



UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN

**Aplicación web para la gestión de los procesos  
de configuración de promociones en una  
empresa de Telefonía.**

Trabajo Especial de Grado  
Presentado ante la ilustre  
Universidad Central de Venezuela  
Por el bachiller:  
Ricardo Mangialomini Milici

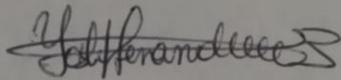
Tutores:  
Profa. Yosly Hernández  
Prof. Leonardo Chacón  
Caracas, 24 de mayo de 2016

## ACTA

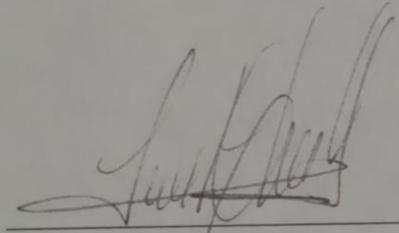
Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **“Aplicación web para la gestión de los procesos de configuración de promociones en una empresa de Telefonía.”** y presentado por el Bachiller Ricardo Mangialomini Milici C.I: 17.908.154, a los fines de optar al título de Licenciado en Computación, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 24 de mayo de 2016 a la 2:00pm horas, para que el autor lo defendiera en forma pública, lo que este hizo en la Sala de Postgrado de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

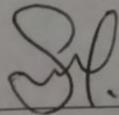
En fe de lo cual se levanta la presente Acta, en Caracas el día 24 de mayo de 2016



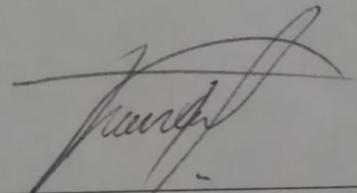
Prof. Yosly Hernández  
(Tutora)



Prof. Leonardo Chacón  
(Tutor)



Prof. Franklin Sandoval  
(Jurado)



Prof. Franky Uzcátegui  
(Jurado)

## **AGRADECIMIENTOS**

A mis padres María y Salvatore, por su apoyo incondicional, los valores y la educación que me han brindado. Por ofrecerme todo lo posible para salir adelante y llevarme siempre por el mejor camino a pesar de las dificultades.

A mi hermano Jose, por ser un ejemplo a seguir y el apoyo que me ha brindado a lo largo de mi vida.

A la ilustre Universidad Central de Venezuela, por permitirme realizar mi carrera profesional y por el conocimiento y los valores que pude adquirir en el camino.

A mis tutores Yosly y Leonardo, por todo el apoyo que me han brindado durante el desarrollo de este Trabajo Especial de Grado.

A mis profesores, por la formación y el apoyo que me han brindado a lo largo de este trayecto académico.

A mis amigos y compañeros de estudio, por todas esas vivencias y horas de estudio que me han permitido crecer como profesional.

Al personal de Telefónica S.A., por permitirme realizar mi Trabajo Especial de Grado dentro de tan prestigiosa empresa.

A todos aquellos que formaron parte de mi educación, que me han ayudado a crecer como profesional y me han dejado alguna enseñanza de vida.

## RESUMEN

La compañía Telefónica S.A. es una de las principales empresas de telecomunicaciones del mundo, enfocada principalmente en los negocios de telefonía fija, móvil, internet y TV digital. La gerencia de configuración de servicios de Telefónica S.A, se encarga de desarrollar una serie de servicios para los clientes de la empresa en las distintas líneas de negocio, entre los cuales destacan las promociones otorgadas a los clientes de la plataforma prepago. Este beneficio requiere del desarrollo de un conjunto de actividades tales como la reserva de identificadores, programación de scripts y pruebas de funcionamiento, que consumen una gran cantidad tiempo y un esfuerzo notable por parte del equipo de desarrollo, lo que genera errores y retrasos en las entregas de los requerimientos. El presente Trabajo Especial de Grado tuvo como objetivo realizar una herramienta que permitiera a sus usuarios configurar de forma sencilla todas las promociones solicitadas en la pila de requerimientos del departamento, con la finalidad de minimizar los errores humanos y reducir el tiempo empleado en la configuración de promociones. La metodología de desarrollo utilizada fue SCRUM y se emplearon tecnologías como Microsoft .NET, C#, SQL Server y Oracle SQL, fue desarrollada una aplicación web que permite realizar todo la gestión de las promociones, el cual incluye la reserva de identificadores, desarrollo de scripts de configuración y la ejecución de los mismos en la base de datos de pruebas de la empresa. Como valor agregado, se pueden almacenar las promociones en una base de datos local, con la finalidad de ser utilizadas a futuro para nuevos requerimientos. Se puede concluir que el desarrollo de este sistema con el uso de las tecnologías mencionadas brinda una ventaja estratégica en cuanto al costo, tiempo y esfuerzo aplicado por el equipo de trabajo, mejorando la calidad y el desarrollo de los procesos de la gerencia. Los resultados obtenidos se pueden tomar como un punto de partida para desarrollar nuevas aplicaciones que permitan simplificar otras tareas asignadas a la Gerencia de Configuración de Servicios de Telefónica S.A.

**Palabras Clave:** Aplicación, Bases de Datos, promociones de telefonía, Tecnologías web, Telefonía.

## ÍNDICE GENERAL

ACTA.....	¡Error! Marcador no definido.
AGRADECIMIENTOS .....	ii
RESUMEN.....	iii
ÍNDICE GENERAL.....	iv
ÍNDICE DE FIGURAS.....	vi
ÍNDICE DE TABLAS .....	viii
INTRODUCCIÓN .....	1
CAPITULO I.....	3
1.1. Contexto de Investigación .....	3
1.2. Planteamiento del problema .....	4
1.3. Justificación .....	8
1.4. Objetivo General.....	9
1.5. Objetivos específicos.....	9
1.6. Antecedentes.....	10
1.7. Alcance .....	10
1.8. Metodología de Desarrollo .....	11
CAPÍTULO II .....	16
2.1. Promociones Movistar .....	16
2.2. Lenguajes de Programación.....	17
2.3. Microsoft .NET.....	18
2.4. C Sharp y .Net Framework.....	20
2.5. Base de Datos .....	23
2.6. Aplicaciones web.....	31
2.7. Arquitectura de Aplicaciones web.....	32
2.8. Beneficios de las aplicaciones web .....	36
2.9. Desventajas de las aplicaciones web .....	36
CAPITULO III.....	37
3.1. Descripción de la aplicación de la tecnología .....	37
3.2. Pruebas de usabilidad .....	56
CAPITULO IV.....	63

4.1. Presentación de la Aplicación – GesPromos.....	63
4.2. Módulos de la Aplicación .....	64
CONCLUSIONES .....	86
RECOMENDACIONES .....	87
REFERENCIAS.....	88
MANUAL DE USUARIO .....	91
MANUAL DE TECNOLOGÍA .....	106
CUESTIONARIO DE USABILIDAD DE GESPROMOS .....	109

## ÍNDICE DE FIGURAS

Figura 1: Diagrama de flujo del ciclo de vida de un requerimiento.....	5
Figura 2: diagrama de flujo: configuración una promoción.....	7
Figura 3: Flujo de SCRUM.....	13
Figura 4. Compilación de un código en C#.....	23
Figura 5: Ejemplo de disposición relacional.....	26
Figura 6: Tecnologías empleadas en el cliente y en el servidor web.....	31
Figura 7: Arquitectura de tres capas.....	33
Figura 8: Modelo Vista Controlador.....	35
Figura 9. Arquitectura de la aplicación.....	42
Figura 10. Código definido para las condiciones iniciales de la promoción.....	42
Figura 11. Parámetros principales de la promoción.....	43
Figura 12. Fragmento de código RAZOR .NET.....	44
Figura 13. Fragmento de código HTML.....	44
Figura 14: Creación del archivo t2_config.sql.....	45
Figura 15. Buscar Código de promoción.....	47
Figura 16. Verificar código de mensaje.....	47
Figura 17. Instalar configuración.....	48
Figura 18. Instalación de la configuración - Insertar modelos.....	49
Figura 19. Ejecutar contingencia.....	50
Figura 20: Modelo de Base de Datos Local.....	51
Figura 21. Código de carga de promociones.....	51
Figura 22. Carga de datos.....	52
Figura 23. Almacenamiento de la promoción.....	52
Figura 24. Código para eliminar promociones.....	54
Figura 25: Fragmento de código CSS.....	54
Figura 26: Test de usabilidad. Respuestas de la primera pregunta.....	57
Figura 27: Test de usabilidad. Respuestas de la segunda pregunta.....	58
Figura 28: Test de usabilidad. Respuestas de la tercera pregunta.....	58
Figura 29: Test de usabilidad. Respuestas de la cuarta pregunta.....	59

Figura 30: Test de usabilidad. Respuestas de la quinta pregunta.....	59
Figura 31: Test de usabilidad. Respuestas de la sexta pregunta.....	60
Figura 32: Test de usabilidad. Respuestas de la séptima pregunta.....	60
Figura 33: Test de usabilidad. Respuestas de la octava pregunta.....	61
Figura 34: Test de usabilidad. Respuestas de la novena pregunta.....	61
Figura 35: Test de usabilidad. Respuestas de la décima pregunta.....	62
Figura 36: Diagrama de flujo para la configuración de promociones mediante GesPromos.....	63
Figura 37. Pantalla de Inicio.....	64
Figura 38. Pantalla de Inicio luego de autenticarse.....	65
Figura 39: Pantalla de inicio desde un dispositivo móvil.....	66
Figura 40. Manual de Configuración.....	67
Figura 41. Lista de errores comunes.....	68
Figura 42: Inicio de Sesión.....	69
Figura 43. Diagrama de flujo – Crear Promoción.....	70
Figura 44. Condiciones iniciales.....	72
Figura 45: Configuración de la promoción. Primera parte.....	73
Figura 46: Configuración de la promoción. Segunda parte.....	74
Figura 47. Gestión de una Promoción.....	75
Figura 48. Archivo t2_config.sql.....	76
Figura 49. Archivo cont_t2_config.sql.....	77
Figura 50: Gestión de la promoción.....	78
Figura 51. Spool de instalación.....	78
Figura 52: Gestión de la promoción.....	79
Figura 53. Spool de contingencia.....	80
Figura 54. Diagrama de flujo – Buscar Promoción.....	81
Figura 55. Diagrama de flujo – Eliminar Promoción.....	82
Figura 56. Gestión de promociones almacenadas.....	83
Figura 57. Buscar Promoción.....	83
Figura 58. Buscar promoción por código.....	84
Figura 59. Buscar promoción por código – Error.....	84
Figura 60. Eliminar promociones.....	85

## ÍNDICE DE TABLAS

Tabla 1: Product Backlog.....	38
Tabla 2: Sprint Backlog. Primera Parte.....	39
Tabla 3: Sprint Backlog. Segunda Parte.....	40
Tabla 4: Sprint Backlog de la primera iteración.....	41
Tabla 5: Sprint Backlog de la segunda iteración.....	45
Tabla 6: Sprint Backlog de la tercera iteración.....	46
Tabla 7: Sprint Backlog de la cuarta iteración.....	48
Tabla 8: Sprint Backlog de la quinta iteración.....	49
Tabla 9: Sprint Backlog de la sexta iteración.....	50
Tabla 10: Sprint Backlog de la séptima iteración.....	53
Tabla 11: Funcionalidades entregadas en cada iteración.....	55

## INTRODUCCIÓN

Las empresas de telecomunicaciones se han apoyado en la tecnología para el desarrollo de todos sus procesos, haciendo uso de plataformas y herramientas que son renovadas día a día en un entorno muy competitivo e innovador. Telefónica S.A es parte importante de este grupo de empresas, dedicando su esfuerzo en satisfacer las necesidades de sus clientes ofreciendo el mejor servicio posible, en una amplia gama de plataformas para asegurar la mejor y más variada comunicación del cliente con su entorno y el resto del mundo.

La tecnología ha permitido el desarrollo de sistemas y aplicaciones para el procesamiento de información y la ejecución de actividades específicas, con el objetivo de ahorrar tiempo, recursos y optimizar procesos en el desarrollo de las actividades de la organización. El uso de aplicaciones para actividades específicas permite a las organizaciones mejorar la producción, reducir costos e incrementar la productividad de sus empleados en sus actividades.

En su afán por brindar un servicio óptimo, Telefónica S.A se ve en la necesidad de evaluar de forma constante sus procesos y llevar a cabo nuevas estrategias para renovar sus actividades. Es por esto que se ha desarrollado una aplicación web para la gestión de los procesos de configuración de promociones de telefonía prepago, con el fin de optimizar el proceso de desarrollo para las distintas promociones otorgadas por la empresa en Venezuela, lo que trae beneficios a todos los empleados implicados en la configuración de los requerimientos ideados por la organización.

Mediante el uso de las tecnologías y la metodología de desarrollo señalada en este documento, fue desarrollada una herramienta capaz gestionar la configuración y el procedimiento de pruebas necesario para la configuración de una promoción, logrando además almacenar las actividades realizadas por sus usuarios.

