder enviar correo electrónico

Se genera la siguiente función que permite enviar correo a través de Office 365, ver Código 2.10 y Código 2.11.

```
public void sendmail(String EmailPara)
{
    MailMessage msg = new MailMessage(); // Objeto que representa un correo
    msg.To.Add(new MailAddress(EmailPara));
    msg.From = new MailAddress("info@tecabad.com", "info");
    msg.Subject = "Clave de comprobación";
    msg.Body = "Su clave de comprobación es: " + codigo();
    msg.IsBodyHtml = true;
    //se especifica la conexión con el servidor de correo
    SmtplibClient client = new SmtplibClient();
    client.UseDefaultCredentials = false;
    client.Credentials = new System.Net.NetworkCredential("info@tecabad.com",
```

Código 2.10 Función para envío de correo (Parte 1)

```

"Passw0rd");//usuario para relay
client.Port = 587; // Puerto de SMTP Seguro
client.Host = "smtp.office365.com";
client.DeliveryMethod = SmtpDeliveryMethod.Network;
client.EnableSsl = true;
//se envia el mensaje de correo a través de servidor
try{ client.Send(msg);
    lblresult.Text = "mensaje enviado Correctamente";
    txtto.Text = "Email"; }
catch (Exception ex)
{
    lblresult.Text = ex.ToString();
}}

```

Código 2.11 Función para envío de correo (Parte 2)

Se utiliza el siguiente formulario web, ver Figura 2.75, para probar el envío de correo.

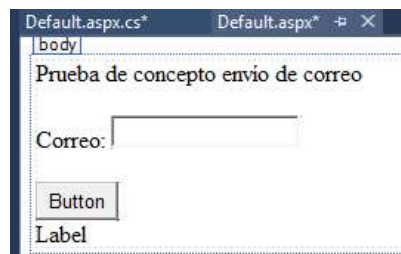


Figura 2.75 Formulario web envío de correo

Como se observa en la Figura 2.76, colocar un correo y se presiona el botón.

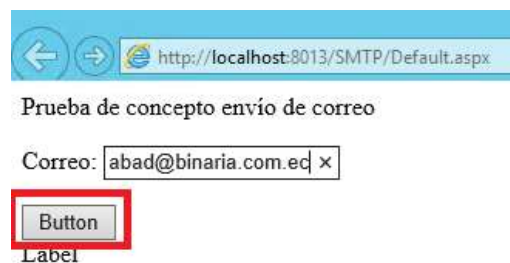


Figura 2.76 Envío de correo

El formulario web devuelve texto "enviado correctamente". Se procede a revisar el envío del correo, como se indica en la Figura 2.77.

Clave de comprobación



Su clave de comprobación es: 6226

Figura 2.77 Clave de comprobación

Se procede con el desarrollo del segundo anillo de seguridad “validación de preguntas”.

2.3.3.4 Validación de preguntas

Una vez que se tiene desarrollado el mecanismo de autenticación de “algo que posee el usuario” se crea el mecanismo de validación “de algo que sabe el usuario”, para esto se propone la validación de ciertas preguntas.

Para la validación de preguntas es necesario almacenar la respuesta, para esto se hará uso de una tabla almacenada en una base de datos, recordando que es una prueba de concepto, tendrá almacenado los valores que se observa en la Figura 2.78, estos se los edita desde la consola de SQL Visual Estudio, posteriormente en la iteración cuarta se crear los mecanismos para que el usuario pueda ingresar por su cuenta los datos para validar las preguntas.

	Id	Pregunta	Respuesta
1	1	¿Cuál es tu color preferido?	Negro
2	2	¿Cuál es tu película preferida?	Advenger
3	3	¿Cuál es tu marca de vehículo preferido?	Renault

Figura 2.78 Tabla preguntas

En esta iteración es necesario establecer la manera de interactuar con la base de datos.

Para esto es necesario realizar una conexión a la base de datos, se utilizará una librería que permite realizar la misma, ver Código 2.12.

```
using System.Data.SqlClient;
```

Código 2.12 Librería para interactuar con la Base de Datos de SQL

La función CargarBase(), ver Código 2.13, que carga los datos de la tabla en un objeto del formulario de tipo *Downlist*.

```
public void CargarBase()
{
    string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=P_Concept_3;Integrated Security=True";
    try
    {
        SqlConnection con = new SqlConnection(datosConexion);
        string sqlquery = "select Pregunta" + ", Id, Respuesta from
preguntas";
        using (con)
        {
            SqlCommand cmd = new SqlCommand(sqlquery, con);
            using (cmd)
            {
                con.Open();
                SqlDataReader rdr = cmd.ExecuteReader();
                DropDownList1.DataSource = rdr;
                DropDownList1.DataValueField = "ID";
                DropDownList1.DataTextField = "Pregunta";
                DropDownList1.DataBind();
                con.Close();
            }
        }
    }
}
```

Código 2.13 Cargar base de datos de SQL en una DownList

Esta función realiza una conexión a la base de datos conocida como P_Concept_3, ver Código 2.14, dentro de la misma se crea un *String* llamado datosConexion, este contiene la información para establecer la comunicación con el servidor de base de datos.

El objeto llamado CON de tipo sqlconnection, crea la conexión en base a los parámetros establecidos en dataConexion.

Para consultar la información de la base de datos, se establece una sentencia de consulta SQL y se la almacena en un *String* llamado SQLQUERY con el siguiente valor "select Pregunta, Id, Respuesta from preguntas" esta sentencia traerá la información que se necesita, ver Figura 2.78.

Para ejecutar la consulta se crea otro objeto de tipo sqlcommand llamado cmd, este comando permite ejecutar la sentencia de SQL a través de la conexión establecida, el resultado es almacenado en el *Datasource* del *dropboxlist*, con esto se obtiene la visualización de la pregunta, ver Figura 2.79.

La función da como valor a mostrar el campo pregunta y como valor de cada fila el ID de la tabla.

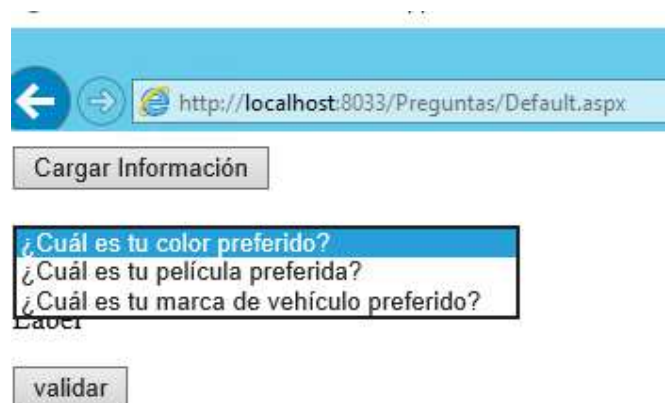


Figura 2.79 Dropboxlist

Se crea una función para validar las respuestas, ver Código 2.14, esta escoge el valor de ID de la tabla, trae el resultado y compara el resultado con el valor del *textBox*.

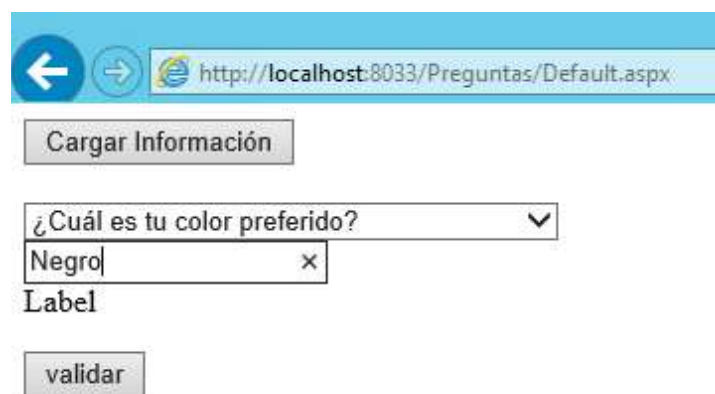
En la Figura 2.80, se realiza la demostración de la validación de respuestas.

```

public void validar()
{
    if (txtvalidar.Text != null)
    {
        string valor = DropDownList1.SelectedValue;
        string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=P_Concept_3;Integrated Security=True";
        SqlConnection con = new SqlConnection(datosConexion);
        string sqlquery = "select Respuesta from preguntas where Id=" +
DropDownList1.SelectedValue;
        using (con)
        {
            SqlCommand cmd = new SqlCommand(sqlquery, con);
            using (cmd)
            {
                con.Open();
                string resultado = cmd.ExecuteScalar().ToString();
                if (txtvalidar.Text == resultado)
                {
                    lblresult.Text = "es igual";
                }
                else
                {
                    lblresult.Text = "No es igual";
                }
                con.Close();
            }
        }
    }
    else
    {
        lblresult.Text = "Por favor colocar un valor";
    }
}

```

Código 2.14 Validar pregunta



¿Cuál es tu color preferido? ▼

Negro x

Label

Figura 2.80 Validación de respuestas

Si se escoge otro color que no sea la respuesta correcta aparecerá el mensaje “No es igual” y si el color es el correcto el mensaje es “es igual” como se observa en la Figura 2.81.

The figure shows two identical form layouts side-by-side. Each layout consists of a 'Cargar Información' button at the top, followed by a dropdown menu with the text '¿Cuál es tu color preferido?'. Below the dropdown is a text input field. In the left form, the dropdown shows 'Negro' and the text below it says 'es igual'. In the right form, the dropdown shows 'Verde' and the text below it says 'No es igual'. At the bottom of each form is a 'validar' button.

Figura 2.81 Ejemplo de validación de respuestas

2.3.4 ITERACIÓN 4 FORMULARIO WEB (VALIDACIÓN DE DATOS)

Esta iteración crea un portal web, mediante el cual el usuario podrá ingresar sus datos de validación de identidad, para lo cual se desarrolla el módulo de ingreso de datos de verificación, diseñado en la sección 2.1.4.5 del presente documento.

Este portal se desarrolla sobre un sitio web alojado en el servidor web llamado tecabad-web, este sitio web atenderá las solicitudes de los clientes utilizando el puerto 4334.

En esta iteración se desarrollará la biblioteca de clases en base al diagrama de clases desarrollada sección 2.1.4.2. A partir desde esta iteración se exportará la biblioteca de clases desarrolla llamada tecabad, a todos los módulos desarrollados posteriormente, de esta manera se pueden usar los objetos de esta biblioteca.

También se utilizará la base de datos llamada tecabad, la cual fue generada en base al diagrama de base de datos desarrollada en la sección 2.1.4.3. Se reutilizará los códigos generados en la segunda y tercera iteración.

2.3.4.1 Biblioteca de Clases

Se procede a desarrollar una biblioteca de clases llamada tecabad en base al diagrama de Clase, esta contiene objetos que representan la lógica del programa como se muestra en la Figura 2.82, para un mayor detalle de los códigos creados en la biblioteca de clases revisar **Error! Reference source not found.**.

La Biblioteca de Clases es compilada para crear un archivo extensión dll, la cual se exporta a los módulos que se desarrollaron, ver Figura 2.83.

Una vez que se exporta la biblioteca llamada tecabad puede ser invocada a través del comando *using* como se observa en la Figura 2.84.

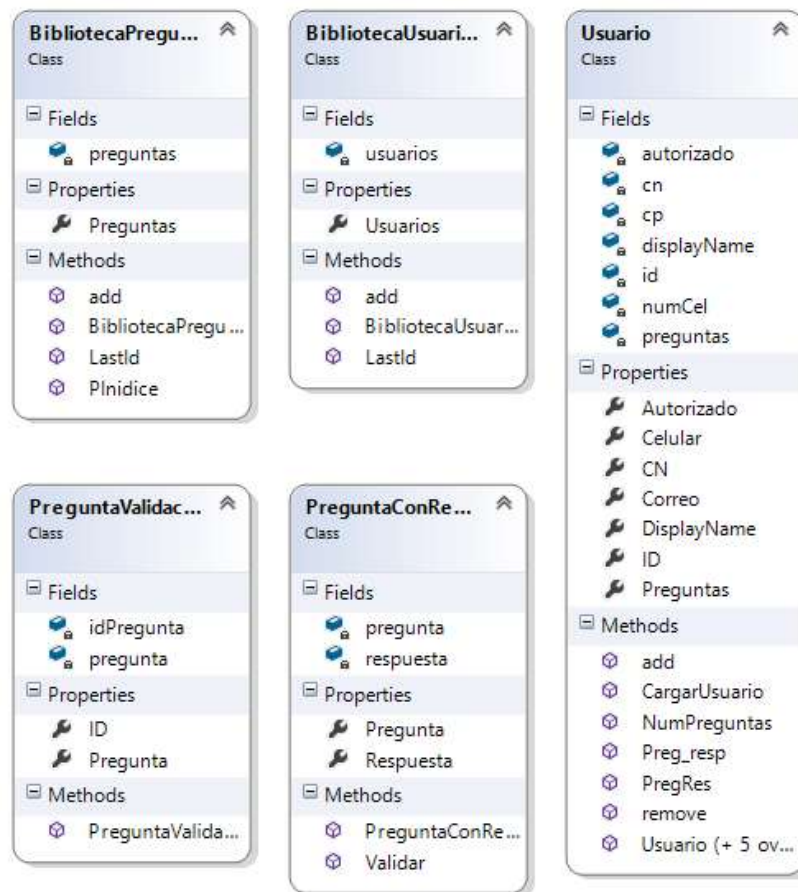


Figura 2.82 Objetos que representan el programa.

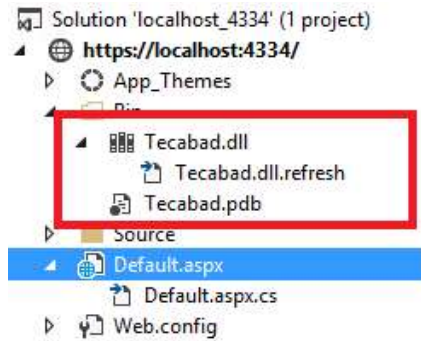


Figura 2.83 Biblioteca de clases

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.DirectoryServices;
using Tecabad;
```

Figura 2.84 Librería Tecabad

2.3.4.2 Asignación del Certificado Digital

Se asigna el certificado digital llamado SISCON SAN previamente instalado en el servidor Web, también se especifica el puerto 4334 para atender las solicitudes de este servicio, como se observa en la Figura 2.85.

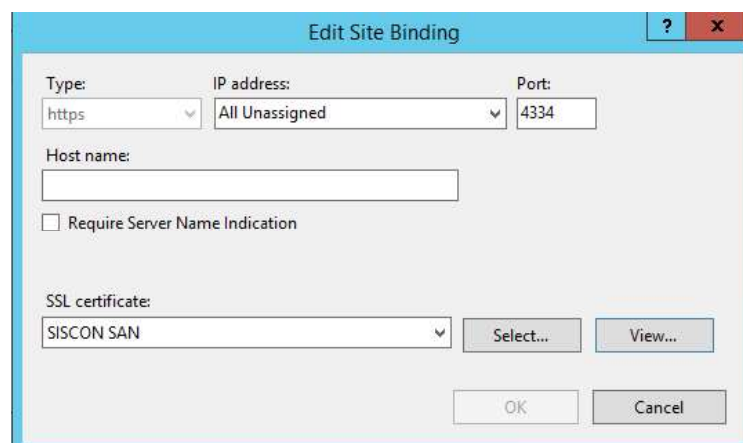


Figura 2.85 Asignación de certificado

2.3.4.3 Autenticación de Windows

Se crea un *website* en el cual se activa la opción de autenticación de Windows y se desactiva la autenticación anónima, como se observa en la Figura 2.86 Autenticación de Windows, de esta manera solo el usuario que tiene la clave del AD puede ingresar a llenar los datos de validación de seguridad.

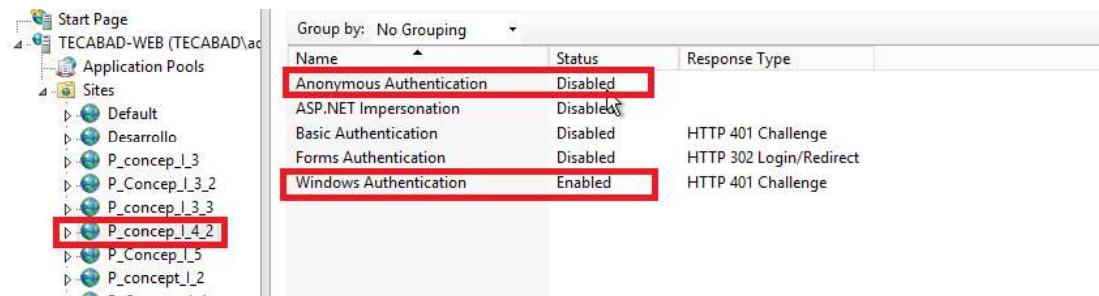


Figura 2.86 Autenticación de Windows

Cuando el usuario quiera acceder al formulario web, este solicitará el ingreso de credenciales, ver Figura 2.87.

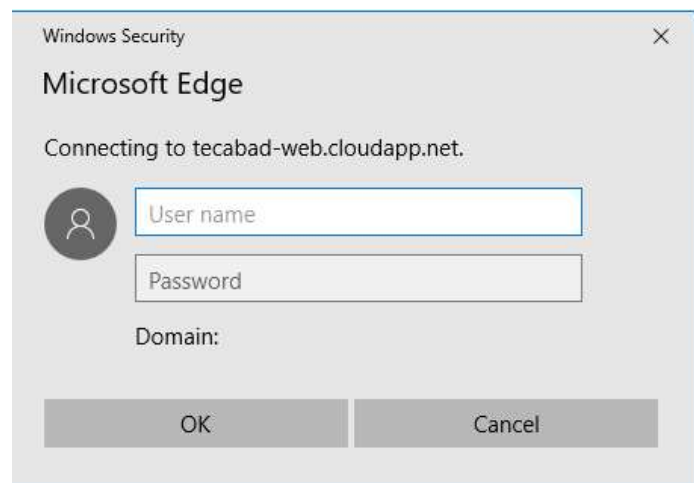


Figura 2.87 Credenciales de Windows

2.3.4.4 Obtención de Datos Usuario

Para el desarrollo de esta iteración se utilizó las librerías, ver Código 2.15.

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Security.Principal;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.DirectoryServices;
using Tecabad;
```

Código 2.15 Librerías iteración 4

Una vez que se procede con la autenticación en el portal, la información del usuario queda almacenada en un objeto de tipo `IPrincipal`, este objeto almacena el dominio y el alias del usuario, como solo se necesita el alias, se separa el alias del dominio mediante el Código 2.16.

```
public String Alias()
{
    IPrincipal myPrincipal = this.User; //nombre del Usuario y dominio
    String Name_ = myPrincipal.Identity.Name; //Nombre del Usuario
    char caracter = '\\'; //Carácter que divide al usuario del Dominio
    int temp = Name_.IndexOf(caracter);
    return Name_.Substring((temp + 1));
}
```

Código 2.16 Obtener el alias del usuario

Se almacena en la variable llamada `name_`, el nombre de cuenta de AD, este presenta el siguiente formato `domino\alias`, como es de interés solo el alias se procede a separar este del registro, para eso se almacena una variable de tipo `Char` con el carácter (`\`), como es un carácter especial se debe de especificar anteponiendo el carácter (`\`), es por eso que se crea la variable y se almacena `\\` con doble *backslash*, el primero es para indicar que es un carácter especial y el otro

es para identificar que el carácter especial que se desea almacenar es el (\) *backslash*.

Una vez que se tiene almacenado el carácter (\), se procede con la función de objeto *string* llamada *IndexOf*, para obtener la posición del carácter (\), se utilizará la función de *Substring* para obtener el texto que se encuentra a partir de la posición del carácter (\) en adelante, de esa manera se obtiene el alias del usuario.

Una vez obtenido el alias del usuario, se consulta el nombre canónico del usuario, para esto se obtiene la ubicación o *path* del usuario en el AD, ver Código 2.17.

```
public String CNAME()
{
    String UserName = Alias();
    DirectoryEntry de = getDirectoryObject();
    DirectorySearcher desearch = new DirectorySearcher();
    desearch.SearchRoot = de;
    desearch.Filter = "(&(objectClass=user)(SAMAccountName="+UserName+")";
    desearch.SearchScope = SearchScope.Subtree;
    SearchResult results = desearch.FindOne();
    return CN(results.Path);
}
```

Código 2.17 Obtención de la ubicación en el Directorio Activo

El objeto de tipo *directoryEntry* almacena la conexión con el Directorio Activo como se revisó en el punto 2.3.2.1 del presente documento, a través de este buscará un objeto dentro del Directorio Activo, obteniendo su ubicación. En un Directorio Activo el atributo que devuelve esta información se llama *SAMAccountName*, con el formato: "CN=test,OU=tecabad,DC=tecabad,DC=com".

El *Canonical Name* es la primera parte del *SAMAccountName*, para obtener esto se especifica la propiedad *Name* al momento de realizar la consulta, ver Código 2.18, y luego se filtra eliminando el texto "CN=". Una vez que obtenido el CN del usuario se crea un objeto de tipo usuario, para eso se utiliza la función llamada "Us", ver Código 2.19.

Nota: Si bien el alias y el CN podrían parecer el mismo valor, este puede variar.

```

public String CN(String Path)
{ //Filtra para solo obtener el CN
    char caracter = '=';
    String Name = new DirectoryEntry(Path, "adminweb", "Passw0rd",
AuthenticationTypes.Secure).Name;
    int temp = Name.IndexOf(caracter);
    return Name.Substring((temp + 1));
}

```

Código 2.18 Canonical Name del usuario

```

public Usuario Us(String CN)
{
//Se crea un objeto de tipo Usuario
    Usuario NewUsuario = new Usuario();
}

```

Código 2.19 Creación de Usuario

Posteriormente se colocan los datos del correo y el celular, ver Código 2.20.

```

//Se crea una conexión a la base de datos y se verifica si existe el Usuario en
la misma.
string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=tecabad;Integrated Security=True";
SqlConnection con = new SqlConnection(datosConexion);
string sqlqueryUser = "Select * from Usuarios Where CN = '" + CN + "'";
using (con)
{
    SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
    using (cmd)
    {
        con.Open();
        SqlDataReader rdr = cmd.ExecuteReader();
        if (rdr.Read()) //Si existe resultado de la BD se procede a extraer la
información del usuario
        {
            NewUsuario.CN = rdr.GetString(rdr.GetOrdinal("CN"));
            NewUsuario.ID = rdr.GetInt32(rdr.GetOrdinal("IDUser"));
            NewUsuario.DisplayName = CNAME();

            Try{ // Se consulta el número de teléfono en caso de existir
NewUsuario.Celular = rdr.GetString(rdr.GetOrdinal("NumeroCelular"));
            } catch {}
            try { //Se consulta el correo electrónico en caso de existir
NewUsuario.Correo = rdr.GetString(rdr.GetOrdinal("CorreoPersonal"));
            } catch { }
        }
        con.Close();
    }
}
}

```

Código 2.20 Carga datos de correo y celular de la Base de Datos

También se cargan las preguntas, ver Código 2.21, para eso se genera una conexión a la base de datos y se consulta las preguntas y respuestas, estas son almacenadas en el objeto usuario mediante una función llamada *add*.

```

SqlConnection con2 = new SqlConnection(datosConexion);

    using (con2)

// Se consulta las preguntas del Usuario

    {

//Se consulta las preguntas del Usuario realizando un INNER entre las bases de
datos

        string sqlqueryPreguntas = "select * FROM UsuarioPreguntas as R
INNER JOIN Preguntas as P ON R.IdPregunta=P.IdPreguntas INNER JOIN Usuarios as U
ON U.IdUser = R.IdUser where U.CN = '" + CN + "'";

        SqlCommand cmd2 = new SqlCommand(sqlqueryPreguntas, con2);

        using (cmd2)
        {
            con2.Open();

            SqlDataReader rdr2 = cmd2.ExecuteReader();

            while (rdr2.Read()) // Se actualizan todas las preguntas
            {
                NewUsuario.Preguntas.Add(new PreguntaConRespuesta(new
                PreguntaValidacion(rdr2.GetInt32(rdr2.GetOrdinal("IdPregunta"
                )))
                , rdr2.GetString(rdr2.GetOrdinal("Pregunta")))
                , rdr2.GetString(rdr2.GetOrdinal("Respuesta"))));
            }

            con2.Close();

        }

    }

    return NewUsuario;

}

```

Código 2.21 Cargar preguntas de la base de datos

Nota: El objeto usuario fue creado en la Biblioteca de Clases llamada Tecabad para mayor referencia revisar **Error! Reference source not found..**

2.3.4.5 Consultar preguntas Existentes

Se crea un Objeto de tipo Biblioteca de Preguntas, ver Código 2.22, este objeto almacena todas las preguntas que se encuentra en la tabla llamada Preguntas almacenada en la Base de Datos.

```

public BibliotecaPreguntas BP()
{
    //Creación del Objeto Biblioteca de preguntas
    BibliotecaPreguntas BiPr = new BibliotecaPreguntas();
    string datosConexion = @"Data Source=tecabad-web\squlexpress;Initial
Catalog=tecabad;Integrated Security=True";

    SqlConnection con = new SqlConnection(datosConexion);

    //Consulta de las preguntas a la tabla de preguntas

    string sqlqueryUser = "select * from Preguntas";

    using (con)
    {
        SqlCommand cmd = new SqlCommand(sqlqueryUser, con);

        using (cmd)
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                BiPr.Preguntas.Add(new
                PreguntaValidacion(rdr.GetInt32(rdr.GetOrdinal("IdPreguntas"))
                , rdr.GetString(rdr.GetOrdinal("Pregunta"))));
            }
            //Carga las preguntas al objeto biblioteca de preguntas
            con.Close();
        }
        return BiPr;
    }
}

```

Código 2.22 Biblioteca de preguntas existentes

2.3.4.6 Preguntas adicionales para el Usuario

Se debe poder diferenciar las preguntas ya seleccionadas por el usuario de las preguntas que le restan por seleccionar, para eso se crea una función llamada