El presente Trabajo Especial de Grado está estructurado en cuatro (4) capítulos, los cuales contienen la siguiente información:

Capítulo I: se explica el contexto de investigación, se realiza la descripción del planteamiento del problema. También se describe la justificación de la aplicación, sus antecedentes, objetivo general, objetivos específicos, alcance y la metodología de desarrollo escogida para el proyecto.

Capítulo II: en este capítulo se expone todo el marco teórico relacionado a las diferentes tecnologías que fueron utilizadas dentro del desarrollo de la aplicación, tales como bases de datos y lenguajes de programación.

Capítulo III: Marco Aplicativo: en este capítulo se describe como fue realizada la aplicación, haciendo uso de la metodología de desarrollo SCRUM y sus elementos. Además, se describe la arquitectura implantada la aplicación.

Capítulo IV: Resultados: se realiza una descripción de la aplicación, mostrando las diferentes vistas, sus funcionalidades y los archivos generados.

Para finalizar, se exponen las conclusiones, algunas recomendaciones para futuros desarrollos y las referencias bibliográficas de los autores citados en este documento.

# CAPITULO I

## PROBLEMA DE INVESTIGACIÓN

En el presente capítulo se expone el contexto de investigación, sus antecedentes, y se realiza el planteamiento del problema identificado en la gerencia de configuración de Prepago de Telefónica S.A, en cuanto a la gestión actual de las promociones otorgadas a sus clientes en la plataforma de prepago. Además se define el objetivo general y los objetivos específicos, se describe la justificación e importancia de haber solucionado dicho problema y se detalla la solución propuesta.

### **1.1. Contexto de Investigación**

Telefónica S.A (2015) es una de las principales empresas de telecomunicaciones del mundo. Su actividad está enfocada principalmente en los negocios de telefonía fija y móvil, con la banda ancha como herramienta clave para el desarrollo de ambos negocios. La compañía tiene presencia en 25 países y cuenta con una base de clientes que supera los 277,8 millones a marzo de 2010. Tiene una fuerte presencia en España, Europa y Latinoamérica, donde la compañía concentra fundamentalmente su estrategia de crecimiento, enfocada en atender las necesidades locales de cada país.

Telefónica S.A, bajo su marca Movistar, ofrece sus servicios a más de 12 millones de usuarios, quienes disfrutan de telefonía móvil, telefonía fija, Internet móvil, TV digital, transmisión de datos y servicios de valor agregado; también brinda soluciones corporativas a empresas y a pequeñas y medianas industrias (Pymes).

Como empresa, Telefónica S.A tiene como visión abrirse camino para seguir transformando posibilidades en realidades, con el fin de crear valor para los empleados, clientes, accionistas, socios a nivel global y a toda la sociedad.

### **1.1.1. Gerencia de Configuración de Servicios**

La gerencia de configuración de servicios de Telefónica S.A, es un departamento que tiene como finalidad configurar en las distintas líneas de negocio una serie de servicios a los clientes de la empresa, entre los que destacan:

- Planes
- Bonos
- Promociones
- SVA (Servicios de valor agregado)

Dentro del personal del departamento, se encuentra en primer lugar el Gerente del área. Luego, aparecen los líderes de prepago y postpago. Seguidamente, se encuentran un conjunto de coordinadores, los cuales deben velar por el desarrollo y entrega de cada requerimiento para las áreas del negocio que tienen asignadas. Por último, aparecen los analistas de prepago y postpago, encargados de desarrollar todos los requerimientos que son asignados a la gerencia según la plataforma que les corresponda.

### **1.2.Planteamiento del problema**

En la gerencia de configuración de Telefónica S.A se gestiona un gran número de requerimientos solicitados por otras áreas del negocio. El volumen de tareas asignadas al departamento puede generar una carga de trabajo bastante alta para el recurso humano disponible en el departamento.

Para llevar a cabo un requerimiento, el mismo debe pasar por un proceso de desarrollo que incluye varias Gerencias o departamentos de Telefónica S.A., en la figura 1 se puede observar el diagrama de flujo para cualquier requerimiento solicitado.

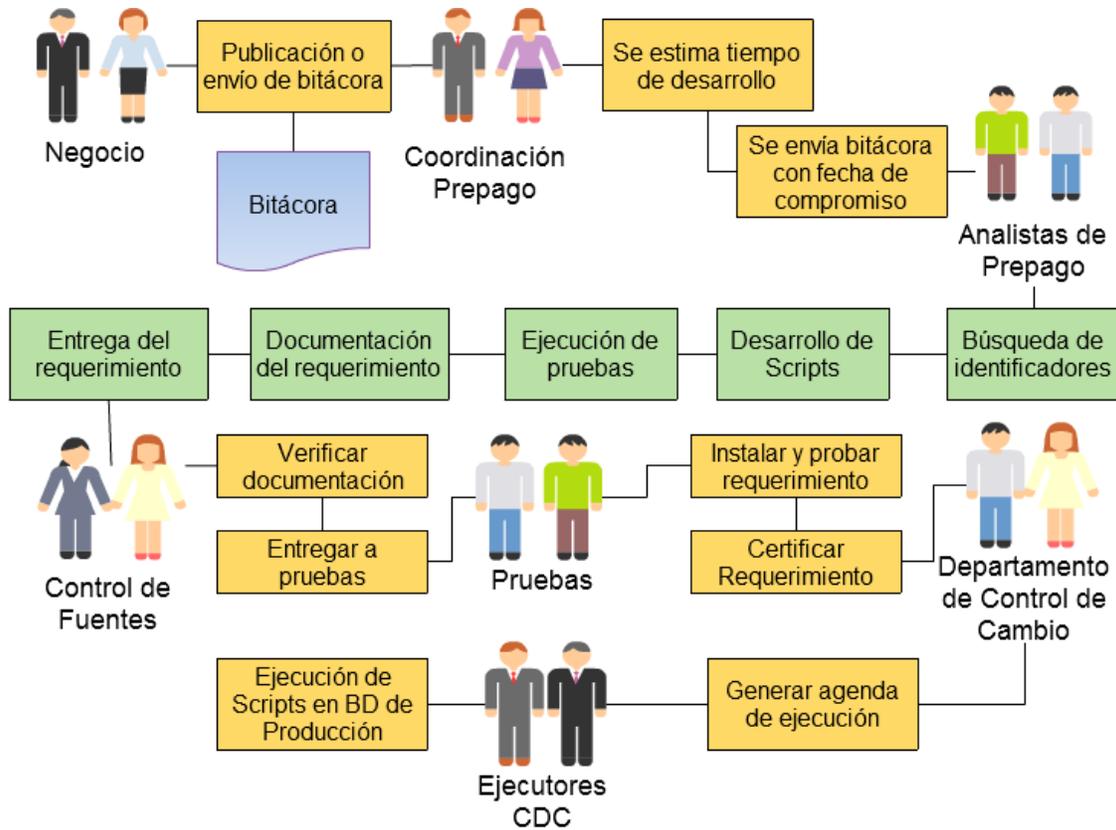


Figura 1. Diagrama de flujo del ciclo de vida de un requerimiento.

Entre los tipos de requerimientos recibidos por la gerencia de configuración de servicios, se tienen las promociones destinadas a la plataforma prepago, las cuales son creadas para incentivar a cierto tipo de clientes según su estatus actual, por lo que se pueden identificar varios tipos y clasificarlas según el beneficio que otorgan.

El proceso de creación de una promoción comienza al momento de recibir una bitácora por parte del negocio. Este documento permite identificar todos los aspectos necesarios para la creación de la promoción, tales como:

- Nombre de la promoción
- Línea del negocio beneficiada
- Beneficio a otorgar
- Condiciones para recibir la promoción
- Tiempo durante el cual el cliente recibe el beneficio

Una vez que el desarrollador ha establecido el tipo de promoción que debe programar, procede a crear el caso. En general, se seguían los siguientes pasos:

- Establecer el escenario que cumple la promoción.
- Establecer los scripts a utilizar para la creación de la promoción. En este paso puede optar por las siguientes opciones, tomando en cuenta que debe adaptar los scripts a este nuevo requerimiento:
  - Utilizar los manuales de configuración destinados a la creación de promociones.
  - Reutilizar código generado para la creación de promociones anteriores, similares a la promoción solicitada.
  - Desarrollar sus propios scripts desde cero.
  - Buscar los identificadores disponibles y reservar los necesarios para la creación de la promoción.
  - Desarrollar los scripts bajo el lenguaje Oracle SQL, utilizando los identificadores reservados y cumpliendo todos los requisitos establecidos en la bitácora.
  - Realizar pruebas en Desarrollo que certifiquen que el requerimiento funciona correctamente.
  - Realizar la documentación establecida
  - Promover el requerimiento a pruebas
  - Solicitar la instalación del requerimiento en la base de datos de pruebas.
  - Corregir, de ser necesario, las incidencias al momento de la instalación.
  - Certificar la correcta instalación del requerimiento en el ambiente de pruebas.
  - Corregir, de ser necesario, las incidencias generadas en las pruebas hechas sobre el requerimiento.
  - Realizar seguimiento de la promoción hasta que la misma se encuentre en producción

En la figura 2, se muestra el diagrama de flujo del proceso de configuración de una promoción:

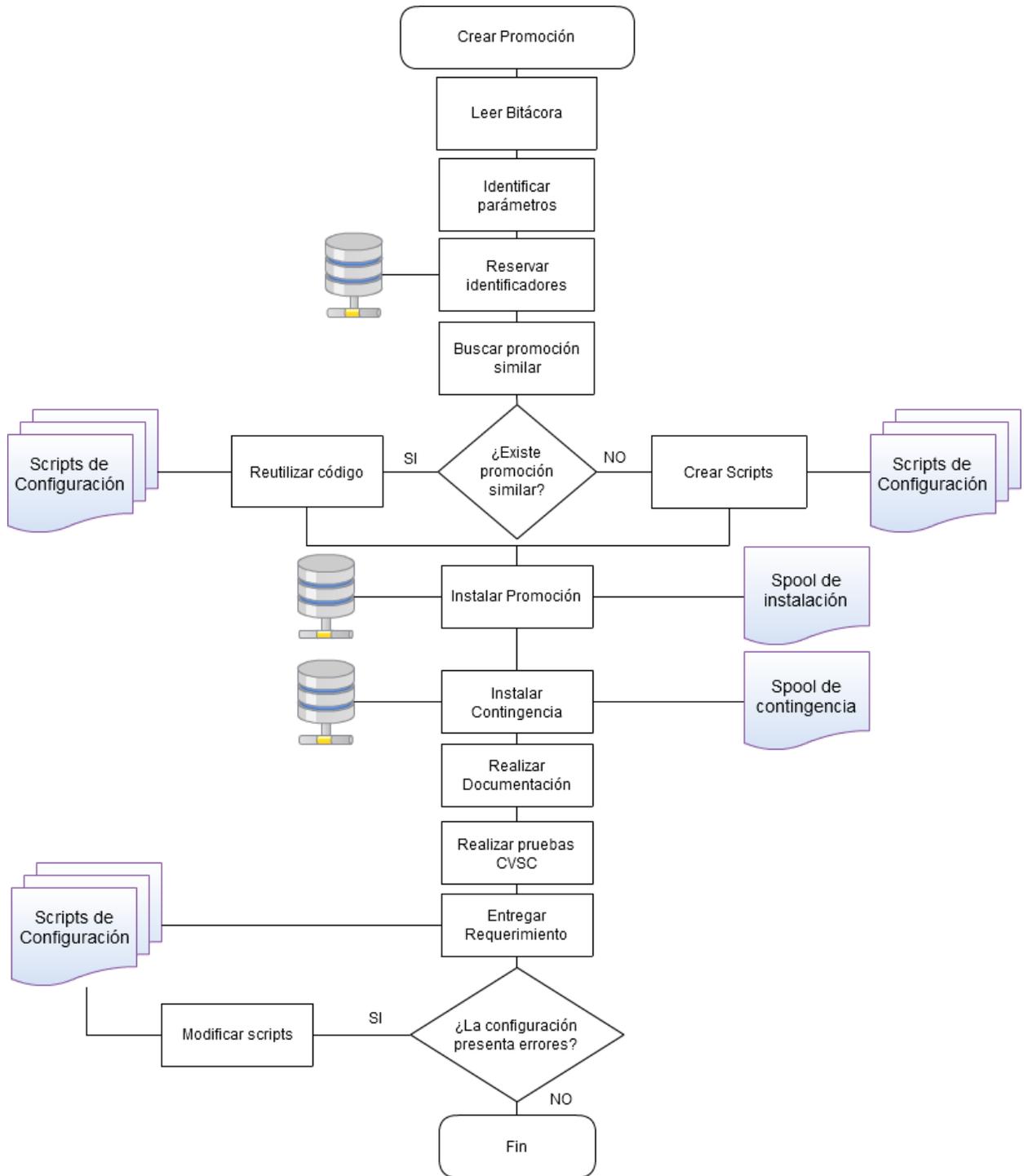


Figura 2: diagrama de flujo: configuración de una promoción.