VerPreguntasDisponibles, ver Código 2.23, que permite tener un objeto de tipo BibliotecaPreguntas, el cual contendrá las preguntas que el usuario no ha seleccionado todavía.

```
public BibliotecaPreguntas VerPreguntasDisponibles(BibliotecaPreguntas BP,
Usuario U)
{
// Función que devuelve las preguntas que le restan por seleccionar al usuario
BibliotecaPreguntas temp = new BibliotecaPreguntas();
// Se ingresan dos parámetros una biblioteca de pregunta con todas las preguntas
y un usuario donde se consultará las preguntas que ya tiene seleccionada.
foreach (PreguntaValidacion p in BP.Preguntas)
{ int i = 0;
foreach (PreguntaConRespuesta PR in U.Preguntas) {
//Se compara si la pregunta es igual
if (PR.Pregunta.Pregunta == p.Pregunta) {
i = 1;}}
//Si la pregunta no es igual se almacena variable temp
if (i == 0)
{temp.Preguntas.Add(p);}}
return temp;
// Retorna un objeto de tipo biblioteca de preguntas con las preguntas que no
están seleccionadas por el usuario.
}
```

Código 2.23 Preguntas disponibles

2.3.4.7 Cargar datos de usuario

Una vez que se tiene todos los elementos que permiten interactuar con la base de datos y crear los objetos necesarios, se procede a cargar la información del usuario en el formulario, ver Código 2.24 y Código 2.25.

```
public void Cargar() {
Usuario U = Us(CNAME()); //Crea el usuario en base al usuario
autenticado
BibliotecaPreguntas Pdisp = VerPreguntasDisponibles(BP(), U);
//Carga las preguntas disponible para el usuario

// Se coloca los datos del usuario en los campos correspondientes del Usuario
txtDisplayName.Text = U.DisplayName;
txtCorreo.Text = U.Correo;
txtCelular.Text = U.Celular;
```

Código 2.24 Cargar datos de usuario en formulario (Parte 1)


```

        LboxP.Items.Clear();
        dlistP.Items.Clear();
// Se carga en listbox las preguntas que tiene el usuario ya asignadas
        foreach (PreguntaConRespuesta P in U.Preguntas)
        {
            LboxP.Items.Add(new ListItem(P.Pregunta.Pregunta,
Convert.ToString(P.Pregunta.ID)));
        }
// Se carga las preguntas disponibles en un downlist
        foreach (PreguntaValidacion PV in Pdisp.Preguntas)
        {
            dlistP.Items.Add(new
ListItem(PV.Pregunta,Convert.ToString(PV.ID)));
        }
    }
}

```

Código 2.25 Cargar datos de usuario en formulario (Parte 2)

Esta función es llamada a través de un botón llamado “Cargar Usuario” que aparece después de que el usuario a ingresado al formulario, ver Figura 2.88.

Figura 2.88 Cargar Datos de usuario

El botón BtnCargar llama al a función cargar, ver Código 2.26, y habilita 3 botones que permitirán ver de manera discriminada esta información.

```
protected void btnCargar_Click(object sender, EventArgs e)
{
    Cargar(); //Carga la información en el formulario.
    btnCargar.Visible = false;
    btnInf.Visible = true;
    btnAddPreguntas.Visible = true;
    btnEdPreg.Visible = true;
    btnInf_Click(sender,e);
}
```

Código 2.26 Botón cargar usuario

2.3.4.8 Presentación de los datos

Una vez que se cargan los datos estos se separa en tres partes:

- Información del usuario.
- Editar preguntas.
- Añadir preguntas.

Como se observa en la Figura 2.89.

EPN

Administración de Información

Opciones

- Información de Usuarios
- Editar Respuesta
- Añadir Preguntas

Alias: test

Nombre: test

Correo: dabad@binaria.com.ec

Celular: +593- 998589043

Guardar

Figura 2.89. Presentación de datos

El primer Botón llamado “Información de Usuarios”, permite editar datos de correo y número de celular, ver Figura 2.90.

The screenshot shows a web interface with a light blue header containing the text 'EPN' and 'Administración de Información'. Below the header, there is a sidebar on the left with the title 'Opciones' and three menu items: 'Información de Usuarios', 'Editar Respuesta', and 'Añadir Preguntas'. The 'Información de Usuarios' item is highlighted with a red box. The main content area displays the user's details for 'Alias: test'. The fields are: 'Nombre: test', 'Correo: dabad@binaria.com.ec', and 'Celular: +593- 998589043'. Each of these three fields is highlighted with a red box. At the bottom of the form is a 'Guardar' button.

Figura 2.90 Información de usuario

En el segundo Botón, permite editar las respuestas referentes a las preguntas de validación, ver Figura 2.91.

The screenshot shows the same web interface as Figure 2.90. In the sidebar, the 'Editar Respuesta' menu item is highlighted with a red box. The main content area shows the user's alias 'Alias: test' and a list of 'Preguntas Escogidas'. The first question, '¿CÚAL ES EL NOMBRE DE TU PRIMERA MASCOTA?', is highlighted with a blue box. Below the list, there is an 'Editar:' field containing the text 'PEPA', which is highlighted with a red box. To the right of this field is an 'Actualizar' button, also highlighted with a red box.

Figura 2.91 Editar preguntas

El tercer Botón, permite añadir la pregunta de validación con su respuesta, ver Figura 2.92.

The screenshot shows the EPN (Escuela Politécnica Nacional) 'Administración de Información' interface. The header includes the EPN logo and the text 'Administración de Información'. On the left, there is a sidebar with 'Opciones' and 'Información de Usuarios', with 'Añadir Preguntas' highlighted in a red box. The main content area shows 'Alias: test' and a form with a dropdown menu containing '¿EN QUE PAIS TE GUSTARIA VIVIR?', an empty text input field, and an 'Añadir Pregunta' button, all of which are also highlighted with red boxes.

Figura 2.92 Añadir pregunta

2.3.4.9 Actualización de datos

Uno de los retos de esta iteración, es trabajar con los objetos y su actualización, ya que a diferencia de un aplicativo tradicional las páginas web no trabajan con estado, es decir, si la página web se actualiza, las referencias se pierden y se borra la información que tenían las mismas, para seguir trabajando en el esquema de capas separadas (aplicación y datos), se actualizan constantemente los recursos.

Se desarrolla la siguiente función, ver Código 2.27

```
public void GuardarBase(Usuario ActUsuario)
{
    // se establece una conexión con la base de datos

    string datosConexion = @"Data Source=tecabad-web\squlexpress;Initial
Catalog=tecabad;Integrated Security=True";
    SqlConnection con = new SqlConnection(datosConexion);

    // se actualiza la base de datos para los parámetros del Usuario

    string sqlqueryUser = "UPDATE Usuarios SET CorreoPersonal= '" +
ActUsuario.Correo + "', NumeroCelular='" + ActUsuario.Celular + "' WHERE
IDUser=" + ActUsuario.ID;
    using (con)
    {
        SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
        using (cmd)
        {
            con.Open();
            SqlDataReader ndr = cmd.ExecuteReader();
            con.Close();
        }
    }
}
```

que actualiza los objetos cada vez que la página se refresca.

Una vez actualizado el objeto usuario, se realiza una conexión a la base de datos para modificar los mismos, ver Código 2.28 y Código 2.29.

Si al momento de actualizar la respuesta no consta la pregunta en la base de datos, se añade la misma, ver Código 2.30.

```

public Usuario Actualizar(Usuario UsuarioUpdate)
{
    //Actualiza el registro de correo
    if (!(txtCorreo.Text == null))
    {
        UsuarioUpdate.Correo = txtCorreo.Text;
    }
    if (!(txtCelular.Text == null)) // Actualiza el número de teléfono
    {
        Try {
            // Convierte el número a entero
            int temp = Convert.ToInt32(txtCelular.Text);
            if ((Convert.ToInt32(temp) > 900000000) &&
                ((Convert.ToInt32(temp) < 999999999)))
            {
                UsuarioUpdate.Celular = Convert.ToString(temp);
            }
            else {
                Response.Write("El número de teléfono es incorrecto");}
        } catch {
            Response.Write("El número de teléfono es incorrecto");
        }
    }
    return UsuarioUpdate;
}

```

Código 2.27 Actualización de Datos del usuario

```

public void GuardarBase(Usuario ActUsuario)
{
    // se establece una conexión con la base de datos

    string datosConexion = @"Data Source=tecabad-web\squlexpress;Initial
Catalog=tecabad;Integrated Security=True";
    SqlConnection con = new SqlConnection(datosConexion);

    // se actualiza la base de datos para los parámetros del Usuario

    string sqlqueryUser = "UPDATE Usuarios SET CorreoPersonal= '" +
ActUsuario.Correo + "', NumeroCelular='" + ActUsuario.Celular + "' WHERE
IDUser=" + ActUsuario.ID;
    using (con)
    {
        SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
        using (cmd)
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            con.Close();
        }
    }
}

```

Código 2.28 Actualizar la base de datos (Parte 1)

```

// se procede a actualizar las preguntas del Usuario
foreach (PreguntaConRespuesta PR in ActUsuario.Preguntas)
{
    SqlConnection con2 = new SqlConnection(datosConexion);
    string sqlqueryPregunta = "UPDATE UsuarioPreguntas SET Respuesta = '" +
PR.Respuesta + "' WHERE IDUser='" + ActUsuario.ID + "' and IdPregunta='" +
PR.Pregunta.ID + "'";
    using (con2)
    {
        SqlCommand cmd2 = new SqlCommand(sqlqueryPregunta, con2);
        using (cmd2)
        {
            con2.Open();
            SqlDataReader rdr = cmd2.ExecuteReader();
            if (!(rdr.Read())) {
                AddPregunta(ActUsuario.ID, PR.Pregunta.ID, PR.Respuesta);
            }
            con.Close();}
    }
}
}
}

```

Código 2.29 Actualizar la base de datos (Parte 2)

```

public void AddPregunta(int IdUser,int IdPreg, string Respuesta) {

// Se conecta a la base de datos
    string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=tecabad;Integrated Security=True";

    SqlConnection con3 = new SqlConnection(datosConexion);

// se añade la respuesta a la tabla que contiene las respuestas de los usuarios
con los parámetros (ID de Usuario, el ID de la pregunta, respuesta).
    string sqlqueryUser = "Insert Into
UsuarioPreguntas(IdUser,IdPregunta,Respuesta) values ('" + IdUser + "','" +
IdPreg + "','" + Respuesta + "')";
    using (con3) {
        SqlCommand cmd3 = new SqlCommand(sqlqueryUser, con3);
        using (cmd3) {
            con3.Open();
            try {
                SqlDataReader rdr = cmd3.ExecuteReader();}
            catch {
            }
            con3.Close();
        }
    }
}
}
}

```

Código 2.30 Añadir pregunta de un usuario en la base de datos

Para ver más detalle del código de la iteración 4 revisar **Error! Reference source not found.**

2.3.5 ITERACIÓN 5 FORMULARIO WEB (ADMINISTRACIÓN DE USUARIOS)

En esta iteración se desarrolla el módulo de administración diseñado en la sección 2.1.4.6 del presente documento.

2.3.5.1 Permiso solo a grupo de Seguridad de AD

El portal de administración hace uso de la autenticación de Windows y permite el acceso a los usuarios que pertenecen al grupo de seguridad llamado ADMINSCP, para lograr restringir el acceso del portal web a un determinado grupo, modificar el archivo de configuración especificando el ingreso al grupo "ADMINSCP" y negando al resto como se muestra en la Figura 2.39.

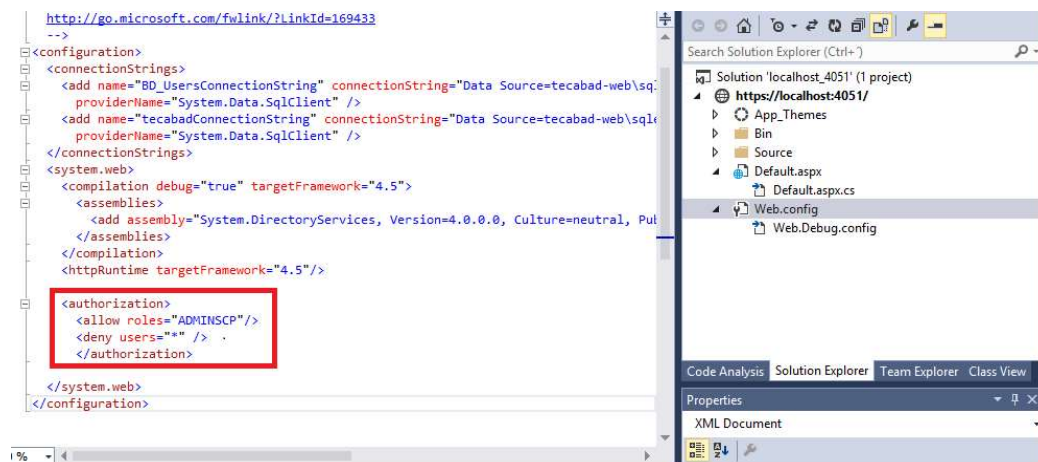


Figura 2.93 Permisos solo a un Grupo de Seguridad

En el Directorio Activo se creó un grupo llamado ADMINSCP como se muestra en la Figura 2.94, otorgando permisos únicamente a los miembros de este grupo, como el usuario llamado Web es el único que se encuentra en este, será el único con permisos para editar los campos de todos los usuarios.

El módulo de administración permite cargar los usuarios del AD en la base de datos, corregir errores de correo y teléfono mal ingresados por el usuario, añadir preguntas de validación y autorizar quien podrá cambiar la contraseña.



Figura 2.94 Grupo permitido para administra sitio Web

2.3.5.2 Actualizar usuarios del AD en la Base de datos

Una de las actividades que realiza el formulario de administración, es poder identificar que usuarios del AD se encuentran ya almacenados en la Base de datos, para eso, utiliza una función llamada Validar, ver Código 2.31, la cual compara si existe un usuario en la base de datos, dentro de esta utiliza un objeto de tipo SqlDataReader la cual utiliza la función *read()* para ejecutar la consulta de un usuario, si encontró el usuario traerá los campos de la base de datos y devolverá un valor de *true* caso contrario devuelve un valor de *false*.

```
public Boolean Validar(String CN)
{
    string datosConexion = @"Data Source=tecabad-web\squlexpress;Initial
Catalog=tecabad;Integrated Security=True";
    SqlConnection con = new SqlConnection(datosConexion);
    string sqlqueryUser = "select * from Usuarios where CN = '" + CN + "'";
    using (con)
    {
        SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
        using (cmd)
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            Boolean temp = rdr.Read();
            con.Close();
            return temp;
        }
    }
}
```

Código 2.31 Validar usuario en la base de datos

Si el usuario del AD no se encuentra en la base de datos, se procede a crearlo en la misma, por defecto se configura que los usuarios nuevos no puedan cambiar la contraseña, ver Código 2.32.

```
public void Insert(String CN)
{
    //Inserta Un usuario Nuevo en la Base de datos
    string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=tecabad;Integrated Security=True";
    SqlConnection con = new SqlConnection(datosConexion);
    string sqlqueryUser = "Insert into Usuarios (IDUser,CN) values ('"+
    BU().LastId() + "',' + CN + "')";
    using (con)
    {
        SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
        using (cmd)
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            con.Close();
        }
    }
}
```

Código 2.32 Insertar nuevo usuario de AD en la Base de Datos

Estas funciones en conjunto son invocadas por la función Cargar, ver Código 2.33, permiten cargar los usuarios nuevos del AD en la Base de datos.

```
public void Cargar(){
    //Actualiza la base de datos con los Usuarios del AD
    DirectoryEntry de = getDirectoryObject();
    DirectorySearcher desearch = new DirectorySearcher(de);
    desearch.PropertiesToLoad.Add("cn");
    desearch.Filter = "(&(objectCategory=user)(objectClass=user)(cn=*))";
    SearchResultCollection queryResults = desearch.FindAll() ;
    foreach (SearchResult Result in queryResults){
        if (!(Validar(CN(Result.Path)))){ //Añadir
            Insert(CN(Result.Path));
        }
    }
}
```

Código 2.33 Cargar usuarios

Al momento de inicializar el portal este actualiza automáticamente los usuarios de AD, pero también puede actualizarse de manera manual al dar clic en el botón actualizar Usuario AD, ver Figura 2.95.

Para actualizar los datos de los usuarios a diferencia de la iteración cuatro, se utiliza un Objeto de tipo `SQLDataSource` la cual se cargará en una vista de grilla.

En el formulario se creó un objeto de tipo `SQLDataSource` y se lo llamó Usuarios, este objeto es parte del formulario por lo que sus datos no se eliminan cuando la página se refresque como se observa en la Figura 2.96.

	IDUser	CN	CorreoPersonal	NumeroCelul
Edit	1	usuario 1	ddeadd@hotmail.com	998589043
Edit	2	usuario 2		
Edit	3	Gabriel Lopez	gabriel.lopez@epn.edu.ec	983582714
Edit	4	Danny Guaman		
Edit	5	test	dabad@binaria.com.ec	998589043
Edit	6	david abad		
Edit	7	test 1		
Edit	8	maria		
Edit	9	Usuario 3		
Edit	10	Web	ddead@hotmail.com	998589043
Edit	11	Vilma Garcia		

Figura 2.95 Actualizar usuarios de AD en DB

La grilla se configura para que contenga los valores de la tabla de usuarios, y se personaliza el campo Autorizado para que este se muestre con *checkbox* como se observa en la Figura 2.96.

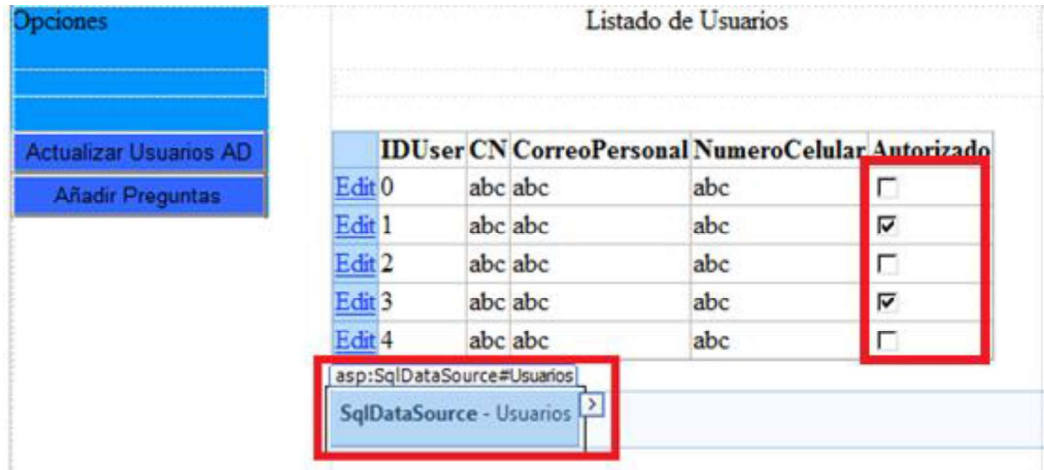


Figura 2.96 SQLDataSource

2.3.5.3 Editar información de usuarios

Para editar un campo de usuario se debe de dar clic en el *link edit* que aparece al lado izquierdo de cada usuario, se habilita los campos de correo y número de teléfono como se observa en la Figura 2.97 y se da clic en *Update*.

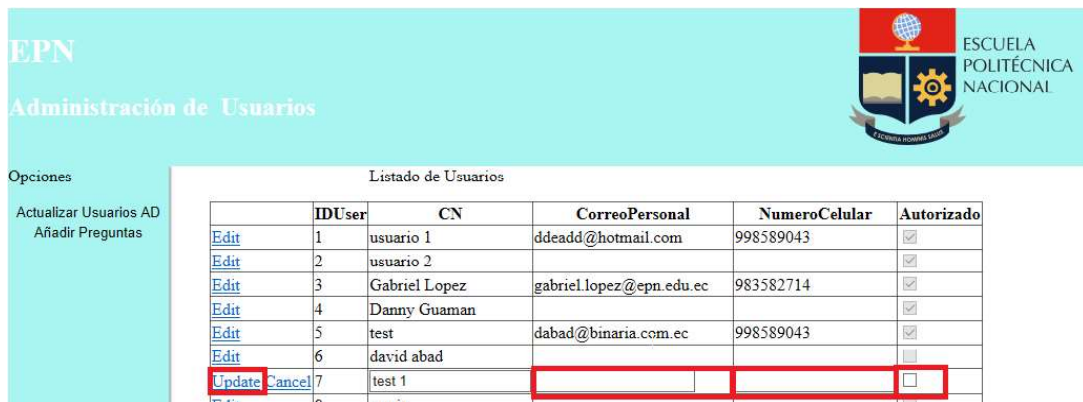


Figura 2.97 Editar campo de usuario

El portal de administración no permite ver las preguntas de los usuarios que han escogido, pero permite añadir más preguntas al sistema para que el usuario pueda escoger alguna, para eso primero se debe poder consultar las preguntas que se encuentran en la tabla llamada preguntas.

2.3.5.4 Añadir preguntas

Para consultar las preguntas, estas son cargadas en un objeto biblioteca de preguntas, ver Código 2.34.

```
public BibliotecaPreguntas BP()
{
    BibliotecaPreguntas BiPr = new BibliotecaPreguntas();
    string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=tecabad;Integrated Security=True";
    SqlConnection con = new SqlConnection(datosConexion);
    string sqlqueryUser = "select * from Preguntas";
    using (con)
    { SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
      using (cmd)
      { con.Open();
        SqlDataReader rdr = cmd.ExecuteReader();
        while (rdr.Read())
        { BiPr.Preguntas.Add(new
PreguntaValidacion(rdr.GetInt32(rdr.GetOrdinal("IdPreguntas"))
, rdr.GetString(rdr.GetOrdinal("Pregunta"))));
        } con.Close();
      } return BiPr;
    }
}
```

Código 2.34 Objeto de Biblioteca de Preguntas

Tener una biblioteca de pregunta, facilita el listar las mismas en un *listbox*, ver Figura 2.98, al ser objetos y no texto es fácil conocer sus atributos.

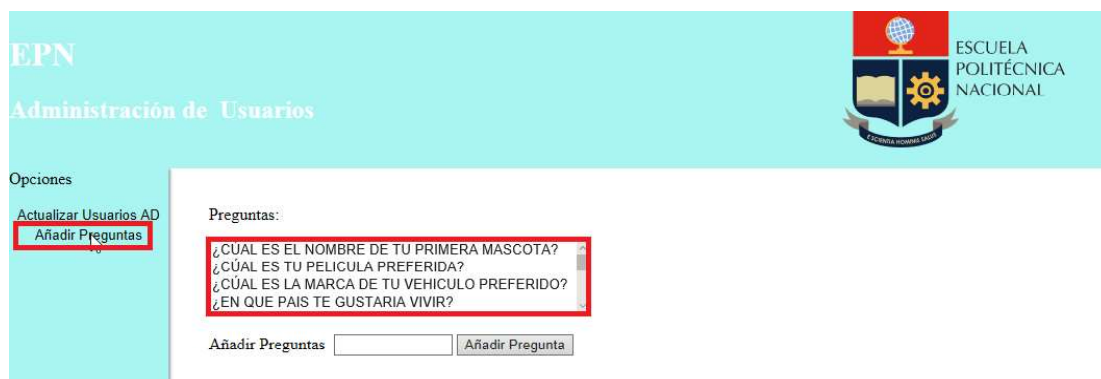


Figura 2.98 Preguntas existentes

Esto permite conocer el ID de la siguiente pregunta al momento de añadir una nueva pregunta. ver Código 2.35.

```

public void addPregunta(String Preg)
{ //Inserta una pregunta nueva en la Base de datos
  string datosConexion = @"Data Source=tecabad-web\squlexpress;Initial
Catalog=tecabad;Integrated Security=True";
  SqlConnection con = new SqlConnection(datosConexion);
  // insterta la pregunta en la tabla preguntas
  string sqlqueryUser = "Insert into Preguntas (IdPreguntas,Pregunta) values
('" + BP().LastId() + "',''" + Preg + "'')";
  using (con)
  {
    SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
    using (cmd)
    {
      con.Open();
      SqlDataReader rdr = cmd.ExecuteReader();
      con.Close();
    }
  }
}

```

Código 2.35 Añadir preguntas a la tabla de Preguntas de la base de datos

Para revisar el código con mayor profundidad ver **Error! Reference source not found..**

2.3.6 ITERACIÓN 6 RELEASE FORMULARIO WEB (CAMBIO DE CONTRASEÑA)

El *release* es el último periodo de desarrollo previo a la puesta de producción de un producto, en este módulo se desarrolló en si el mecanismo para resetear la contraseña que en conjunto con la iteración cuarta y quinta formarán el sistema del prototipo.

Los pasos para el reseteo de la contraseña se representan en el siguiente diagrama de flujo, ver Figura 2.99.

Este módulo abarca las configuraciones y funciones utilizadas en la segunda y tercera iteración, como son generar un código aleatorio, enviar un SMS, enviar un correo, validar las preguntas y cambio de contraseña.