Cuando se llevaban a cabo todas las actividades relacionadas a la creación del requerimiento, se podían presentar ciertos inconvenientes, tales como:

- Utilizar identificadores reservados por otros desarrolladores.

- Omitir segmentos de código necesarios para la creación de la promoción.
- Cometer errores en el desarrollo del código.
- No realizar las pruebas adecuadas al momento de desarrollar el código.

Dichos inconvenientes, generaban problemas en función del tiempo, entre los cuales destacaban:

- Uso excesivo de tiempo en identificar el tipo de promoción a desarrollar
- Pérdida de tiempo al realizar pruebas por problemas en el desarrollo
- Pérdida de tiempo en la búsqueda de identificadores
- Los problemas encontrados en el área de pruebas generaban incidencias que afectaban al equipo de desarrollo.
- Se debía invertir tiempo en corregir las incidencias detectadas luego de haber entregado el requerimiento al equipo de pruebas.

Un desarrollador, por lo general, requiere de cuatro (4) a cinco (5) días para realizar una promoción y entregarla al equipo de pruebas. Además de tiempo utilizado para corregir las posibles incidencias recibidas luego de entregar el requerimiento.

Los inconvenientes antes mencionados, traían como consecuencia que el desarrollador realizara la entrega de su requerimiento luego de la fecha estipulada, ya sea por errores en la configuración o el tiempo implementado en las actividades descritas. Este problema a su vez generaba mayor retraso en otros equipos de trabajo, que deben manipular el requerimiento luego de esta entrega. Por este motivo, en el equipo de la Gerencia de Configuración de Servicios surge la necesidad de desarrollar una aplicación para sistematizar el proceso de configuración de promociones.

### **1.3. Justificación**

Este Trabajo Especial de Grado surgió ante la necesidad de optimizar la gestión y el proceso de configuración de las promociones de la plataforma de prepago en la empresa Telefónica S.A. Se desarrolló una aplicación que permite a los desarrolladores realizar los requerimientos asignados de forma rápida y sencilla.

Mediante el llenado de formularios, el usuario puede indicar todas las condiciones solicitadas por el negocio, sin la necesidad de desarrollar código bajo ningún lenguaje de programación. Durante el llenado de esta información, esta nueva herramienta realiza una búsqueda de los identificadores disponibles para ser utilizados en la promoción, ahorrando un trabajo previo al usuario.

Adicional al proceso de configuración de scripts, la aplicación permite al usuario ejecutar estos archivos en la base de datos y obtener un registro de ello. Esta opción ahorra al desarrollador un tiempo considerable en la gestión de las pruebas del requerimiento.

Como valor agregado, la herramienta contiene un repositorio de información, que permite al usuario guardar las promociones realizadas, con la finalidad de ser utilizadas en futuros desarrollos.

La aplicación contribuye de forma importante en la gestión de este tipo de requerimientos, debido a que disminuye de forma significativa la gestión de todo el proceso, además de reducir al mínimo los errores que surgen durante el desarrollo.

#### **1.4.Objetivo General**

Desarrollar una aplicación web para la gestión del proceso de configuración de las distintas promociones otorgadas en la plataforma de clientes de prepago de la empresa de telecomunicaciones Telefónica S.A.

#### **1.5.Objetivos específicos**

Los objetivos específicos considerados para dar cumplimiento al objetivo general se enumeran a continuación:

1. Identificar requerimientos funcionales y no funcionales de la aplicación.
2. Realizar el análisis de factibilidad para la selección de una arquitectura web que responda a los requerimientos de la aplicación.
3. Diseñar la interfaz usable de la aplicación para cada una de sus funcionalidades.
4. Integrar la aplicación al sistema de Base de Datos de desarrollo actual de Telefónica S.A.

5. Crear el modelo lógico y físico de una base de datos local para el almacenamiento de las configuraciones del usuario.
6. Aplicar pruebas de funcionalidad y usabilidad sobre la aplicación.

### **1.6. Antecedentes**

La gestión de requerimientos utilizada actualmente dentro de la Gerencia de Configuración de Servicios de Telefónica S.A, incluye el uso de una herramienta llamada Dimensions, creada por la empresa Serena. Esta aplicación permite gestionar todas las entregas realizadas y almacena todos los cambios hechos en los archivos entregados. De forma adicional, permite copiar estos archivos en nuevas entregas, lo que permite al usuario reutilizar el código configurado en promociones anteriores. De esta forma, el usuario debe buscar los archivos que cumplan con las condiciones requeridas para luego adaptarlo a la nueva entrega. En su defecto, debe realizar el desarrollo de la promoción desde cero.

No existen actualmente aplicaciones que permitan realizar los scripts de configuración de algún requerimiento, además de ejecutarlos y almacenar un registro de esta actividad.

### **1.7. Alcance**

El alcance de este proyecto es el desarrollo de una aplicación web que permita automatizar la configuración de promociones en la plataforma de prepago, abarcando las siguientes líneas de negocio:

- Telefonía Móvil
- Telefonía Fija

Es importante destacar que en Telefónica S.A Venezuela el grupo de clientes de prepago se encuentra en una plataforma distinta a los clientes de postpago, motivo por el cual este trabajo de investigación solo abarcará el primer grupo antes mencionado.

La aplicación web a desarrollar, incluirá la realización de las siguientes tareas:

- Desarrollar todos los scripts necesarios para la creación de una promoción determinada
- Indicar todos los identificadores reservados para el desarrollo de una promoción
- Instalar en la base de datos de desarrollo todos los elementos requeridos.

- Generar informes de prueba sobre la instalación de los scripts desarrollados.
- Disminuir el tiempo invertido en el desarrollo de una promoción.
- Minimizar los errores generados durante el desarrollo y las pruebas del requerimiento.
- Proporcionar la seguridad suficiente para tener acceso a las bases de datos de desarrollo.

## **1.8. Metodología de Desarrollo**

Este TEG fue desarrollado aplicando Scrum, debido a que esta metodología de Gestión de Trabajo se adapta fácilmente a los cambios que puedan ocurrir durante el desarrollo del proyecto. Para Fowler (2005), SCRUM se concentra en gestionar los aspectos relacionados al desarrollo de software, dividiendo el desarrollo del mismo en iteraciones de 30 días y monitoreando su evolución por medio de reuniones diarias. Esta metodología no pone mucho énfasis en las prácticas de ingeniería de software, por lo que usualmente se complementa con otras metodologías, tales como XP.

Para Palacio (2014), el marco Técnico de Scrum está formado por Roles, Elementos y Eventos. Dichos eventos se presentan en un ciclo de vida iterativo llamado Sprint, que establece un período de tiempo fijado previamente. El Sprint es clave en el desarrollo del proyecto, dado que al final de cada Sprint debe presentarse una parte del producto terminada y operativa.

### **1.8.1. Roles Scrum**

**Product Owner:** Es considerado el jefe del proyecto. Entre sus funciones está ordenar según la prioridad que considere el Product Backlog, además de añadir, modificar y eliminar tareas según considere. Debe negociar con los desarrolladores en que momento debe realizarse cada tarea. Es el responsable del valor del producto.

Quien asume este rol, debe conocer perfectamente el entorno de negocio del cliente, la visión del producto, las necesidades concretas del proyecto, las posibilidades y el plan de inversión que este presenta, así como el retorno esperado de la inversión realizada.

**Scrum Team:** Es el nombre que adopta el grupo de desarrolladores.

Para Palacio (2014), es recomendable que este grupo tenga entre 4 y 8 personas, de forma que se pueda mantener una comunicación directa y se eviten roces habituales en una dinámica

de grupo. El grupo debe considerarse un equipo multifuncional, en el sentido que trabajen de forma solidaria y la responsabilidad sea compartida, a pesar de que algunos miembros sean especialistas en áreas concretas. El equipo debe trabajar de forma cohesionada y organizada, siguiendo un ambiente de respeto, colaboración y conocimiento del proyecto y el método de trabajo, manteniendo como propósito el conseguir el mayor valor posible para la visión del cliente. El equipo debe considerarse en capacidad de autoorganizarse y tomar sus propias decisiones sobre el desarrollo.

Scrum Master: para Palacio (2014), este es el nombre que adopta uno de los programadores del Scrum Team, el cual tiene como función ser el moderador de reuniones diarias que se realizan dentro del equipo. Debe proporcionar la asesoría y formación necesaria tanto al Scrum Team como al Product Owner. Aunque este miembro del equipo presenta un mayor liderazgo que el resto, debe colocarse siempre al servicio y en ayuda del equipo.

### 1.8.2. Elementos básicos de SCRUM

Gracia (2006) enumera los elementos básicos dentro de la metodología SCRUM, los cuales se presentan a continuación:

- Una lista con las funcionalidades de la aplicación ordenadas de mayor a menor importancia. Esta lista se llama "**Product Backlog**". No hace falta que esta lista contenga todas las funcionalidades inicialmente. Las funciones de la lista deben ser algo "tangible", es decir, que si nuestro programa implementa una de esas funciones, un usuario que use nuestro programa puede ver que esa función está implementada. Estas funciones "cuadran" muy bien con las historias de usuario de la programación extrema.
- De la lista anterior, se toman las primeras funcionalidades, se descomponen en tareas y son anotadas en una lista que se llama "**Sprint Backlog**". Estas tareas serán realizadas en el siguiente mes.

En la figura 3, se puede observar el flujo utilizado en SCRUM para gestionar los proyectos. Se puede observar como una tarea se toma del Sprint Backlog y la misma pasa a ser una funcionalidad luego de terminar una iteración.

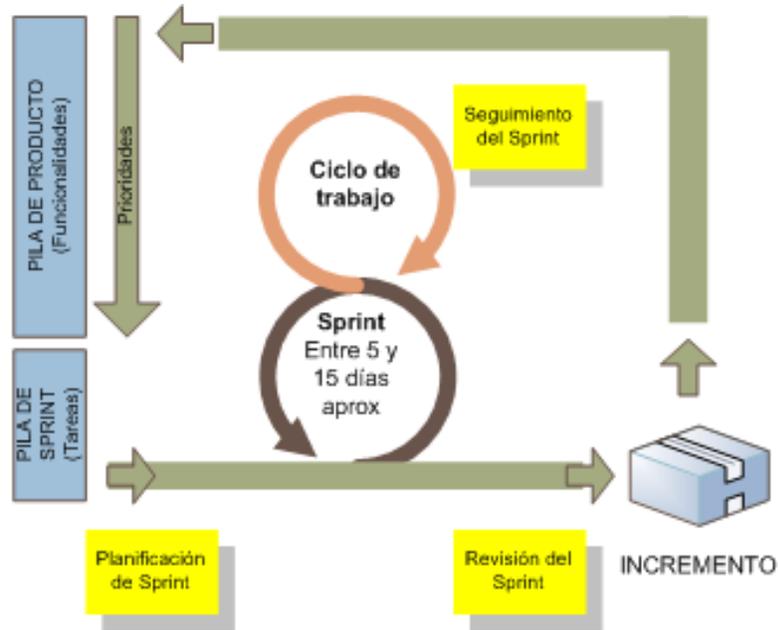


Figura 3: Flujo de SCRUM. Fuente: Hernández (2012).

### 1.8.3. Ciclo de vida de SCRUM

Para Gracia (2006), una vez definido el Product Owner, el Scrum Team y el Product Backlog, se puede dar comienzo al proyecto. En primer lugar, se celebra una reunión llamada **Sprint Planning Meeting**. Esta reunión debe celebrarse al comienzo de cada iteración, la cual debe tener una duración de 30 días o el tiempo que decidan establecer. Luego de definir el tiempo, se toman las tareas establecidas en el Product Backlog una a una y se discute si pueden ser realizadas durante una iteración. De ser necesario, una tarea puede descomponerse en varias subtareas. Es importante tomar todas las tareas que el Scrum Team se considere capaz de realizar durante una iteración. El **Sprint Planning Meeting** es el momento de negociar todos los aspectos señalados.

Una vez terminada la reunión, todas las tareas señaladas para la siguiente iteración son agregadas a un **Sprint Backlog**. Al finalizar el tiempo establecido, es deber del Scrum Team entregar al Product Owner una versión programada de todas las tareas agregadas al **Sprint Backlog**.

Al finalizar el tiempo establecido, se realiza una reunión llamada **Sprint Review**, en la que se muestra la versión programada según lo establecido en el Sprint Backlog. Se realizan opiniones y posibles mejoras para los próximos Sprints. Inmediatamente, se realiza otra reunión llamada **Sprint Retrospective**, en la que deben discutirse todos los aspectos posibles a mejoras para las futuras iteraciones y como solucionar los problemas que puedan presentarse.

Al final de cada iteración, se realiza un nuevo Sprint Planning Meeting, donde se retoma el Product Backlog y se asignan nuevas tareas al Sprint Backlog. Este proceso debe repetirse hasta que el Product Backlog no tenga tareas por realizar.

Luego de conocer los roles, los elementos y el ciclo de vida de la metodología, es importante seguir un conjunto de reglas para poder aplicarla.