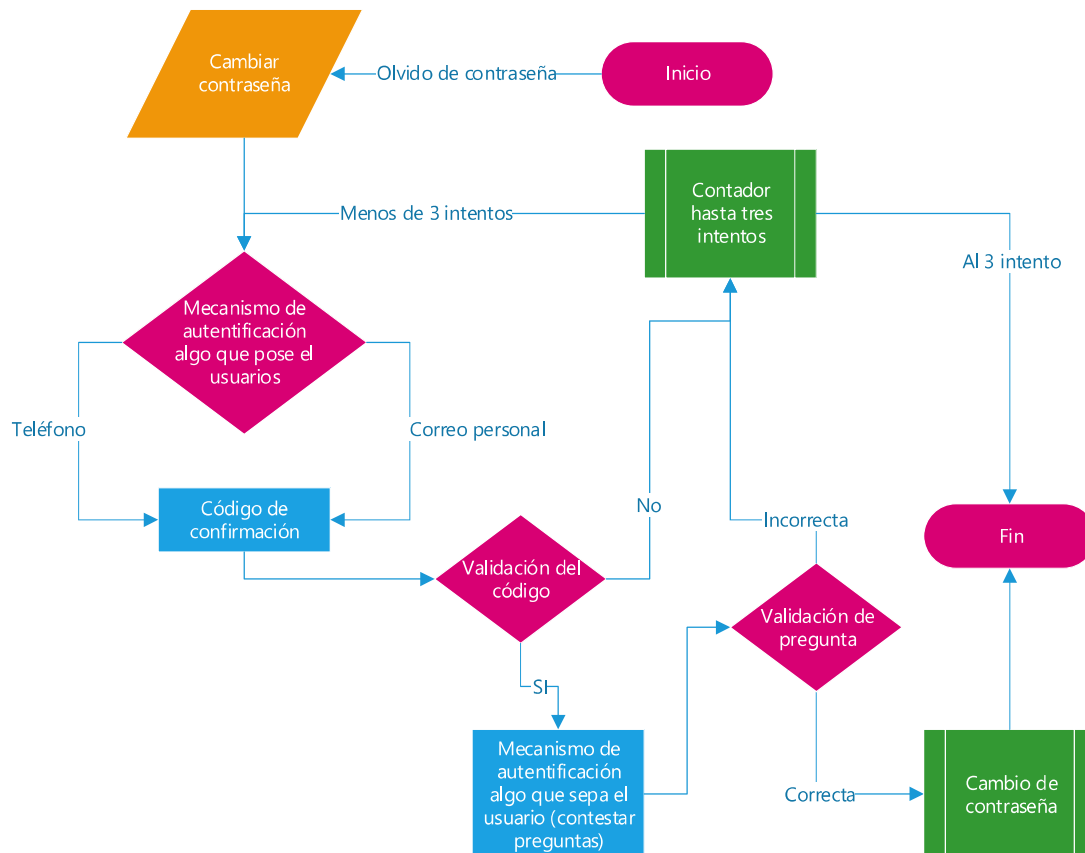


Figura 2.99 Flujo Grama Cambio de contraseña

Uno de los principales retos de este módulo fue cambiar de una ventana a la otra manteniendo un estado y configuraciones, considerando que se está trabajando desde páginas web y al refrescar pierde estos registros, para solventar este inconveniente se usarón variables de sesión las cuales persisten durante la sesión establecida con el servidor web.

Se utiliza las siguientes variables:

- Intentos → Se inicializa en cero, pero incrementa si un proceso es erróneo.

- Cname → Contiene el valor del usuario al que está cambiando la contraseña.
- Estado → Indica en que parte del mecanismo se encuentra el usuario, este presenta tres estados Anillos 1, Anillos 2, SCP, si en cualquier momento el usuario no se encuentra en el estado correspondiente este es dirigido a la página por defecto.

Para poder intercambiar entre los diferentes formularios web, se procede a utilizar la función *Server.Transfer* como se observa en la Figura 2.100, esta función permite redirigir las páginas que se encuentren en el mismo sitio web.

```
        Server.Transfer("/Default.aspx");
    }

    String SesionActiva = (string)(Session["Estado"]);
    if (SesionActiva != "SCP")
    {
        Server.Transfer("/Default.aspx");
    }
}

protected void btnCambiar_Click(object sender, EventArgs e)
{
```

Figura 2.100 Función Server Transfer

Para este módulo se utiliza el puerto 443, y no se utiliza la autenticación con Windows porque el usuario olvidó la clave de esta.

Así mismo como las interacciones en las diferentes ventanas presentan solo una actividad por formulario, en el diseño general se especifica que solo se use la columna derecha del cuerpo del formulario, tal como se observa en la Figura 2.101.

En la pantalla cambio de contraseña especificar el usuario con el formato (Dominio/usuario), ver Figura 2.102.

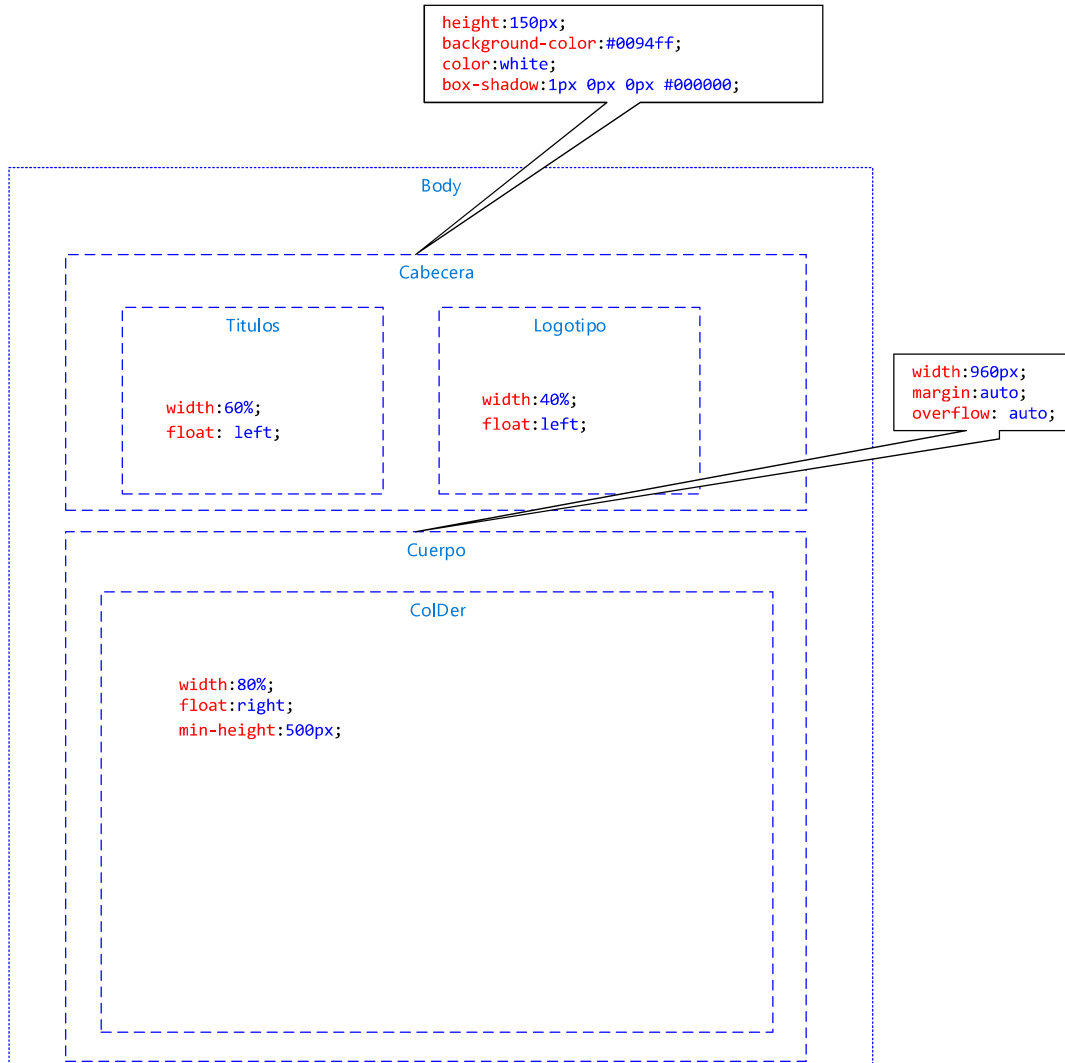


Figura 2.101 Diseño Web cambio de contraseña



Figura 2.102 Pantalla cambio de contraseña

2.3.7 VERIFICACIÓN DE EXISTENCIA DEL USUARIO EN EL AD

Como primera actividad, se debe verificar que el usuario exista en el directorio activo, para eso se utiliza la función Cname, ver Código 2.36, donde si el usuario existe este devuelve un valor con los parámetros del mismo, pero si no existe un usuario devuelve un valor de *null*.

```
private String CNAME()
{
    try
    {
        String UserName = Alias(txtName.Text);
        DirectoryEntry de = getDirectoryObject();
        DirectorySearcher desearch = new DirectorySearcher();
        desearch.SearchRoot = de;
        desearch.Filter = "(&(objectClass=user)(SAMAccountName=" +
UserName + "))";
        desearch.SearchScope = SearchScope.Subtree;
        SearchResult results = desearch.FindOne();
        return CN(results.Path);
    }
    catch
    {
        Response.Write("Usuario Incorrecto");
        return null;
    }
}
```

Código 2.36 Verificación de usuario en el AD

La ejecución de la validación de usuario se la llama desde el botón recuperar contraseña, ver Código 2.37.

```
protected void btnRC_Click(object sender, EventArgs e)
{
    if (txtName != null)
    {
        String CN = CNAME();
        if (CN != null)
        {
            Session["CNAME"] = CN;
            Session["Estado"] = "Anillos1";
            Server.Transfer("Anillo1/Anillos1.aspx");
        }
    }
    Intentos();
}
```

Código 2.37 Botón Recuperar contraseña

Si esta es errónea se llama al proceso de Intentos, ver Código 2.38, que valida el número de veces que trata de consultar el usuario, por cada vez que falle este aumenta el contador si el contador llega a 3 el botón de cambio de contraseña es deshabilitado y el usuario es forzado a cerrar la página web.

```
private void Intentos()
{
    int intento = Convert.ToInt32((String)Session["Intentos"]);
    intento++;
    Session["Intentos"] = Convert.ToString(intento);
    if (intento >= 3) {
        btnRC.Visible = false;
        Response.Write(" ha excedido el número de intentos, cierre el
Explorador y vuelvalo abrirlo");
    }
}
```

Código 2.38 Intentos Fallidos

Si el usuario ingresado existe el sistema pasa al formulario llamado anillo 1, en este formulario el usuario especifica como recibir el código de validación, sea a través de un correo electrónico o un SMS, ver Figura 2.103.



The image shows a web form titled "EPN Cambio de Contraseña". Below the title, there is a text input field containing "test". Underneath, the text "Seleccione el mecanismo de validación" is followed by two radio button options: "SMS" and "Correo". The "Correo" option is selected. Below the radio buttons is a button labeled "Enviar" and an empty text input field.

Figura 2.103 Envío de código de validación

2.3.8 PRIMER ANILLO DE SEGURIDAD

Una vez que se pasa al formulario llamado anillo 1 este debe consultar el usuario del Directorio Activo, para eso hace uso de la función Us que devuelve el objeto Usuario, ver Código 2.39, del cual se extrae la información para la validación.

```
private Usuario Us(String CN)
{
    Usuario NewUsuario = new Usuario();
    //Inserta Un usuario Nuevo en la Base de datos
    string datosConexion = @"Data Source=tecabad-web\sqlexpress;Initial
Catalog=tecabad;Integrated Security=True";
    SqlConnection con = new SqlConnection(datosConexion);
    string sqlqueryUser = "Select * from Usuarios Where CN = '" + CN +
    "'";
    using (con)
    {
        SqlCommand cmd = new SqlCommand(sqlqueryUser, con);
        using (cmd)
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            if (rdr.Read()) {
                NewUsuario.CN = rdr.GetString(rdr.GetOrdinal("CN"));
                NewUsuario.ID = rdr.GetInt32(rdr.GetOrdinal("IDUser"));
                NewUsuario.DisplayName = CN;
                try { NewUsuario.Celular =
rdr.GetString(rdr.GetOrdinal("NumeroCelular")); }
                catch { }
                try { NewUsuario.Correo =
rdr.GetString(rdr.GetOrdinal("CorreoPersonal")); }
                catch { }
                try { Boolean aut =
rdr.GetBoolean(rdr.GetOrdinal("Autorizado"));
                NewUsuario.Autorizado = aut; }
                catch { NewUsuario.Autorizado = false; }
            } con.Close(); } }
    SqlConnection con2 = new SqlConnection(datosConexion);
    using (con2)
    {
        string sqlqueryPreguntas = "select * FROM UsuarioPreguntas as R
INNER JOIN Preguntas as P ON R.IdPregunta=P.IdPreguntas INNER JOIN Usuarios
as U ON U.IdUser = R.IdUser where U.CN = '" + CN + "'";
        SqlCommand cmd2 = new SqlCommand(sqlqueryPreguntas, con2);
        using (cmd2)
        {
            con2.Open();
            SqlDataReader rdr2 = cmd2.ExecuteReader();
            while (rdr2.Read())
            {
                NewUsuario.Preguntas.Add(new PreguntaConRespuesta(new
PreguntaValidacion(rdr2.GetInt32(rdr2.GetOrdinal("IdPregunta"))
, rdr2.GetString(rdr2.GetOrdinal("Pregunta")))
, rdr2.GetString(rdr2.GetOrdinal("Respuesta"))));
            }
            con2.Close();
        }
    }
}
```

Código 2.39 Cargar usuario

Una vez que se tiene el objeto usuario, ya se conoce los datos del teléfono y correo del mismo, al cual se envía el código de verificación, tal como se observó en la iteración 3 en la parte de Envío de correo o Envío SMS, dependiendo de lo que seleccionó el usuario, ver Figura 2.104.



The screenshot shows a light blue header with the text "EPN" and "Cambio de Contraseña". Below the header, the word "test" is visible. A red rectangular box highlights a section containing the text "Seleccione el mecanismo de validación". Underneath this text are two radio button options: "SMS" (which is unselected) and "Correo" (which is selected). Below the radio buttons is a button labeled "Enviar".

Figura 2.104 Selección para envío de SMS o Correo

Una vez que se confirma el código de seguridad ver Figura 2.105 se procede con una redirección al formulario llamado Anillo2, como se observa en el siguiente Código 2.40.