#### **1.8.4. Reglas básicas de SCRUM**

Según Gracia (2006), es importante cumplir las siguientes reglas para aplicar de forma correcta la metodología SCRUM:

- Una vez que se pasan las tareas más prioritarias del "Product Backlog" al "Sprint Backlog", estas no se pueden cambiar, esto quiere decir, que el trabajo de un mes queda fijado. Esta es la regla más importante de todas.
- Al final del mes, este periodo se le llama "Sprint", se tiene que tener un ejecutable con las funcionalidades del "Sprint Backlog".
- Todo el mundo puede añadir funcionalidades al "Product Backlog", pero sólo una persona puede ordenarlo. A esta persona se le denomina "Product Owner". Es el responsable del producto final.
- Cada día se hace una reunión de menos de 15 minutos, en la que se reúne el Scrum Team y el Scrum Master, en la que cada miembro del equipo expone sólo los siguientes temas:
  - Trabajo realizado el día anterior
  - Trabajo previsto a realizar el día de hoy.
  - Necesidades o impedimentos a eliminar para poder realizar el trabajo.

Estas reuniones se realizan con la finalidad de solucionar problemas o eliminar los obstáculos que se presentan diariamente a los programadores.

Entre las prácticas aplicadas en SCRUM, en este trabajo especial de grado se hizo uso de las iteraciones, permitiendo la creación de prototipos funcionales al finalizar cada una de estas. También se utilizó el Product Backlog para definir todas las funcionalidades a realizar antes de comenzar el proyecto y del Sprint Backlog para seleccionar las tareas a desarrollar en cada iteración. Es importante el uso de las reglas básicas de Scrum, con el objeto de realizar un análisis constante sobre el desarrollo del proyecto.

Una vez establecido el contexto de investigación, los antecedentes, los objetivos y la justificación, se procedió a desarrollar una solución al problema identificado, utilizando herramientas, lenguajes y algunas prácticas descritas en el siguiente capítulo.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

En el presente capítulo se exponen las teorías utilizadas en el desarrollo de este Trabajo Especial de Grado, además de los lenguajes, aplicaciones y frameworks que sirvieron para dar implementar la aplicación.

#### **2.1. Promociones Movistar**

Entre los beneficios otorgados por Telefónica S.A a sus clientes en Venezuela, se pueden encontrar las promociones. Para Mesia y Moncada (2012), las promociones que ofrece la empresa otorgan ciertos beneficios a los clientes, entre los cuales se tienen:

- Bonos en minutos o segundos.
- Bonos en mensajes.
- Bonos de navegación (MB)
- Bonos de saldo.
- Descuentos sobre la renta de algún servicio.
- Línea gratis al momento de comprar un equipo.

Mesia y Moncada (2012), nos indican que las promociones pueden separarse en dos grupos:

- Captación: este tipo de Promociones están orientadas a captar nuevos clientes. En general, otorgan beneficios en la renta del plan o de los servicios que el cliente debe tener en el momento de su activación inicial. En algunos casos, se otorga la línea de manera gratuita.
- Fidelización: en este tipo de promociones se presentan casos muy dinámicos, ya que el cliente puede cambiar de estatus en el tiempo. Existen distintos escenarios que pueden plantearse según el estatus del cliente, que permiten al usuario plantear distintas estrategias al momento de plantear una promoción. Los escenarios más comunes para activar este tipo de promociones son:
  - Clientes Sin Renta.
  - Clientes Con Renta y Clientes Sin Renta Con Sva.
  - Clientes que no tengan activo algún Sva de Datos y o mensajería.
  - Clientes que posean planes con rentas Inferiores a un monto estimado por el negocio.

Una vez un cliente cumpla con alguno de los escenarios descritos, se le activa de manera automática la promoción que haya sido modelada para dicho propósito. Las promociones son aplicadas en las diferentes líneas de negocios, que son:

- Telefonía Móvil
- Telefonía Fija
- Internet Móvil
- Tv Digital.

Todas las promociones otorgadas por Telefónica S.A son creadas tanto para clientes prepago como para clientes postpago. Sin embargo, estos grupos de clientes se manejan en plataformas separadas, por lo que el proceso de configuración de cada método de pago se realiza de forma distinta y por diferentes equipos de desarrollo.

## **2.2. Lenguajes de Programación**

Para Ruiz (2001), los lenguajes de programación son lenguajes formales, que son diseñados concienzudamente con el fin de ser interpretados por un computador. Un lenguaje de programación debe ser capaz de cumplir ciertos requisitos, tales como:

- a) Para cada problema es posible programar una solución.
- b) El lenguaje puede ser implementado en cualquier computador.

Los lenguajes de programación se han desarrollado desde la década de 1950, inspirando a la creación de sucesivos lenguajes a lo largo de la historia. Este proceso de desarrollo ha venido creando paradigmas en la programación, aunque no evita que otros nuevos sean diseñados en el futuro. Entre ellos, destaca la programación orientada a objetos.

Según Ruiz (2001), el paradigma orientado a objetos se basa en los conceptos de objetos y clases de objetos. Un objeto es una variable equipada con un conjunto de operaciones que le pertenecen o están definidas para ellos.

Una de las bondades importantes de los lenguajes orientados a objetos es que las definiciones de los objetos pueden usarse una y otra vez para construir múltiples objetos con las mismas

propiedades o modificarse para construir nuevos objetos con propiedades similares pero no exactamente iguales.

En este Trabajo Especial de grado, se implementaron las siguientes tecnologías de desarrollo para realizar la aplicación. La mayoría de las tecnologías fueron utilizadas por solicitud de Telefónica S.A, con motivo de adaptarse a sus estándares en las herramientas que la empresa maneja.

### **2.3. Microsoft .NET**

Para Charre y Serrano (2002), .Net es un plan de Microsoft para realizar un modelo de computación distribuida mediante las redes de internet. Está basado en estándares establecidos como XML y HTML y tiene como objetivo aprovechar la infraestructura de internet, redes y protocolos, para transmitir servicios a cualquier punto, sin dar importancia al hardware, sistema operativo o lenguaje que utilizado.

En este modelo, no existe necesariamente una arquitectura cliente-servidor, sino un conjunto de sistemas que ofrecen y consumen servicios, que se ejecutan en el lugar donde se considere más lógico hacerlo.

Para Microsoft (2015), .NET es un framework, un entorno de ejecución administrado que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: el entorno en tiempo de Ejecución de Lenguaje Común (CLR, por sus siglas en inglés, Common Language Runtime), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias aplicaciones.

Los objetivos establecidos por Microsoft (2015) para el diseño de .NET Framework son los siguientes:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.

- Proveer un entorno de ejecución de código que minimiza los conflictos en el despliegue y versionado de software.
- Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceros desconocidos o que no son de plena confianza.
- Procurar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.
- Brindar al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en la Web.
- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

### **2.3.1. Entorno en tiempo de Ejecución de Lenguaje Común**

Microsoft (2015) considera el entorno en tiempo de Ejecución de Lenguaje Común (CLR, por sus siglas en inglés, Common Language Runtime), el motor en tiempo de ejecución de .NET, se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la comunicación remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que promueven su seguridad y solidez.

El entorno administrado del motor en tiempo de ejecución elimina muchos problemas de software comunes. Entre estos se pueden mencionar el control automático de la disposición de los objetos, la administración de referencias a éstos y su liberación cuando ya no se utilizan. Esta administración automática de la memoria soluciona los dos errores más comunes de las aplicaciones: la pérdida de memoria y las referencias no válidas a la memoria.

CLR permite a los desarrolladores crear aplicaciones en el lenguaje que prefieran y seguir sacando todo el provecho del motor en tiempo de ejecución, la biblioteca de clases y los componentes escritos en otros lenguajes por otros colegas. El proveedor de un compilador puede elegir destinarlo al motor en tiempo de ejecución. Los compiladores de lenguajes que se destinan a .NET Framework hacen que las características de .NET Framework estén disponibles

para el código existente escrito en dicho lenguaje, lo que facilita enormemente el proceso de migración de las aplicaciones existentes.

### **2.3.2. Biblioteca de clases de .NET Framework**

Microsoft (2015) considera la biblioteca de clases de .NET Framework como una colección de tipos reutilizables, integrados estrechamente con CLR. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones.

La biblioteca de clases de .NET Framework permite realizar diversas tareas de programación comunes, como son la administración de cadenas, recolección de datos, conectividad de bases de datos y acceso a archivos. Además de estas tareas habituales, la biblioteca de clases incluye tipos adecuados para diversos escenarios de desarrollo especializados.

Según Charte y Serrano (2002), todas las clases de la biblioteca pueden ser utilizadas desde cualquier lenguaje, marcando diferencia con otros, ya que cada uno contiene su propio modelo de componentes o biblioteca específica, que es incompatible por regla con la de otros lenguajes.

### **2.4.C Sharp y .Net Framework**

Para Microsoft (2016), C# es un lenguaje orientado a objetos con seguridad de tipos que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Se puede utilizar C# para crear aplicaciones cliente de Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, entre otros. Visual C# proporciona un editor de código avanzado, cómodos diseños de interfaz de usuario, depurador integrado y numerosas herramientas más para facilitar el desarrollo de aplicaciones basadas el lenguaje C# y .NET Framework.

#### **2.4.1. Lenguaje C#**

Microsoft (2016), considera que la sintaxis de C# es muy expresiva, por lo que puede ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales

como tipos de valor que admiten valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos. Admite iteradores, que permiten a los implementadores de clases definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente. El lenguaje de consultas integrado (LINQ, por sus siglas en inglés: Language-Integrated Query) convierte la consulta fuertemente tipada en una construcción de lenguaje de primera clase.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave override como medio para evitar redefiniciones accidentales. En C#, una estructura es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.

Además de estos principios básicos orientados a objetos, Microsoft considera que C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

- Firmas de métodos encapsulados denominadas delegados, que habilitan notificaciones de eventos con seguridad de tipos.
- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.
- Lenguaje de consultas integrado (LINQ) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos.

El proceso de compilación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los

métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos.

#### **2.4.2. Arquitectura de la plataforma .NET Framework**

Microsoft (2016) asegura que los programas de C# se ejecutan en .NET Framework, un componente que forma parte de Windows y que incluye un sistema de ejecución virtual denominado CLR y un conjunto unificado de bibliotecas de clases.

El código fuente escrito en C# se compila en un lenguaje intermedio. Este código y recursos tales como mapas de bits y cadenas se almacenan en disco en un archivo ejecutable denominado ensamblado, cuya extensión es .exe o .dll generalmente. Un ensamblado contiene un manifiesto que proporciona información sobre los tipos, la versión, la referencia cultural y los requisitos de seguridad del ensamblado.

Cuando se ejecuta un programa de C#, el ensamblado se carga en CLR, con lo que se pueden realizar diversas acciones en función de la información del manifiesto. A continuación, si se cumplen los requisitos de seguridad, se realiza una compilación en el momento para convertir el código de lenguaje intermedio en instrucciones máquina nativa. El entorno también proporciona otros servicios relacionados con la recolección de elementos no utilizados, el control de excepciones y la administración de recursos. El código ejecutado se denomina algunas veces "código administrado", en contraposición al "código no administrado" que se compila en lenguaje máquina nativo destinado a un sistema específico. En la figura 4 se muestran las relaciones en tiempo de compilación y tiempo de ejecución de los archivos de código fuente de C#, las bibliotecas de clases de .NET Framework, los ensamblados y CLR.

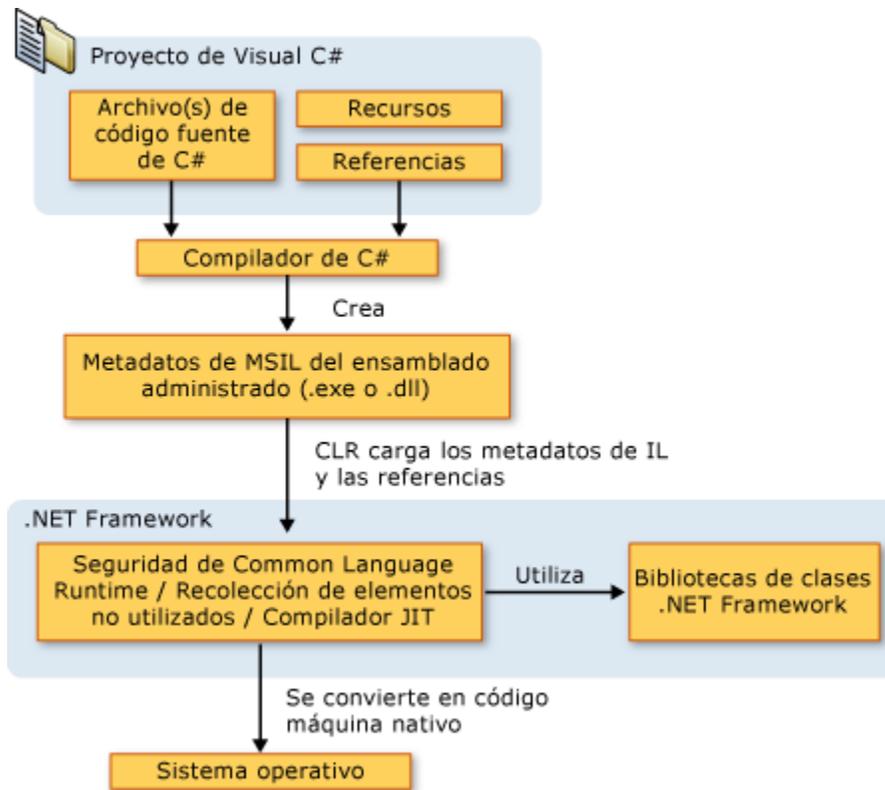


Figura 4. Compilación de un código en C#. Fuente: Microsoft (2016).

La interoperabilidad del lenguaje es una característica clave de .NET Framework. Como el código de lenguaje intermedio generado por el compilador de C# cumple la especificación de tipos común (CTS), este código generado en C# puede interactuar con el código generado en las versiones .NET de Visual Basic, Visual C++ o cualquiera de los más de veinte (20) lenguajes conformes a CTS. Un único ensamblado puede contener varios módulos escritos en diferentes lenguajes .NET, y los tipos admiten referencias entre sí como si estuvieran escritos en el mismo lenguaje.

## 2.5.Base de Datos

Según Ortiz (2000), una base de datos es fundamentalmente un repositorio de información pasivo: las operaciones sobre sus datos son realizadas por aplicaciones externas que han de ser invocadas de forma explícita.

El principal objetivo de una base de datos es, como su nombre indica, almacenar grandes cantidades de datos organizados siguiendo un determinado esquema o "modelo de datos" que facilite su almacenamiento, recuperación y modificación.

En base a los aportes de Pinto (2004), las bases de datos son el conjunto de informaciones almacenadas en un soporte legible por ordenador y organizadas internamente por registros (formado por todos los campos referidos a una entidad u objeto almacenado) y campos (cada uno de los elementos que componen un registro). Permite recopilar todo tipo de información, con la finalidad atender las necesidades de un amplio grupo de usuarios. Su tipología es variada y se caracterizan por una alta estructuración y estandarización de la información.

### **2.5.1. Lenguaje de Base de Datos**

Elmasri y Navathe (2002), consideran que pueden encontrarse los siguientes lenguajes dentro de una arquitectura de tres niveles:

Lenguaje de Definición de Datos (LDD, por sus siglas): se utiliza para definir el esquema conceptual e interno de la Base de Datos y cualquier correspondencia entre ambos. Si el SGBD mantiene una clara separación entre estos niveles, entonces solo será utilizado para definir el esquema conceptual.

Lenguaje de Definición de Almacenamiento (LDA, por sus siglas): se utiliza para definir el esquema interno. Este lenguaje también puede utilizarse para establecer las relaciones con el esquema conceptual.

Lenguaje de Definición de Vistas (LDV, por sus siglas): es utilizado para especificar las vistas de usuario y sus relaciones con el esquema conceptual.

Lenguaje de Manipulación de Datos (LMD, por sus siglas): nos permite manipular los datos dentro de las tablas mediante operaciones. Las operaciones más comunes son: Inserción, recuperación modificación y eliminación de los datos.

Existen dos tipos principales de LMD:

LMD de alto nivel, de no procedimiento u orientados a conjunto: Se utilizan para especificar operaciones complejas de base de datos de forma concisa. Puede encontrarse embebido dentro de un lenguaje de propósito general o ser ejecutado desde un terminal.

LMD de bajo nivel, de procedimiento u orientados a registros: Se utiliza para recuperar registros u objetos individuales de la base de datos y los procesa por separado. Debe encontrarse embebido dentro de un lenguaje de propósito general.

En los SGBD actuales, los lenguajes antes mencionados no se consideran lenguajes diferentes. Se utilizan como un lenguaje integrado que permite definir esquemas conceptuales, vistas, manipular datos y definir su almacenamiento (SQL es uno de estos lenguajes).

## **2.5.2. Modelos de Base de Datos**

Para Elmasri y Navathe (2002), un modelo de datos es una colección de conceptos que sirven para describir una estructura de la Base de Datos. La estructura de la base de datos nos permite definir los tipos de datos, los vínculos y las restricciones que deben cumplirse entre los ellos. Entre estos, destaca el modelo de datos relacional.

### **2.5.2.1. Modelo de datos relacional**

Para Ortiz (2000), un modelo de datos relacional está basado en el concepto de relación, siendo una relación un conjunto de n-tuplas. Una tupla puede relacionar tanto entidades como interrelaciones N:M.

Las relaciones también pueden ser vistas como tablas, en las que cada tupla es una fila de la tabla. Por otra parte, los nombres de las columnas de la tabla son los atributos.

Un SMDBD ofrece la posibilidad de interrelacionar dos o más relaciones existentes en una base de datos. De hecho, es ésta la facultad que dota de mayor potencia al modelo. Ortiz destaca que todas las interrelaciones se realizan mediante la comparación de valores, ya sean éstos identificadores de objetos o atributos de estos objetos. Para que esto sea posible, es condición indispensable que los valores a comparar pertenezcan al mismo dominio.

En la figura 5, se pueden observar las relaciones entre tres tablas mediante sus atributos.

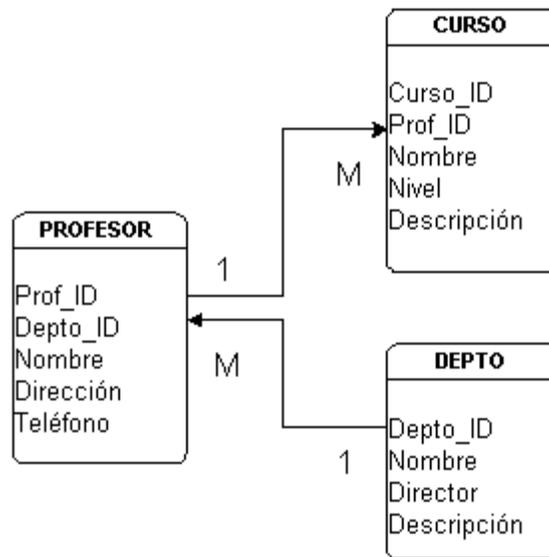


Figura 5: Ejemplo de disposición relacional. Fuente: Ortiz (2000).

Existen tres tipos de interrelaciones, que mediante la comparación de valores permiten acceder a toda la información sin crear redundancia de datos:

- **Uno a muchos (1:N):** Un registro en la tabla A puede tener muchos registros en la tabla B. Los registros de la tabla B solo pueden estar relacionados a un registro de la tabla A.
- **Muchos a muchos (N:M):** Un registro en la tabla A puede tener muchos registros en la tabla B y los registros de la tabla B pueden estar relacionados con muchos registros de la tabla A. Esta relación generalmente se logra mediante una tabla intermedia.
- **Uno a uno (1:1):** Un registro en la tabla A puede tener un solo registro en la tabla B. Esta relación es poco frecuente, ya que podrían combinarse los atributos en una sola tabla. Generalmente se utiliza para reducir el número de campos de una tabla muy grande.

### 2.5.3. Sistema Manejador de Base de Datos

Según Elmasri y Navathe (2002), un Sistema Manejador de Bases de Datos (SMBD) es una colección de programas que permiten a los usuarios crear y mantener una base de datos. Además, lo consideran un sistema de software que facilita la definición, construcción y manipulación de las bases de datos. Estas funciones se pueden detallar de la siguiente forma:

Definición: consiste en especificar los tipos de datos, las estructuras y las restricciones que se van a aplicar a los datos a almacenar.

Construcción: es el proceso de almacenar los datos en un medio de almacenamiento controlado, por medio del SMBD.

Manipulación: son aquellas funciones que permiten consultar la base de datos, actualizar los cambios en la base de datos y generar informes a partir de los datos.

### **2.5.3.1. Tipos de Sistemas Manejadores de Base de Datos**

En la actualidad, una gran cantidad de compañías han aplicado su conocimiento en el desarrollo de teorías de las bases de datos. A consecuencia de esto, se pueden encontrar en el mercado una gran cantidad de formatos y tipos de Bases de datos. Entre los diseñadores de estos SMBD (por sus siglas) encontramos a Microsoft, compañía que desarrolla SQL Server y Oracle.

#### **2.5.3.1.1. Oracle**

El SMBD Oracle es un producto multiplataforma fabricado por Oracle Corporation. Según Elmasri y Navathe (2002), un servidor Oracle contiene una Base de Datos Oracle, que incluye un log y ficheros de control, además de una instancia Oracle, que consta de los procesos del sistema Oracle y el conjunto de procesos de usuario.

El sistema de Oracle utiliza el lenguaje SQL para definir y manipular los datos. También utiliza el lenguaje PL/SQL para controlar el flujo del lenguaje SQL, usar variables y gestionar el control de errores. El sistema Oracle también permite que lenguajes como Java o C puedan acceder a la data.

##### **2.5.3.1.1.1. Estructura de la Base de Datos de Oracle:**

Para Elmasri y Navathe (2002), la base de datos de Oracle posee dos estructuras primarias. Una estructura física, que representa los datos almacenados, y una estructura lógica, que representa de forma abstracta los datos almacenados. La base de datos contiene los siguientes tipos de ficheros:

- Data Files: contienen datos en sí.
- Redo Log: registran todos los cambios realizados sobre los datos. Se utilizan en un proceso de recuperación, a ejecutarse si los cambios realizados no llegan a guardarse en la memoria permanente.
- Ficheros de control: contienen información como el nombre de la base de datos, nombres de los ficheros, ubicación y marcas de tiempo. Son utilizados en procesos de recuperación de datos.
- Ficheros de traza: existen procesos de background que tienen asociados un fichero, con el objetivo almacenar el seguimiento realizado a los principales eventos ocurridos en la base de datos

#### **2.5.3.1.1.2.Características de Oracle**

Gutiérrez-Soto (2008), menciona un conjunto de características en los SMBD, entre las cuales se pueden destacar las siguientes:

- Soporte de una gran cantidad de datos.
- Soporte para el acceso simultaneo de una gran cantidad de usuarios a los datos.
- Seguridad y acceso restringido a los datos, otorgado a cada usuario según sus necesidades.
- Integridad en la estructura de la base de datos.
- Posee una estructura cliente - servidor.
- Maneja una estructura de base de datos distribuida.
- Capacidad para ser ejecutado en distintas plataformas.
- Compatibilidad con otras aplicaciones.

#### **2.5.3.1.1.3.Beneficios de Oracle**

La Corporación Oracle (2015), considera que en sus soluciones de software ofrece a sus clientes una sólida estructura de datos que permite ofrecer los siguientes beneficios:

- Seguridad. Se minimizan los posibles riesgos a la estructura o a la información.
- Estabilidad en la ejecución, con un nivel de fallos mínimo.
- Accesibilidad para cualquier usuario, agregando funcionalidades para facilitar el uso de la aplicación.
- Flexibilidad de uso para distintos fines.

- Transparencia en la gestión de los datos.
- Escalabilidad.

#### **2.5.3.1.1.4.Desventajas de Oracle**

Calle (2012), considera que a pesar del dominio casi total en el mercado de servidores empresariales, considera que este SDBD tiene las siguientes desventajas:

- Presentan un elevado costo para la formación de personal, sólo últimamente han aparecido buenas publicaciones con aspectos técnicos distintos a la instalación y administración de las bases de datos Oracle.
- El costo del producto es bastante elevado.
- Si se realiza una mala configuración sobre Oracle el sistema puede ser exageradamente lento.

La web es un medio que permite a sus usuarios buscar, enviar y recibir información con algún propósito en específico. Esta vía de comunicación ofrece grandes facilidades con una rapidez cada vez mayor. Es por esto que surgen las aplicaciones web, con la finalidad de explotar este recurso para cualquier propósito.

#### **2.5.3.1.2. Microsoft SQL Server**

Para Microsoft (2016), SQL Server es un sistema de administración y análisis de bases de datos relacionales para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos.

Microsoft SQL Server proporciona un rendimiento, una disponibilidad y una facilidad de uso innovadores para las aplicaciones más importantes. Además, ofrece capacidades en memoria en la base de datos principal para el procesamiento de transacciones en línea (OLTP) y el almacenamiento de datos, que complementan su capacidad de almacenamiento de datos en memoria e inteligencia de negocios existentes para lograr la solución de base de datos en memoria que la mencionada corporación considera como la más completa del mercado.

En su versión SQL Server 2014, este Sistema Manejador de Bases de Datos proporciona nuevas soluciones de copia de seguridad y de recuperación ante desastres, así como de arquitectura híbrida con Windows Azure, lo que permite a los clientes utilizar sus actuales conocimientos con características locales que aprovechan los centros de datos globales de Microsoft. Además, SQL Server 2014 aprovecha las nuevas capacidades de Windows Server 2012 y Windows Server 2012 R2 para ofrecer una mayor escalabilidad a las aplicaciones de base de datos en un entorno físico o virtual.