The screenshot shows the same light blue header with "EPN" and "Cambio de Contraseña". Below the header, the word "test" is visible. The text "Seleccione el mecanismo de validación" is present, with the "Correo" radio button selected. Below this, there is a button labeled "Enviar" followed by the text "El código de Verificación a sido enviado al siguiente correo: da****@****ec". Underneath this text is a text input field containing the number "7692" and a small "x" icon to its right. At the bottom of the highlighted area is a button labeled "Comprobar".

Figura 2.105 Código de verificación

```
protected void btnComprobar_Click(object sender, EventArgs e)
{
    Intentos();
    string comparar= (string)(Session["CODE"]);
    if (txtCod.Text == comparar)
    {
        Session["Estado"] = "Anillo2";
        Server.Transfer("/Anillo2/Anillo2.aspx");
    }
    else
    {
        Response.Write(" Código Erroneo ");
    }
}
```

Código 2.40 Verifica el código enviado

2.3.9 SEGUNDO ANILLO DE SEGURIDAD

El segundo anillo de seguridad permite al usuario seleccionar una pregunta previamente registrada por el usuario, para eso el usuario debe dar clic en el botón escoger pregunta como se muestra en la Figura 2.106



Figura 2.106 Escoger pregunta de validación

Se escoge una de las preguntas de validación como se observa en la Figura 2.107

Una vez seleccionada la pregunta se procede a contestar, ver Figura 2.108, y si esta es correcta permite el acceso al cambio de contraseña.

EPN

Cambio de Contraseña

test

Conteste la siguiente pregunta:

- ¿CUAL ES EL NOMBRE DE TU PRIMERA MASCOTA?
- ¿CUAL ES TU PELICULA PREFERIDA?
- ¿CUAL ES LA MARCA DE TU VEHICULO PREFERIDO?
- ¿CUAL ES TU COLOR PREFERIDO?
- ¿CUAL ES TU CIUDAD PREFERIDA?

Figura 2.107 Seleccionar pregunta

EPN

Cambio de Contraseña

test

Conteste la siguiente pregunta:

¿CUAL ES EL NOMBRE DE TU PRIMERA MASCOTA? ▾

Respuesta

Figura 2.108 Contestar la pregunta

2.3.10 FORMULARIO SCP

Una vez que se contesta la pregunta de validación, se pasa al formulario scp en este se procede a cambiar la contraseña, como se observa en la Figura 2.109, para esto utiliza el Código 2.41.

test

Nueva Contraseña:

Repetir Contraseña:

Figura 2.109 Cambio de contraseña

```

protected void btnCambiar_Click(object sender, EventArgs e)
{
    try
    {
        if (txtPass.Text == txtCpass.Text)
        {
            string UsuarioActivo = (string)(Session["ALIAS"]);
            DirectoryEntry userAD = GetUser(UsuarioActivo);
            userAD.Invoke("SetPassword", new object[] { txtPass.Text });
            Response.Write("El usuario fue cambiado de contraseña");
            Server.Transfer("/Default.aspx");
        }
        else
        {
            Response.Write("La Contraseña no coincide");
        }
    }
    Catch {}
}

```

Código 2.41 Cambio de contraseña

Donde se carga el usuario del AD, posteriormente procede a cambiar la contraseña del mismo, para una mayor referencia del código, ver **Error! Reference source not found..**

Capítulo 3

3 PRUEBAS DE FUNCIONAMIENTO

En el ambiente desarrollado, se realizar pruebas del funcionamiento a los módulos de Administración, Ingreso de Datos de Verificación y Cambio de Contraseña.

3.1 PRUEBAS DE FUNCIONAMIENTO MÓDULO ADMINISTRACIÓN

3.1.1 ACTUALIZACIÓN DE USUARIOS DE AD

El módulo de Administración permite cargar los usuarios del AD en la Base de Datos, para efectuar la prueba se procede con la creación de un usuario en el AD ver Figura 3.1.

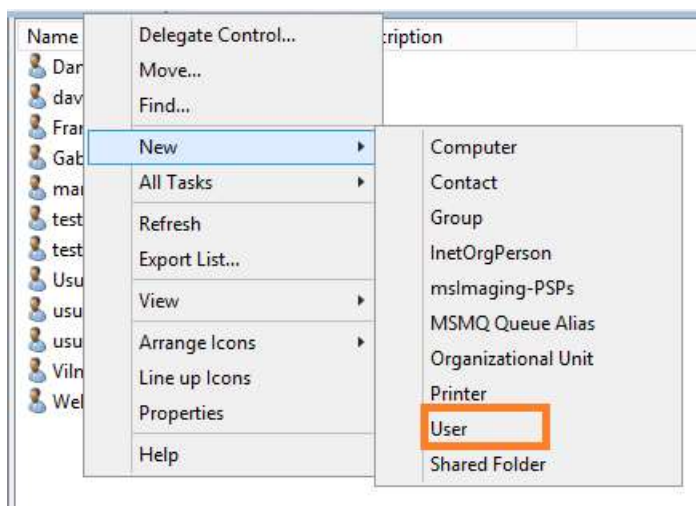


Figura 3.1 Creación de Usuario en el AD

Se crea el usuario llamado Prueba Funcionamiento como se observa en la Figura 3.2.

Se abre el portal de administración, ver Figura 3.3, y dar clic en actualizar usuarios de AD.

New Object - User

Create in: tecabad.com/tecabad

First name: Prueba Initials:

Last name: Funcionamiento

Full name: Prueba Funcionamiento

User logon name: pfuncionamiento @tecabad.com

User logon name (pre-Windows 2000): TECABAD\ pfuncionamiento

< Back Next > Cancel

Figura 3.2 Creación de usuario para pruebas de Funcionamiento

EPN
Administración de Usuarios

ESCUELA POLITÉCNICA NACIONAL

Opciones

Actualizar Usuarios AD
Añadir Preguntas

Listado de Usuarios

	IDUser	CN	CorreoPersonal	NumeroCelular	Autorizado
Edit	1	usuario 1	ddeadd@hotmail.com	998589043	<input checked="" type="checkbox"/>
Edit	2	usuario 2			<input checked="" type="checkbox"/>
Edit	3	Gabriel Lopez	gabriel.lopez@epn.edu.ec	983582714	<input checked="" type="checkbox"/>
Edit	4	Danny Guaman			<input checked="" type="checkbox"/>
Edit	5	test	dabad@binaria.com.ec	998589043	<input checked="" type="checkbox"/>
Edit	6	david abad			<input type="checkbox"/>
Edit	7	test 1			<input type="checkbox"/>
Edit	8	maria			<input type="checkbox"/>
Edit	9	Usuario 3			<input type="checkbox"/>
Edit	10	Web	ddead@hotmail.com	998589043	<input type="checkbox"/>
Edit	11	Vilma Garcia			<input checked="" type="checkbox"/>
Edit	12	Franklin Sanchez	franklin.sanchez@epn.edu.ec	986183599	<input checked="" type="checkbox"/>

Figura 3.3 Actualizar usuarios de AD

Observar que el usuario creado se añadió a la Base de Datos, al momento de aumentar el usuario este es creado sin permisos para poder cambiar la contraseña, como se observa en la Figura 3.4.

EPN ESCUELA POLITÉCNICA NACIONAL

Administración de Usuarios

Opciones
Actualizar Usuarios AD
Añadir Preguntas

Listado de Usuarios

IDUser	CN	CorreoPersonal	NumeroCelular	Autorizado
Edit 1	usuario 1	ddeadd@hotmail.com	998589043	<input checked="" type="checkbox"/>
Edit 2	usuario 2			<input checked="" type="checkbox"/>
Edit 3	Gabriel Lopez	gabriel.lopez@epn.edu.ec	983582714	<input checked="" type="checkbox"/>
Edit 4	Danny Guaman			<input checked="" type="checkbox"/>
Edit 5	test	dabad@binaria.com.ec	998589043	<input checked="" type="checkbox"/>
Edit 6	david abad			<input type="checkbox"/>
Edit 7	test 1			<input type="checkbox"/>
Edit 8	maria			<input type="checkbox"/>
Edit 9	Usuario 3			<input type="checkbox"/>
Edit 10	Web	ddead@hotmail.com	998589043	<input type="checkbox"/>
Edit 11	Vilma Garcia			<input checked="" type="checkbox"/>
Edit 12	Franklin Sanchez	franklin.sanchez@epn.edu.ec	986183599	<input checked="" type="checkbox"/>
Edit 13	Prueba Funcionamiento			<input type="checkbox"/>

Figura 3.4 Autorización de cambio de contraseña

3.1.2 EDICIÓN DE USUARIO

Se edita la información de correo y teléfono desde el portal de administración, para eso, dar clic en editar, ver Figura 3.5, esto habilitará los campos para poder editar la información.

Edit 11	Vilma Garcia			<input checked="" type="checkbox"/>
Edit 12	Franklin Sanchez	franklin.sanchez@epn.edu.ec	986183599	<input checked="" type="checkbox"/>
Edit 13	Prueba Funcionamiento			<input type="checkbox"/>

Figura 3.5 Editar información del usuario

Editar los datos y presionar en actualizar, ver Figura 3.6.

EPN
Administración de Usuarios



ESCUELA POLITÉCNICA NACIONAL

Opciones
Actualizar Usuarios AD
Añadir Preguntas

Listado de Usuarios

	IDUser	CN	CorreoPersonal	Número Celular	Autorizado
Edit	1	usuario 1	ddeaad@hotmail.com	998589043	<input checked="" type="checkbox"/>
Edit	2	usuario 2			<input checked="" type="checkbox"/>
Edit	3	Gabriel Lopez	gabriel.lopez@epn.edu.ec	983582714	<input checked="" type="checkbox"/>
Edit	4	Danny Guaman			<input checked="" type="checkbox"/>
Edit	5	test	dabad@binaria.com.ec	998589043	<input checked="" type="checkbox"/>
Edit	6	david abad			<input type="checkbox"/>
Edit	7	test 1			<input type="checkbox"/>
Edit	8	maria			<input type="checkbox"/>
Edit	9	Usuario 3			<input type="checkbox"/>
Edit	10	Web	ddead@hotmail.com	998589043	<input type="checkbox"/>
Edit	11	Vilma Garcia			<input type="checkbox"/>
Edit	12	Franklin Sanchez	franklin.sanchez@epn.edu.ec	986183599	<input checked="" type="checkbox"/>
Update Cancel	13	Prueba Funcionamiento	ddeaad@hotmail.com	998589043	<input type="checkbox"/>

Figura 3.6 Actualizar Datos

3.1.3 AÑADIR PREGUNTAS

En el módulo añadir preguntas, se añade la pregunta ¿cuál es la marca de tu primer celular?, Ver Figura 3.7.

EPN
Administración de Usuarios



ESCUELA POLITÉCNICA NACIONAL

Opciones
Actualizar Usuarios AD
Añadir Preguntas

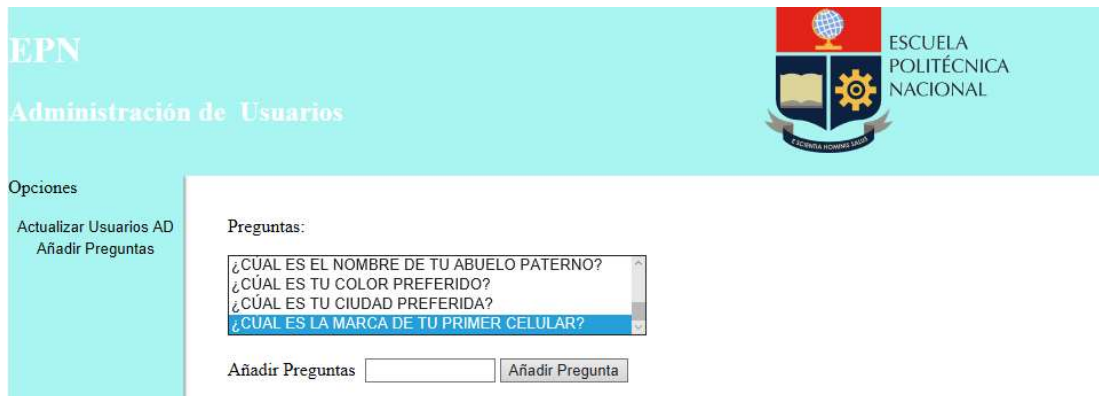
Preguntas:

¿CUAL ES EL NOMBRE DE TU PRIMERA MASCOTA?
 ¿CUAL ES TU PELICULA PREFERIDA?
 ¿CUAL ES LA MARCA DE TU VEHICULO PREFERIDO?
 ¿EN QUE PAIS TE GUSTARIA VIVIR?

Añadir Preguntas

Figura 3.7 Añadir pregunta al banco de Preguntas

Se comprueba que la pregunta ingresó al banco de preguntas, ver Figura 3.8.



EPN ESCUELA POLITÉCNICA NACIONAL

Administración de Usuarios

Opciones

Actualizar Usuarios AD
Añadir Preguntas

Preguntas:

- ¿CUAL ES EL NOMBRE DE TU ABUELO PATERNO?
- ¿CUAL ES TU COLOR PREFERIDO?
- ¿CUAL ES TU CIUDAD PREFERIDA?
- ¿CUAL ES LA MARCA DE TU PRIMER CELULAR?

Añadir Preguntas Añadir Pregunta

Figura 3.8 Pregunta añadida

3.2 PRUEBA DE FUNCIONAMIENTO MÓDULO USUARIOS

El módulo de Usuario permite al usuario editar los datos necesarios para poder validar su identidad al momento de cambiar la contraseña, el módulo de información presenta la información del correo y número de celular. Ver Figura 3.9.



EPN ESCUELA POLITÉCNICA NACIONAL

Administración de Información

Opciones

Información de Usuarios
Editar Preguntas
Añadir Preguntas

Alias: pfuncionamiento

Nombre: Prueba Funcionamiento

Correo: ddeaad@hotmail.com

Celular: +593- 998589043

Guardar

Figura 3.9 Información del Usuario Correo y Celular.

3.2.1 CAMBIO DE CORREO Y CELULAR

Se verifica poder cambiar la información del correo ddeaad@hotmail.com a David.abad@epn.edu.ec, así como la información de celular de +593-998589043 al +593-998589056, ver Figura 3.10.



The screenshot shows the EPN 'Administración de Información' interface. On the left, a sidebar contains 'Opciones' with sub-items 'Información de Usuarios', 'Editar Preguntas', and 'Añadir Preguntas'. The main content area is titled 'Alias: pfuncionamiento' and contains a form with the following fields: 'Nombre:' with the value 'Prueba Funcionamiento', 'Correo:' with the value 'david.abad@epn.edu.ec', and 'Celular: +593-' followed by '998589056'. A 'Guardar' button is located at the bottom of the form. The EPN logo and 'ESCUELA POLITÉCNICA NACIONAL' are visible in the top right corner.

Figura 3.10 Cambio de Correo y celular

3.2.2 AÑADIR PREGUNTAS DE VALIDACIÓN

Verificación de añadir preguntas, se añade la pregunta ¿Cuál es la marca de tu primer celular?, ver Figura 3.11.



The screenshot shows the EPN 'Administración de Información' interface. On the left, a sidebar contains 'Opciones' with sub-items 'Información de Usuarios', 'Editar Preguntas', and 'Añadir Preguntas'. The 'Añadir Preguntas' option is highlighted. The main content area is titled 'Alias: pfuncionamiento' and contains a form with a question: '¿CUAL ES LA MARCA DE TU PRIMER CELULAR?'. Below the question, there is a text input field containing 'NOKIA', a close button 'x', and an 'Añadir Pregunta' button. The EPN logo and 'ESCUELA POLITÉCNICA NACIONAL' are visible in the top right corner.

Figura 3.11 Añadir pregunta módulo Usuario

3.2.3 EDITAR RESPUESTA

Verificación de poder editar respuestas, seleccionar la pregunta a cambiar su respuesta, editar la respuesta y dar clic en actualizar, ver Figura 3.12.



Figura 3.12 Editar respuesta

3.3 PRUEBA DE FUNCIONAMIENTO MÓDULO CAMBIO DE CONTRASEÑA

El módulo de cambio de contraseña es el encargado de validar la identidad del usuario previo al cambio de contraseña.

3.3.1 BÚSQUEDA DEL USUARIO EN EL AD

El sistema valida si el usuario pertenece al directorio activo, para probar utilizar el usuario "tecabad\pfuncionamiento", ver Figura 3.13.

Si el usuario es correcto este continuará al siguiente módulo.



EPN

Cambio de Contraseña

ESCUELA POLITÉCNICA NACIONAL

SCP Sistema de Cambio de Contraseña

Usuario: (Dominio\Usuario)

Figura 3.13 Busca el usuario en el AD

3.3.2 INGRESO INCORRECTO DEL USUARIO

Al ingresar el nombre errado del usuario, en la parte superior de la pantalla aparece el mensaje “el usuario es incorrecto”, ver Figura 3.14.



Usuario Incorrecto

EPN

Cambio de Contraseña

ESCUELA POLITÉCNICA NACIONAL

SCP Sistema de Cambio de Contraseña

Usuario: (Dominio\Usuario)

Figura 3.14 Usuario Incorrecto

3.3.2.1 Error al tercer intento

Durante todo el proceso de autenticación, si el usuario se equivoca por tercera vez el programa se bloquea y pedirá cerrar el explorador para poder continuar, ver Figura 3.15, esto para disminuir los intentos de ataques que podría sufrir la página.



Figura 3.15 Bloqueo por intentos fallidos

3.3.3 PRUEBAS PRIMER ANILLO DE SEGURIDAD

Una vez ingresado un usuario válido del Directorio Activo, el sistema pasa a utilizar el primer anillo de seguridad para validar la persona, se prueba el envío de SMS y correo.

3.3.3.1 Envío de código a celular

Elegir SMS y enviar, en la parte superior izquierda la pantalla, aparece un mensaje indicando que el mensaje fue enviado correctamente, ver Figura 3.16.



Figura 3.16 Envío de código por mensaje de texto

En breves momentos, el celular recibe un SMS con el código de verificación, ver Figura 3.17.

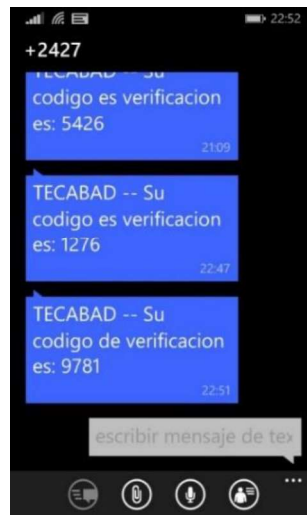



Figura 3.17 Envio de SMS

3.3.3.2 Envío de código por correo

Elegir correo y enviar, aparece mensaje indicando su envío exitoso, ver Figura 3.18.

mensaje enviado Correctamente

EPN
Cambio de Contraseña



ESCUELA
POLITÉCNICA
NACIONAL

Prueba Funcionamiento

Seleccione el mecanismo de validación

SMS
 Correo

El código de Verificación a sido enviado al siguiente correo: da****@*****ec

Figura 3.18 Envío de código por correo electrónico

En el correo personal, verificar la llegada del código a través de un mensaje donde el destinatario es “info@tecabad.com”, ver Figura 3.19.

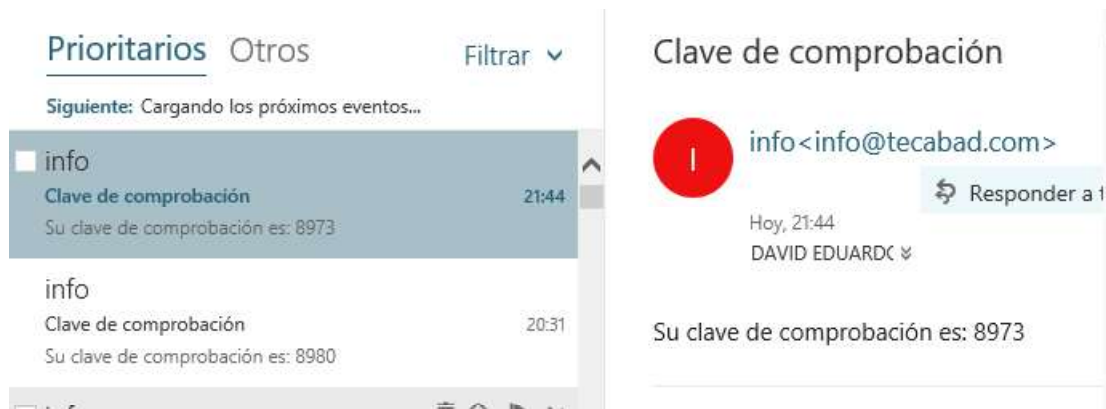


Figura 3.19 Verificación en el correo

3.3.3.3 Comprobación del Código

Colocar el código de verificación, ver Figura 3.20, y dar clic en comprobar.

mensaje enviado Correctamente.

EPN

ESCUELA
POLITÉCNICA
NACIONAL

Cambio de Contraseña

Prueba Funcionamiento

Seleccione el mecanismo de validación

SMS
 Correo

El código de Verificación a sido enviado al siguiente correo: da****@*****ec

Figura 3.20 Comprobación del código

Se comprueba la verificar el código se avanza a la siguiente pantalla, ver Figura 3.21.



Figura 3.21 pantalla para escoger pregunta de Validación

3.3.4 PRUEBAS SEGUNDO ANILLO DE SEGURIDAD

3.3.4.1 Preguntas de validación

Verificación de las preguntas de validación, ver Figura 3.22, escoger y contestar una de las preguntas, dar clic en contestar, si la pregunta es contestada correctamente, el sistema pasará a la pantalla que permite cambiar la contraseña.

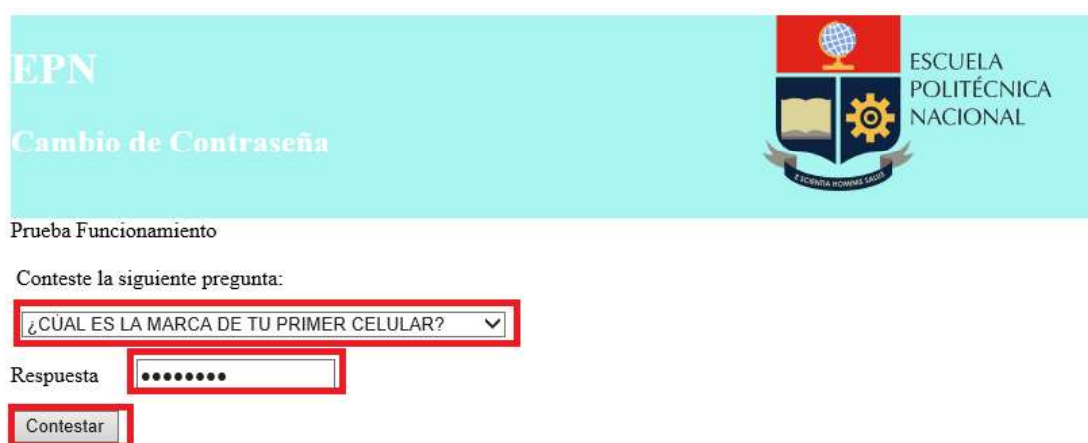
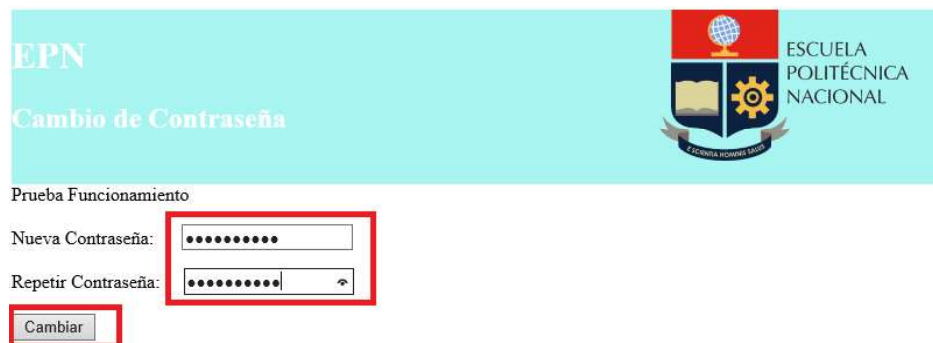


Figura 3.22 Contestar pregunta de validación

3.3.5 CAMBIAR CONTRASEÑA

Después de haber pasado los dos anillos de seguridad, colocar la nueva contraseña, ver Figura 3.23, colocar nuevamente la contraseña para asegurar su correcto ingreso.



The screenshot shows the 'Cambio de Contraseña' (Change Password) interface for EPN (Escuela Politécnica Nacional). The header includes the EPN logo and the text 'ESCUELA POLITÉCNICA NACIONAL'. Below the header, there is a section titled 'Prueba Funcionamiento' (Functionality Test). This section contains two password input fields: 'Nueva Contraseña:' (New Password) and 'Repetir Contraseña:' (Repeat Password). Both fields are currently filled with ten dots. A 'Cambiar' (Change) button is located below the second field. Red boxes highlight the 'Nueva Contraseña' field, the 'Repetir Contraseña' field, and the 'Cambiar' button.

Figura 3.23 Cambiar contraseña

Una vez cambiada la contraseña aparece un mensaje en la parte superior izquierda, indicando que el usuario fue cambiado de contraseña, ver Figura 3.24, y el sistema regresa a la pantalla de inicial.



The screenshot shows the 'Cambio de Contraseña' interface after a successful password change. A red-bordered message box at the top left contains the text 'El usuario fue cambiado de contraseña' (The user's password was changed). Below the message, the header includes the EPN logo and the text 'ESCUELA POLITÉCNICA NACIONAL'. The main content area is titled 'SCP Sistema de Cambio de Contraseña'. Below this title, there is a 'Usuario:' label followed by an empty text input field and the text '(Dominio\Usuario)'. At the bottom, there is a 'Recuperar Contraseña' (Recover Password) button.

Figura 3.24 Mensaje de confirmación

Capítulo 4

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Dentro del análisis de los procesos de autenticación se observó que la combinación de los dos mecanismos de autenticación “de algo que posee” y “algo que sabe”, fue la mejor estrategia que se pudo emplear ya que se adapta al problema planteado que es poder cambiar la contraseña de manera segura desde fuera de la organización, sin necesitar algún dispositivo adicional como un lector de huellas digitales.