#### **2.5.3.1.2.1. Beneficios SQL Server**

Ramos (2009), destaca las siguientes características de este SMBD:

- Soporte en la ejecución transacciones.
- Escalabilidad, estabilidad y seguridad de la aplicación.
- Soporta procedimientos almacenados.
- Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos.

Según Ramos (2009), SQL Server también otorga ciertos beneficios para los desarrolladores, entre los cuales destacan:

- Evita el uso de ficheros. Permite administrar todas las bases de datos desde un entorno gráfico, otorgando gran comodidad al momento del desarrollo.
- Permite agregar y administrar bases de datos alojadas en otros servidores de la base de datos local.
- El manejo de la seguridad es bastante alto. Permite administrar permisos con un alto nivel de detalle.

#### **2.5.3.1.2.2. Desventajas de SQL Server**

Según Ramos (2009), el SMBD de Microsoft presenta las siguientes desventajas:

- Consume una gran cantidad de memoria RAM para la instalación y ejecución del software.

- Posee muchas restricciones para la ejecución de pruebas.
- La relación calidad precio es baja en comparación con otros SDBD como Oracle.

## **2.6.Aplicaciones web**

Una aplicación web es una herramienta almacenada en un servidor web, la cual no requiere ser instalada para poder utilizarse. Para acceder a ella solo debe ser necesario un navegador web y poseer acceso a internet o a la intranet donde se encuentre alojado el servidor en cuestión.

Según Luján (2002), una aplicación web es un tipo especial de aplicación cliente – servidor, donde el cliente, el servidor y el protocolo mediante el cual se comunican están estandarizados y no deben ser creados por el desarrollador.

Para Luján (2002), el protocolo mediante el cual el cliente y el servidor se comunican es HTTP, el cual forma parte de la familia de protocolos de comunicaciones TCP/IP. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores.

Según Luján (2002), un cliente web es un programa con el que el usuario interactúa con el servidor web para solicitar el envío de los recursos que desea obtener mediante HTTP.

El servidor web es un programa que se encuentra en la espera permanente de las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web. En la figura 6 se muestra la interacción entre el servidor y el cliente mediante el protocolo http, además de un conjunto de tecnologías que pueden utilizarse en cada uno.

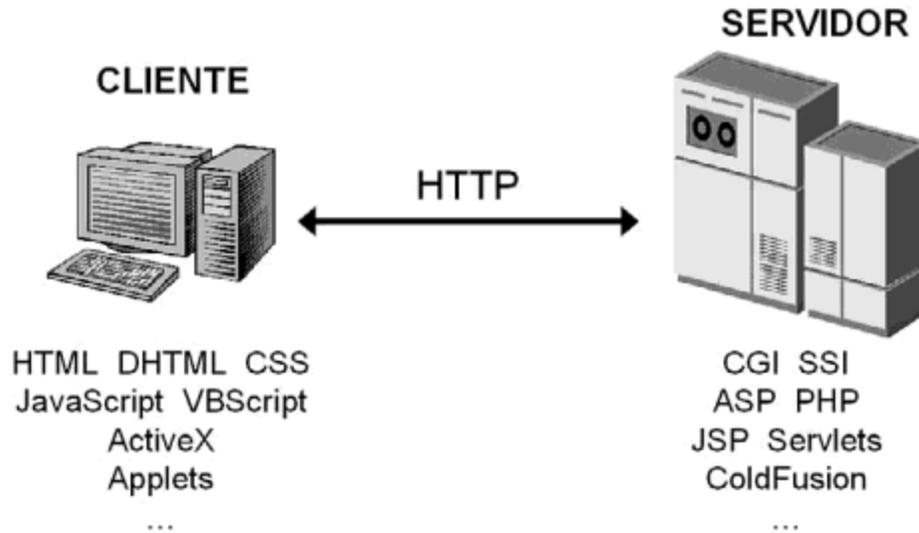


Figura 6: Tecnologías empleadas en el cliente y en el servidor web. Fuente: Luján (2002)

Con base a las definiciones anteriores, se puede definir una aplicación web como un programa o herramienta a la que un usuario accede por medio de un navegador web (Google Chrome, Mozilla Firefox, Safari, Internet Explorer, entre otros) a un servidor web, utilizando internet o una intranet empresarial como medio de comunicación entre ellas.

## 2.7.Arquitectura de Aplicaciones web

Para Gallego (2014), la arquitectura básica de una aplicación web está formada por los siguientes elementos:

- **Navegador web:** Es una herramienta que permite al cliente hacer peticiones y solicitar recursos a los servidores web. Cuando realiza una petición al servidor y recibe un recurso como respuesta, se lo muestra al usuario.
- **Servidor web:** Recibe peticiones por parte de un navegador, enviando un recurso como respuesta o un informe de error en caso de que el recurso solicitado no exista.
- **Protocolo HTTP:** Es un protocolo basado en TCP/IP, utilizado para que el navegador realice peticiones y el servidor responda.
- **HTML:** Es el formato básico de los documentos de la web. Es un formato basado en etiquetas, que permite estructurar el contenido de la página.

Según Gallego (2014), no todas las aplicaciones web utilizan la misma arquitectura, considerando que las se diferencian en lo estática o dinámica que sea la web.

Una aplicación web puede ser dinámica en el cliente y/o en el servidor. Algunos lenguajes o frameworks permiten que una aplicación web sea dinámica, ya sea del lado del cliente o del lado del servidor. Entre estas tecnologías se pueden enumerar a Javascript, Java, Microsoft .NET, PHP, Ruby on Rails, Python, Django, etc.

En base a los que explica Gallego (2014), se pueden clasificar las arquitecturas web en los siguientes tipos:

**Cliente estático – Servidor estático:** en un principio, todas las páginas web tenían esta arquitectura, con el propósito de realizar publicaciones y que el usuario tuviera acceso a esta información.

En esta arquitectura, el cliente realiza una petición al servidor mediante el protocolo http, recibiendo un fichero en respuesta por parte de este último. Para finalizar, el navegador web visualiza la página HTML con imágenes y estilos CSS.

Para Gallego (2014), con este tipo de arquitectura el servidor siempre retorna los mismos recursos, por lo que una aplicación web que utilice esta arquitectura debe considerarse una página web.

**Cliente estático – Servidor dinámico:** una Arquitectura con un Cliente estático y un servidor Dinámico es un ejemplo de arquitectura de tres capas, tal como se muestra en la figura 7:

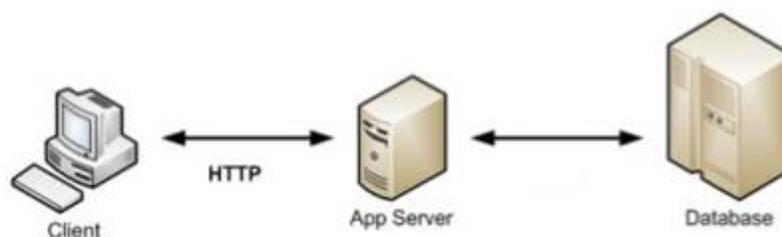


Figura 7: Arquitectura de tres capas. Fuente: Gallego (2014).

**Navegador:** Capa de presentación.

**Servidor web:** Capa de Aplicación o lógica del negocio.

**Base de Datos:** Capa de datos.

Gallego (2014) explica que un servidor web, al momento de recibir una petición, procede a devolver el contenido del disco o ejecuta un código, que a su vez genera un recurso automáticamente. Cuando se ejecuta un código, normalmente se hacen consultas a la base de datos que permiten recuperar información. Es habitual que la página HTML sea generada de forma dinámica, generando además recursos como imágenes, PDF, entre otros.

Las primeras aplicaciones web fueron creadas bajo este concepto y sigue aplicándose en la actualidad.

**Cliente dinámico – Servidor estático:** en esta arquitectura, las páginas contienen lenguajes que permitan incluir efectos gráficos para cubrir todas las necesidades del lado del cliente. El servidor de la página web es estático y envía una respuesta en base a lo solicitado por el navegador.

Si la aplicación web es dinámica del lado del cliente, aporta ventajas como la carga parcial de datos sin necesidad de recargar por completo la página y mejorando la experiencia del usuario.

**Cliente dinámico – Servidor dinámico:** esta arquitectura es la más utilizada actualmente para el desarrollo de aplicaciones web, ya que permite el uso de distintas tecnologías tanto del lado del cliente como del lado del servidor.

Además de las arquitecturas planteadas anteriormente, existe un patrón de desarrollo llamado Modelo Vista Controlador, ideal para separar el proyecto en capas, independientemente de las tecnologías que se apliquen.

### **2.7.1. Modelo Vista – Controlador**

Para Camarena, Trueba, Martínez y López (2012), el patrón ideal para el desarrollo de aplicaciones web es el modelo vista controlador (MVC). Este modelo separa el proyecto en tres elementos: la lógica de control, la lógica de negocio y la lógica de presentación, lo que permite mejorar la calidad y la mantenibilidad de la aplicación.

MacWilliams et Al. (2003) citado en Camarena et Al. (2012), consideran que el modelo MVC es el más extendido para el desarrollo de aplicaciones donde se deban manejar interfaces de

usuario, centrándose de los datos y las vistas, mientras que el controlador es el encargado de relacionarlos.

Para Ferro et Al. (2003) citado en Camarena et Al. (2012), la principal característica de esta arquitectura es aislar la vista del modelo. En la figura 8 se puede apreciar la separación de las tres capas y los componentes que la hacen funcional. La independencia entre capas hace que este modelo sea deseable para realizar proyectos de grandes dimensiones.

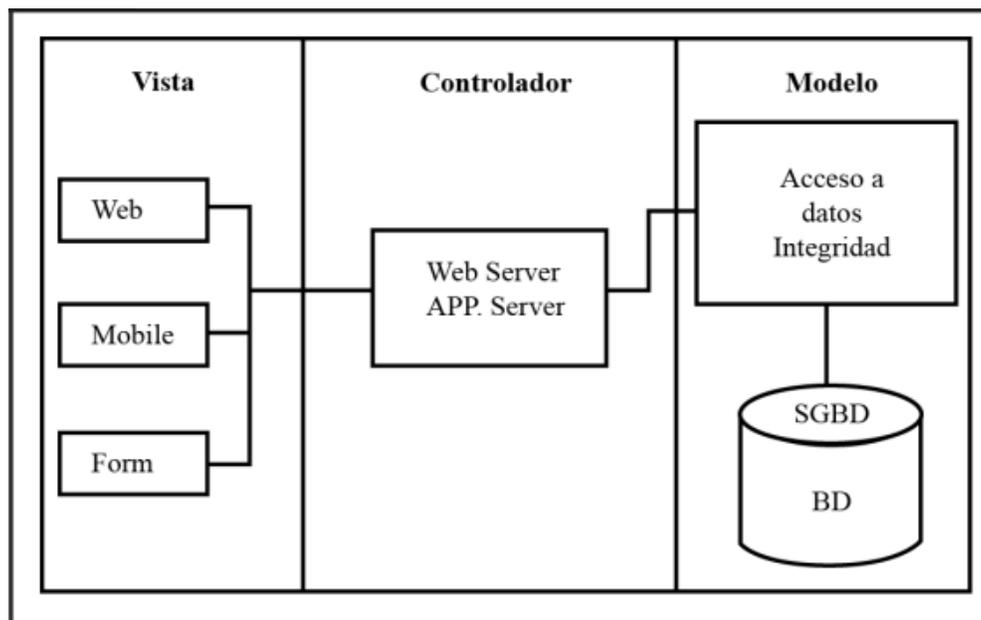


Figura 8: Modelo Vista Controlador. Fuente: Camarena et Al. (2012).

Según Camarena et Al. (2012), el patrón MVC otorga los siguientes beneficios:

- Permitir la sustitución de las interfaces de usuario de forma práctica.
- Generar componentes de las interfaces.
- Diseñar vistas simultáneas del mismo modelo.
- Aplicar fácilmente cambios de las interfaces.

Es importante destacar que el uso de este patrón implica algunos inconvenientes, tales como:

- La complejidad aumenta rápidamente.
- La vista y el modelo están muy acopladas.

## **2.8. Beneficios de las aplicaciones web**

Para Rivas (2010), las aplicaciones web aportan los siguientes beneficios:

- No necesitan ser instaladas: en algunos casos las aplicaciones pueden solicitar la descarga de algunos plugins que permita realizar funciones avanzadas.
- Fácil acceso: se puede acceder a la aplicación por medio de cualquier computadora, teniendo como único requisito una conexión a internet.
- No es necesario actualizar: Las aplicaciones web se actualizan automáticamente, en algunos casos la empresa desarrolladora notifica la actualización de la aplicación y las mejoras o correcciones que esta incluye.
- No es necesario utilizar un sistema operativo específico: Mientras la aplicación sea compatible con el navegador utilizado por el usuario, la aplicación puede ser utilizada desde cualquier sistema operativo.
- Aplicaciones gratis: Las mayoría de las aplicaciones web presentes en internet no tienen costo alguno por su uso. En su defecto, solo cobran por beneficios adicionales.