Los dos mecanismos de seguridad planteados se ven reflejados en dos anillos de seguridad, el primer anillo de seguridad que utiliza el método “de algo que posee” el usuario, como es un celular o un correo personal, lo convierte en un mecanismo extremadamente confiable, el cual es complementado con el segundo anillo de seguridad que utiliza el método de autenticación “de algo que posee”, ya que en el caso de que el dispositivo celular o el correo personal puedan ser accedidos por una persona extraña, este tendría que conocer aspectos personales del usuario lo que aumenta el mecanismo de seguridad.

El proyecto demuestra la factibilidad de utilizar la metodología XP para el desarrollo de este programa, al ser una metodología altamente adaptativa se modifica para suplir la ausencia de un segundo programador, XP permite cualquier adaptación con la única condición que se especifique la misma desde un principio.

Al utilizar la metodología XP permite observar entregables rápidos que muestran avances del prototipo, al realizar la misma poco a poco pudo dar una visión clara

de cómo ir integrando todos los componentes de software necesarios para el desarrollo total del proyecto.

El prototipo utiliza como plataforma de infraestructura, nuevas tecnologías como la nube de Azure, la misma provee con las máquinas virtuales para el desarrollo de este, así como Office 365 como servicio en la nube que se encuentra integrado al Directorio Activo.

El software que permita realizar el cambio de contraseña de un usuario del Directorio Activo de manera segura utiliza formularios ASP de .NET y las librerías de .net *Framework*., el mismo está conformado por tres formularios web, uno administrativo que permite cargar la información del AD en la base de datos, así como editar el correo y el teléfono de los usuarios y añadir preguntas que los puedan utilizarse para validación. Un segundo formulario que permite a los usuarios editar su información de correo y teléfono, así como sus preguntas de validación y un tercer módulo que permite el cambio de contraseña desde fuera de la organización.

4.2 RECOMENDACIONES

El desarrollo en web tiene limitantes sobre todo cuando un formulario realiza un proceso de *PostBack* que coloca a los objetos que se utilicen como variables sin valores, desvía el desarrollo tradicional que se tiene a nivel de aplicación, por lo que se deberá considerar nuevas estrategias de desarrollos como son utilizar objetos que se encuentren atados a los formularios o ver mecanismo para contrarrestar los efectos de un *PostBack* como puede ser volver a cargar los objetos cada vez que esto sucede.

Se recomienda como plataforma de SMS al servicio de Twilio ya que este ofrece librerías de desarrollo en varios lenguajes de programación, como fue el caso de este proyecto que utilizó C#.

El uso de nuevas tecnologías como Office 365 y Azure que permite una rápida implementación de los servicios a utilizar como fueron las máquinas virtuales en la plataforma de Azure, y el servicio de correo en Office 365.

La utilización de certificados digitales que permiten la codificación de la comunicación entre el servidor y el navegador del cliente, garantizando una conexión segura con el servidor.

El uso de autenticación de Windows en los sitios de internet de un servidor de IIS, al activar esta característica se puede obtener de manera muy rápida un sistema de autenticación, sobre todo si el servidor de IIS es un servidor miembro de un dominio.

REFERENCIAS

- [1] Microsoft, «Información general sobre ASP.NET,» Microsoft, Noviembre 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx). [Último acceso: 20 Julio 2016].
- [2] M. D. Network, «Estructura de archivos de configuración de ASP.NET,» Microsoft, Noviembre 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/w7w4sb0w\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/w7w4sb0w(v=vs.90).aspx).
- [3] M. D. Network, «Información general sobre la depuración en ASP.NET,» Microsoft, Noviembre 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/ms227556\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/ms227556(v=vs.100).aspx).
- [4] M. D. Networking, «Información general sobre el ciclo de vida de una aplicación ASP.NET para IIS 5.0 y 6.0,» Noviembre 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/ms178473\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/ms178473(v=vs.100).aspx).
- [5] M. D. Nerwork, «Información general sobre los diseñadores de controles ASP.NET,» Microsoft, Noviembre 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/wxh45wzs\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/wxh45wzs(v=vs.100).aspx).
- [6] Microsoft, «Información general sobre páginas Web ASP.NET,» Microsoft, Noviembre 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/428509ah\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/428509ah(v=vs.100).aspx). [Último acceso: 25 noviembre 2016].
- [7] Microsoft, «Información general acerca de .NET Framework,» Microsoft, [En línea]. Available: [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx). [Último acceso: 25 Septiembre 2016].
- [8] Microsoft, «Introducción a Active Directory,» 18 Octubre 2000. [En línea]. Available: <https://support.microsoft.com/es-es/help/196464>.
- [9] Microsoft Press, «Installing and administering Active Directory,» de *Installing and Configuring Windows Server 2012 R2*, Redmond, Washington: Box Twelve Communications, 2014, pp. 2-13.
- [10] Microsoft, «¿Qué es Office 365?,» mkarich, [En línea]. Available: <https://blogs.technet.microsoft.com/microsoftlatam/2011/05/05/qu-es-office-365/>. [Último acceso: 25 noviembre 2016].
- [11] O. 365, «Office 365 Plan Options,» 24 mayo 2016. [En línea]. Available: <https://technet.microsoft.com/en-us/library/office-365-plan-options.aspx>.
- [12] welivesecurty, «Introducción a la autenticación: cómo probar que realmente eres tú,» ESET, 4 Mayo 2016. [En línea]. Available: <https://www.welivesecurity.com/la-es/2016/05/04/autenticacion-como-probar-que-eres-tu/>.
- [13] I. B. d. conocimiento, «Identificación y autenticación,» IBM, 13 Diciembre 2016. [En línea]. Available:

- https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q009740_.htm.
- [14] C. d. S. d. ESET, «Las cuatro “A” de la administración de cuentas,» ESET, 23 Febrero 2016. [En línea]. Available: <https://www.welivesecurity.com/la-es/2016/02/23/administracion-de-cuentas/>.
- [15] pandaid.com, «¿Qué es la Autenticación Fuerte?,» pandaid.com, enero 2017. [En línea]. Available: <http://www.pandaid.com/que-es-la-autenticacion-fuerte/>.
- [16] A. M. Ballesté, A. Solanas y J. C. Roca, «Identificación, autenticación y control de acceso,» [En línea]. Available: [https://www.exabyteinformatica.com/uoc/Dactiloscopia/Identidad_digital/Identidad_digital_\(Módulo_1\).pdf](https://www.exabyteinformatica.com/uoc/Dactiloscopia/Identidad_digital/Identidad_digital_(Módulo_1).pdf). [Último acceso: 25 octubre 2016].
- [17] J. Shier, «Esenciales de seguridad: ¿Qué es la autenticación de dos factores?,» naked Security by SOPHOS, 15 abril 2013. [En línea]. Available: <https://nakedsecurity.sophos.com/es/2013/10/10/security-essentials-what-is-two-factor-authentication/>.
- [18] H. F. Tipton y M. Krause, «Information Security Management Handbook,» de *Information Security Management Handbook*, CISSP, 2004.
- [19] X. Ferrer Ardanaz, «Composición de la identidad digital,» Publica, [En línea]. Available: <https://blogs.deusto.es/master-informatica/composicion-de-la-identidad-digital/>.
- [20] T. Armas, «Complejidad Y Longitud De Las Contraseñas,» owasp.org, Noviembre 2012. [En línea]. Available: https://www.owasp.org/index.php/Complejidad_Y_Longitud_De_Las_Contrase%C3%B1as.
- [21] S. Pakala, «Preguntas Frecuentes sobre Seguridad en Aplicaciones Web (OWASP FAQ),» OWASP.org, 25 enero 2005. [En línea]. Available: <https://www.um.es/atika/documentos/FAQSeguridadAplicacionesWebOWASP.pdf>.
- [22] J. Santos, «Windows 8 permite iniciar sesión con imágenes en vez de contraseñas,» GENBETA, 12 Agosto 2012. [En línea]. Available: <https://www.genbeta.com/windows/windows-8-permite-iniciar-sesion-con-imagenes-en-vez-de-contrasenas>.
- [23] P. mediacenter, «Cómo convertir un pendrive en la llave de tu cuenta de Google,» Panda mediacenter, 10 Agosto 2015. [En línea]. Available: <http://www.pandasecurity.com/spain/mediacenter/seguridad/pendrive-llave-cuenta-google/>.
- [24] D. G. BILIC, «¿Qué tan seguro es un mensaje SMS?,» ESET, 15 Diciembre 2015. [En línea]. Available: <https://www.welivesecurity.com/la-es/2015/12/15/que-tan-seguro-es-mensaje-sms/>.

- [25] A. Sanchez, «MasterCard quiere que el selfie sea el próximo método de pago,» hipertextual, 25 febrero 2016. [En línea]. Available: <https://hipertextual.com/2016/02/mastercard-selfie>.
- [26] A. F. Garzón Velosa, N. Pérez Valencia y J. F. Ochoa Burgos, Mecanismo de Autenticación e identificación, Bogotá: Servicio Nacional de Aprendizaje SENA, 2014.
- [27] 123RF, Artist, *Foto de archivo - Iconos de autenticación biométrica establecidos con iris cara retina y reconocimiento de huellas dactilares símbolos aislados ilustración vectorial*. [Art]. 123RF, 2016.
- [28] A. Rodríguez, «La problemática de la biometría como método de autenticación,» CERTSI, 25 Septiembre 2013. [En línea]. Available: <https://www.certsi.es/blog/problematika-biometria-autenticacion>.
- [29] el Universo, «Ecuador tiene 16,9 millones de líneas celulares, cifra que supera a su población,» El universo, 22 enero 2013. [En línea]. Available: <http://www.eluniverso.com/2013/01/22/1/1356/ecuador-tiene-169-millones-lineas-celulares-cifra-supera-poblacion.html>.
- [30] C. d. S. d. ESET, «Doble factor de autenticación: ¿qué es y por qué lo necesito?,» ESET, 19 febrero 2014. [En línea]. Available: <https://www.welivesecurity.com/la-es/2014/02/19/doble-factor-autenticacion-que-es-porque-lo-necesito/>.
- [31] L. Calabria y P. Píriz, Metodología XP, Montevideo, Uruguay : Universidad ORT Uruguay, 2003.
- [32] Microsoft, «¿Qué es IaaS?,» AZURE, [En línea]. Available: <https://azure.microsoft.com/es-es/overview/what-is-iaas/>. [Último acceso: 4 noviembre 2016].
- [33] twilio, «Api SMS,» [En línea]. Available: <https://www.twilio.com/sms/api>.
- [34] Microsoft, «Instalar IIS y los módulos de ASP.NET,» Febrero 2012. [En línea]. Available: [https://technet.microsoft.com/es-es/library/hh831475\(v=ws.11\).aspx](https://technet.microsoft.com/es-es/library/hh831475(v=ws.11).aspx). [Último acceso: 10 marzo 2016].
- [35] L. Koskela, Application of the new production philosophy to construction, Finland: VTT Building Technology, 1992, p. 13.
- [36] Microsoft, «DataSource,» [En línea]. Available: <https://www.bing.com/search?q=DataSource+aspnet&qs=n&form=QBRE&sp=-1&pq=datasource&sc=0-10&sk=&cvid=1FDD8BB7BB6F4780BFA2DB5AF316137F>. [Último acceso: 30 Octubre 2016].
- [37] Microsoft, «Descripción de la funcionalidad de dominios y bosques,» TechNet, [En línea]. Available: [https://technet.microsoft.com/es-es/library/cc771294\(v=ws.11\).aspx](https://technet.microsoft.com/es-es/library/cc771294(v=ws.11).aspx). [Último acceso: 20 Octubre 2016].
- [38] Microsoft, «Developer Netwok,» 2007. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx). [Último acceso: 26 noviembre 2016].
- [39] Microsoft, «DropDownList Class .Net,» [En línea]. Available: [https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.dropdownlist\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.dropdownlist(v=vs.110).aspx).

- [40] «IDataItemContainer Interface .Net Framework,» [En línea]. Available: [https://msdn.microsoft.com/en-us/library/system.web.ui.idataitemcontainer\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.web.ui.idataitemcontainer(v=vs.110).aspx).
- [41] CRC Press LLC, Information Security Managemer Hansbook, vol. 5ta, H. F. Tipton y M. Krause, Edits., Boca Raton, Florida: AUERBACH, 2004.
- [42] microsoft, «msdn subcription,» [En línea]. Available: <https://msdn.microsoft.com/en-us/subscriptions/aa336858.aspx>. [Último acceso: 5 noviembre 2016].
- [43] A. Hofacker, Rapid lean construction - quality rating model, Manchester: s.n., 2008.
- [44] Microsoft, «Roles, servicios de rol y características,» 1 diciembre 2016. [En línea]. Available: [https://technet.microsoft.com/es-es/library/cc754923\(v=ws.11\).aspx](https://technet.microsoft.com/es-es/library/cc754923(v=ws.11).aspx).
- [45] Aprendiendo Exchange.com, «Relay en Exchange 2013 / 2016,» Daniel Núñez Banega , 20 Abril 2016. [En línea]. Available: <http://aprendiendoexchange.com/relay-en-exchange-2013>.