## **2.9. Desventajas de las aplicaciones web**

Para Rivas (2010), las aplicaciones web tienen las siguientes desventajas:

- Es necesaria una conexión a internet: las aplicaciones web dependen de una conexión a internet y sin la misma no puede ser utilizada.
- No tienes tus datos: las aplicaciones deben tener un nivel de seguridad alta, debido a que tus datos pueden estar expuestos fácilmente.
- Compatibilidad de idiomas: en general las aplicaciones son compatibles con pocos idiomas, lo que puede generar poca satisfacción entre sus usuarios.
- Dependencia de los plugins: algunas aplicaciones requieren del uso de plugins para su funcionamiento.
- Espacio de almacenamiento: las aplicaciones por lo general tienen un espacio limitado para su almacenamiento. Sin embargo, ofrecen un espacio extra por un costo adicional.

Luego de estudiar las tecnologías de desarrollo y los aspectos teóricos mencionados en este capítulo, se consideró viable el inicio del desarrollo de la aplicación propuesta como solución al problema.

## **CAPITULO III**

### **MARCO APLICATIVO**

En el siguiente capítulo, se describe el proceso de desarrollo del Trabajo Especial de Grado, utilizando la metodología SCRUM para llevar a cabo cada fase del proyecto y los modelos aplicados. Además, se especifica la arquitectura utilizada en la aplicación.

#### **3.1. Descripción de la aplicación de la tecnología**

Para gestionar el desarrollo del Trabajo Especial de grado, se aplicó la metodología Ágil denominada Scrum, adaptando sus principales características al equipo de trabajo y sus necesidades. La primera tarea a realizar fue definir los roles del equipo de trabajo, los elementos y los eventos a realizar durante el desarrollo del proyecto.

##### **3.1.1. Roles Scrum**

Con motivo de que solo una persona es integrante del desarrollo del proyecto, se define un solo rol para el desarrollo del mismo. Es su deber cumplir las siguientes obligaciones:

- Añadir, modificar, eliminar tareas y dar la propiedad que considere.
- Conocer perfectamente el entorno del negocio, la visión del producto, las necesidades del proyecto y sus posibilidades.
- Capacidad de auto organizarse y tomar sus propias decisiones con respecto al proyecto.
- Proporcionar la formación necesaria para el desarrollo del proyecto.

Los coordinadores de la Gerencia de Configuración de servicios actuaron como stakeholders en el desarrollo del proyecto, mostrando su opinión en cada sprint y destacando las necesidades de cada funcionalidad implementada.

##### **3.1.2. Elementos básicos de Scrum**

A continuación se muestran los elementos de la metodología que fueron utilizados para el desarrollo del proyecto:

**Product Backlog:** Es una lista de funcionalidades ordenada de mayor a menor según el orden de importancia, representando algo tangible para el usuario al momento de utilizar la aplicación. En la tabla 1 se representa el product backlog realizado para el desarrollo del proyecto.

Tabla 1: Product Backlog.

ID	HISTORIA
1	Diseño de la arquitectura de la aplicación
2	Creación de Vistas
3	Envío de Datos
4	Desarrollo de Scripts
5	Descarga de Scripts
6	Edición de promociones
7	Selector de Identificadores para una promoción
8	Consulta de identificador de mensajes
9	Ejecución de Scripts en BD Oracle
10	Creación y descarga de Spools de ejecución
11	Creación de Base de datos Local
12	Almacenamiento de promociones configuradas
13	Recuperación de promociones configuradas
14	Creación de promociones a partir de una promoción almacenada
15	Eliminación de promociones configuradas
16	Actualización de la interfaz
17	Control de sesiones

**Sprint Backlog:**

De la lista anterior, las funcionalidades fueron descompuestas en tareas y fueron agregadas a una lista llamada Sprint Backlog. Al realizar esto, se obtuvieron las tareas descritas en las tablas dos (2) y tres (3).

Tabla 2: Sprint Backlog. Primera Parte.

<b>ID</b>	<b># TAREA</b>	<b>DESCRIPCIÓN</b>
1	1	Diseño de la arquitectura de la aplicación
2	2	Creación de vistas principales
	3	Creación de CSS Básico
3	4	Creación de modelos de datos
	5	Formulario de condiciones iniciales de la promoción
	6	Formulario de datos de la promoción.
4	7	Desarrollo de querys
	8	Creación de archivos
	9	Creación de script de configuración
	10	Creación de scripts de contingencia
5	11	Creación de vista de descarga de datos
6	12	Creación de vista de edición de datos.
	13	Almacenamiento carga de datos para la ser reutilizados durante la configuración.
7	14	Establecer conexiones con la Base de datos Oracle
	15	Consultar y seleccionar identificador de promoción
	16	Almacenar identificador si el usuario lo requiere
8	17	Solicitar código de mensaje
	18	Consultar Base de datos
	19	Validar código.
9	20	Establecer conexiones con la Base de datos Oracle
	21	Establecer querys a ejecutar.
	22	Identificar parámetros a utilizar.
10	23	Creación de archivos de descarga.
	24	Creación de enlaces de descarga de archivos.

Tabla 3: Sprint Backlog. Segunda Parte.

<b>ID</b>	<b># TAREA</b>	<b>DESCRIPCIÓN</b>
11	25	Diseño de la Base de Datos
	26	Creación de tablas
	27	Creación de restricciones
12	28	Establecer conexiones con la Base de datos SQL Server
	29	Establecer queries a ejecutar.
	30	Identificar parámetros a utilizar en cada tabla
13	31	Creación de vistas de recuperación de datos
	32	Identificar promoción a recuperar.
	33	Establecer conexiones con la Base de datos SQL Server
	34	Establecer queries a ejecutar.
14	35	Carga de datos obtenidos de la recuperación
	36	Creación de vistas de reedición
15	37	Identificar Promoción a eliminar
	38	Establecer conexiones con la Base de datos SQL Server
	39	Establecer queries y parámetros a ejecutar.
16	40	Modificación de las vistas
	41	Modificación del CSS
17	42	Módulo de inicio de sesión en la aplicación

Luego de tener definidos los elementos, se dio inicio al proyecto. En primer lugar, se celebró un Sprint Planning meeting, con lo comenzó la primera iteración del proyecto. En esta primera reunión se estableció el tiempo de cada iteración en treinta (30) días cada uno, además de las establecer las tareas a realizar.

En cada Sprint Planning Meeting se seleccionaron un conjunto de tareas para ser realizadas en cada iteración, obteniendo un total de siete iteraciones para el desarrollo del proyecto.

### Primera iteración:

Para dar inicio al proyecto, se tomaron las primeras tareas del Sprint Backlog como objetivos para cumplir la primera iteración, tal como se puede observar en la tabla 4:

Tabla 4: Sprint Backlog de la primera iteración.

ID	# TAREA	DESCRIPCIÓN
1	1	Diseño de la arquitectura de la aplicación
2	2	Creación de vistas principales
	3	Creación de CSS Básico
3	4	Creación de modelos de datos
	5	Formulario de condiciones iniciales de la promoción
	6	Formulario de datos de la promoción.

Al iniciar la primera iteración, se realizó el diseño de la arquitectura de la aplicación. Para el desarrollo de esta herramienta, se decidió separar el modelo en capas. Para esto, se hizo uso del patrón de desarrollo Modelo Vista Controlador (MVC, por sus siglas). Este modelo divide el proyecto en tres elementos: la lógica de control, la lógica de negocio y la lógica de presentación, lo que permite mejorar la calidad y la mantenibilidad de la aplicación.

El uso del patrón de desarrollo MVC permitió realizar el proyecto de forma organizada, armonizando las funcionalidades que fueron realizadas. Separar el proyecto en varios elementos permite una mayor flexibilidad en el desarrollo, además de la posibilidad de reutilizar gran cantidad de código en las distintas funcionalidades aplicadas.

En la figura 9, se visualiza la arquitectura de la aplicación. Todos los archivos pertenecientes a la aplicación se encuentran en una de las tres (3) capas del modelo. En la Vista se encuentran todos los elementos que diseñan las vistas a ser mostradas en el navegador web del usuario. El modelo representa los repositorios de la aplicación, es decir, la base de datos local, desarrollada en SQL Server y la base de datos de Telefónica S.A, desarrollada en Oracle. Las funcionalidades de la aplicación son parte del controlador.

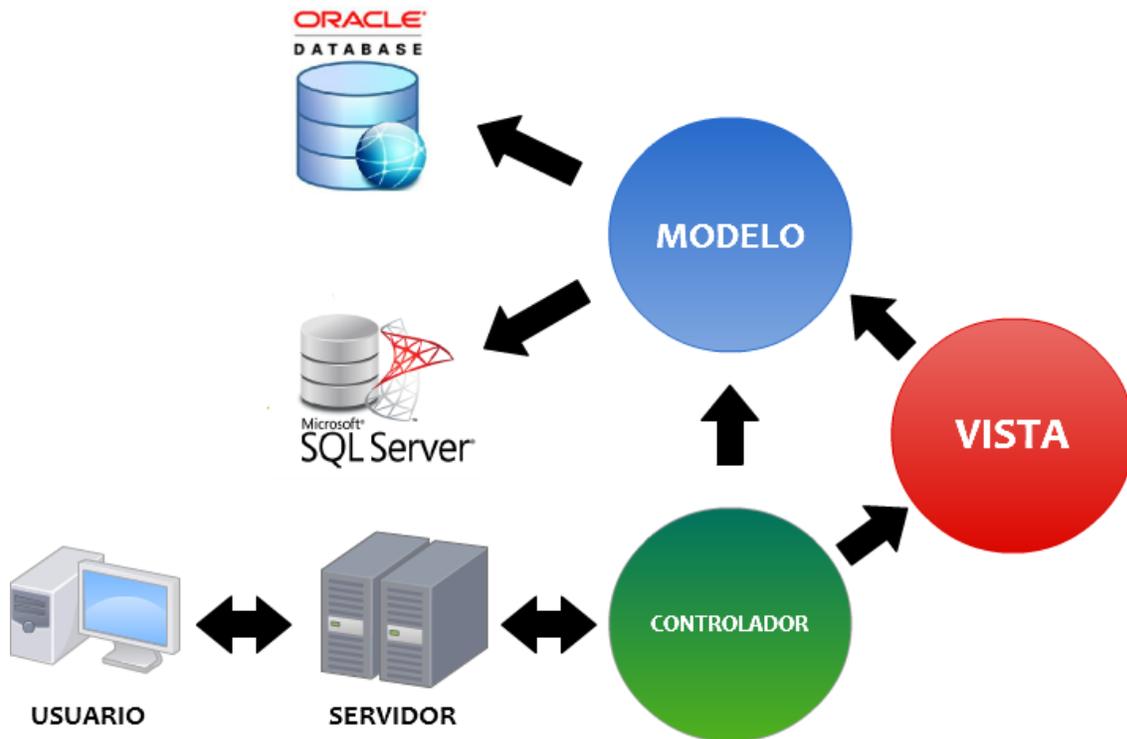


Figura 9. Arquitectura de la aplicación.

Luego de realizar el diseño de la arquitectura de la aplicación, se crearon las vistas básicas que requería la aplicación. Como segunda tarea, se aplicó un CSS para establecer un diseño de las vistas. Luego se establecieron las clases a utilizar para manipular los datos de las promociones. En la figura 10, se muestra la clase asociada a los parámetros iniciales de la promoción.

```
public class CondIniciales
{
    public int c_modelos { get; set; }
    public int c_elementos { get; set; }
    public int c_age_aut { get; set; }
    public int consulta_msj { get; set; }
    public bool solicita_cpromo { get; set; }
}
```

Figura 10. Código definido para las condiciones iniciales de la promoción.

En la figura 11 se muestran la clase desarrollada para almacenar los datos principales de la promoción:

```

public class ParamPromo
{
    public int cpromo { get; set; }
    public string nombre_promo { get; set; }
    public string dia_desde { get; set; }
    public string mes_desde { get; set; }
    public string ano_desde { get; set; }
    public string dia_hasta { get; set; }
    public string mes_hasta { get; set; }
    public string ano_hasta { get; set; }
    public bool valida_serial { get; set; }
    public string tipo_promo { get; set; }
    public string restriccion { get; set; }
    public string negocio { get; set; }
    public string tecnologia { get; set; }
    public bool op_ainic { get; set; }
    public bool op_cser { get; set; }
    public bool op_cpla { get; set; }
    public bool op_ctec { get; set; }
    public bool op_ccov { get; set; }
    public bool op_aprom { get; set; }
    public int cdiseno { get; set; }
    public string[] nombre_elemento { get; set; }
    public int[] cod_elemento { get; set; }
    public string[] tipo_elemento { get; set; }
    public int descuento { get; set; }
    public int tiempo { get; set; }
    public string duracion { get; set; }
    public bool cond_admin_status { get; set; }
    public bool cond_cierre_promo { get; set; }
    public bool cond_cond_adicional { get; set; }
    public bool cond_cplan_mercserial { get; set; }
    public int cmensaje { get; set; }
    public string descripcion_msj { get; set; }
    public string nmensaje { get; set; }
    public string[] modelo { get; set; }
    public string[] agente_autorizado { get; set; }
}

```

Figura 11. Parámetros principales de la promoción.

En las últimas dos tareas pertenecientes a esta iteración. Se desarrollaron los formularios necesarios para que el usuario introduzca todos los datos referentes a la promoción. En la figura

12 se observa una porción del código escrito en Razor .NET, mientras que la figura 13 se observa el código traducido en HTML, que es el observado por el usuario en el navegador web.

```

<tr>
  <td colspan="2">
    <p>
      @Html.Label("Líneas de Negocio:");*<br />
      @Html.RadioButtonFor(p => p.negocio, "F") Fijo<br />
      @Html.RadioButtonFor(p => p.negocio, "C") Móvil<br />
    </p>
  </td>
  <td colspan="1">
    <p>
      @Html.Label("Tecnología:");*<br />
      @Html.RadioButtonFor(p => p.tecnologia, "GSM") GSM<br />
      @Html.RadioButtonFor(p => p.tecnologia, "CDMA") CDMA<br />
    </p>
  </td>
</tr>

```

Figura 12. Fragmento de código RAZOR .NET

```

<tr>
  <td colspan="2">
    <p>
      <label for="L_neas_de_Negocio:">L&#237;neas de Negocio:</label>*<br />
      <input id="negocio" name="negocio" type="radio" value="F" /> Fijo<br />
      <input id="negocio" name="negocio" type="radio" value="C" /> Móvil<br />
    </p>
  </td>
  <td colspan="1">
    <p>
      <label for="Tecnolog_a:">Tecnolog&#237;a:</label>*<br />
      <input id="tecnologia" name="tecnologia" type="radio" value="GSM" /> GSM<br />
      <input id="tecnologia" name="tecnologia" type="radio" value="CDMA" /> CDMA<br />
    </p>
  </td>
</tr>

```

Figura 13. Fragmento de código HTML

Al finalizar esta iteración, se presentó un prototipo funcional con las vistas básicas de la aplicación, que permitió al usuario ingresar los datos en el la aplicación web. De forma adicional, se realizaron pruebas para verificar que el usuario solo pueda ingresar los datos solicitados en el formulario.

### Segunda iteración:

Las tareas señaladas en la tabla 5 fueron las implementadas en la segunda iteración:

Tabla 5: Sprint Backlog de la segunda iteración.

ID	# TAREA	DESCRIPCIÓN
4	7	Desarrollo de querys
	8	Creación de archivos
	9	Creación de script de configuración
	10	Creación de scripts de contingencia
5	11	Creación de vista de descarga de datos

Para cumplir con las tareas señaladas, se realizaron los querys necesarios para la desarrollo de una promoción de prepago de Telefónica S.A, utilizando además los datos que el usuario debe ingresar mediante los formularios creados en la primera iteración. Luego de ser desarrollados estos querys, fueron creadas las instrucciones para generar dos archivos, que deben contener todos los scripts que el usuario requiera para crear su promoción. En la figura 14 se muestra la forma en que estos archivos son generados

```
using (System.IO.StreamWriter sw = new System.IO.StreamWriter(a_config, true))
{
    sw.WriteLine("spool t2_config.txt \n \n");

    //TCONFIG_PROMOCION
    sw.WriteLine("-----TCONFIG_PROMOCION \n \n");
    sw.WriteLine("UPDATE tconfig_promocion \n SET npromocion = '" + p.nombre_promo + "', \n finicio = TO_DATE ('" + p.dia_desde + "/" +

    //TIPO_SERVICIOXPROMOCION
    sw.WriteLine("-----TIPO_SERVICIOXPROMOCION \n \n");
    sw.WriteLine("INSERT INTO ttipo_servicioxpromocion(cpromocion, iservicio) VALUES ('" + p.cpromo + "', '" + p.negocio + "') \n \n");

    //TIPO_CENTRALXPROMOCION
    sw.WriteLine("-----TIPO_CENTRALXPROMOCION \n \n");
    sw.WriteLine("INSERT INTO ttipo_centralxpromocion(cpromocion, ctipo_central) VALUES ('" + p.cpromo + "', '" + central + "');\n \n");

    //TECNOLOGIAXPROMOCION
    sw.WriteLine("-----ITECNOLOGIAXPROMOCION \n \n");
    sw.WriteLine("INSERT INTO ttecnologiaxpromocion(cpromocion, ctecnologia) VALUES ('" + p.cpromo + "', '" + tec + "');\n \n");
}
```

Figura 14: Creación del archivo t2\_config.sql

Una vez generados los archivos, se realizaron una serie de pruebas para verificar que dichos archivos no tuvieran errores. Estos archivos fueron probados mediante un SMDB en la Base de Datos de Oracle para verificar que se ejecuten correctamente.

Para finalizar la iteración, se agregó una vista que permite al usuario descargar los archivos de configuración de la promoción. Al finalizar todas las tareas, la aplicación podía generar los scripts deseados por el usuario y permitía su descarga.

### Tercera iteración:

Para cumplir con los objetivos de la tercera iteración, fueron implementadas las tareas descritas en la tabla 6:

Tabla 6: Sprint Backlog de la tercera iteración.

ID	# TAREA	DESCRIPCIÓN
6	12	Creación de vista de edición de datos.
	13	Almacenamiento carga de datos para la ser reutilizados durante la configuración.
7	14	Establecer conexiones con la Base de datos Oracle
	15	Consultar y seleccionar identificador de promoción
	16	Almacenar identificador si el usuario lo requiere
8	17	Solicitar código de mensaje
	18	Consultar Base de datos
	19	Validar código.

En esta iteración se desarrolló una vista para la edición de datos, en caso de que el usuario desee editar los valores que ingresó previamente. Para realizar esta tarea se almacenan los datos en algunas variables del código, de forma que puedan ser manipuladas fácilmente.

La siguiente tarea fue establecer la primera conexión con la base de datos Oracle, que se encuentra alojada en los servidores de Telefónica S.A. En esta conexión se realiza una consulta a la base de datos, que trae como resultado un posible identificador a ser utilizado para la promoción.

En la figura 15, se muestra el código referente a la búsqueda de un identificador de promoción:

```
public void obtener_codigo_promocion()
{
    OracleConnection conn = new OracleConnection(connString);
    conn.Open();
    OracleCommand cmd = new OracleCommand();
    cmd.Connection = conn;
    String query = "SELECT CPROMOCION FROM TCONFIG_PROMOCION WHERE NPROMOCION LIKE 'Promocion Reservada desarrollo' AND CPROMOCION>1000 AND ROWNUM<2";
    cmd.CommandText = query;
    cmd.CommandType = CommandType.Text;
    OracleDataReader reader = cmd.ExecuteReader();
    reader.Read();
    cpromo_oracle = reader.GetOracleNumber(0);
    conn.Close();
}
```

Figura 15. Buscar Código de promoción.

Para finalizar, se desarrolló una función que permite al usuario verificar la disponibilidad de un código para ser utilizado como identificador de mensajes. En la figura 16 se muestra el código que permite implementar esta funcionalidad:

```
public void consultar_codigo_mensaje()
{
    OracleConnection conn = new OracleConnection(connString);
    conn.Open();
    OracleCommand cmd = new OracleCommand();
    cmd.Connection = conn;
    String query = "SELECT COUNT(*) FROM TMENSAJE WHERE CMENSAJE="+c_in.consulta_msj;
    cmd.CommandText = query;
    cmd.CommandType = CommandType.Text;
    OracleDataReader reader = cmd.ExecuteReader();
    reader.Read();
    cmensaje_oracle = reader.GetOracleNumber(0);
    conn.Close();
}
```

Figura 16. Verificar código de mensaje.

En esta iteración no hubo necesidad de realizar mayores pruebas sobre la aplicación, dado que los identificadores obtenidos de la Base de Datos se muestran en el formulario de configuración de la promoción.

#### Cuarta iteración:

Para cumplir con la cuarta iteración fueron desarrolladas las tareas descritas en la tabla 7:

Tabla 7: Sprint Backlog de la cuarta iteración.

ID	# TAREA	DESCRIPCIÓN
9	20	Establecer conexiones con la Base de datos Oracle
	21	Establecer queries a ejecutar.

En las tareas mencionadas, se establecieron nuevas conexiones a la base de datos de Oracle. En este caso fueron utilizadas dichas conexiones para insertar los queries obtenidos en la

creación de archivos, adaptando el código y los parámetros a los métodos de C# para manipular los repositorios Oracle. Al finalizar esta iteración la aplicación permitía al usuario insertar los scripts de configuración a la base de datos.

En la figura 17, se muestra un fragmento de código que indica cómo se ejecuta el script en la base de datos de Oracle.

```
public string ejecutar_query(OracleConnection connection, OracleCommand command, string query)
{
    string resultado;
    try
    {
        command.CommandText = query;
        int rowsUpdated = command.ExecuteNonQuery();
        resultado = rowsUpdated + " fila(s) afectada(s).";
    }
    catch (Exception e)
    {
        resultado = e.ToString() + "\n Existe un error en la configuracion de esta instrucción.";
    }
    return resultado;
}
```

Figura 17. Instalar configuración.

En la figura 18, se muestra como los datos indicados por el usuario son manipulados para luego ser insertados en la Base de Datos de Oracle.

```
public string insert_query_modelos(OracleConnection connection, string modelo, string query) {
    OracleCommand command_31 = connection.CreateCommand();

    OracleParameter param_31_1 = new OracleParameter("i_cpromocion", OracleType.Number, 6, "cpromocion");
    param_31_1.Value = e.cpromo;
    command_31.Parameters.Add(param_31_1);

    OracleParameter param31_2 = new OracleParameter("i_modelo", OracleType.VarChar, 10, "cmodelo");
    param31_2.Value = modelo;
    command_31.Parameters.Add(param31_2);

    string resultado = ejecutar_query(connection, command_31, query);
    return resultado;
}
```

Figura 18. Instalación de la configuración - Insertar modelos.

Para verificar el funcionamiento de las tareas agregadas en esta iteración, fueron realizados registros de la ejecución y consultas en la Base de Datos de Oracle para comprobar que la instalación se realiza correctamente.

### **Quinta iteración:**

En esta fase se llevaron a cabo las tareas señaladas en la tabla 8:

Tabla 8: Sprint Backlog de la quinta iteración.

<b>ID</b>	<b># TAREA</b>	<b>DESCRIPCIÓN</b>
9	22	Identificar parámetros a utilizar.
10	23	Creación de archivos de descarga.
	24	Creación de enlaces de descarga de archivos.

Para finalizar el proceso de instalación en la base de datos de Oracle, se realizaron las tareas para ejecutar los scripts de contingencia. Luego de esto, se desarrollaron las instrucciones para generar los archivos que muestren evidencia del proceso de instalación y contingencia de la promoción, incluyendo los enlaces de descarga de los mismos. Mediante estas evidencias y la revisión de la Base de Datos por medio de un SMBD, se realizaron las pruebas para verificar que las tareas realizadas funcionan de forma correcta.

En la figura 19, se muestra un fragmento de código que indica cómo se ejecuta el script de contingencia en la base de datos de Oracle.

```

public string contingencia_promo(OracleConnection connection, string query)
{
    OracleCommand command = connection.CreateCommand();
    OracleParameter parameter1 = new OracleParameter("param_cpromo", OracleType.Number, 6, "cpromocion");
    parameter1.Value = e.cpromo;
    command.Parameters.Add(parameter1);

    string resultado_ejecucion = ejecutar_query(connection, command, query);
    return resultado_ejecucion;
}

public string contingencia_msj(OracleConnection connection, string query)
{
    OracleCommand command = connection.CreateCommand();
    OracleParameter parameter2 = new OracleParameter("param_cmsj", OracleType.Number, 5, "cmensaje");
    parameter2.Value = e.cmensaje;
    command.Parameters.Add(parameter2);

    string resultado_ejecucion = ejecutar_query(connection, command, query);
    return resultado_ejecucion;
}

```

Figura 19. Ejecutar contingencia.

### Sexta iteración:

Para cumplir los objetivos establecidos para la sexta iteración, se realizaron las tareas señaladas en la tabla 9:

Tabla 9: Sprint Backlog de la sexta iteración.

ID	# TAREA	DESCRIPCIÓN
11	25	Diseño de la Base de Datos
	26	Creación de tablas
	27	Creación de restricciones
12	28	Establecer conexiones con la Base de datos SQL Server
	29	Establecer querys a ejecutar.
	30	Identificar parametros a utilizar en cada tabla
13	31	Creación de vistas de recuperación de datos
	32	Identificar promoción a recuperar.
	33	Establecer conexiones con la Base de datos SQL Server
	34	Establecer querys a ejecutar.
14	35	Carga de datos obtenidos de la recuperación
	36	Creación de vistas de reedición

En esta fase del proyecto, se diseñó un repositorio de local de información, utilizando las bases de datos de SQL Server para almacenar aquellas promociones configuradas por medio de la

aplicación. En la figura 20, se muestra el diseño de las tablas realizadas para almacenar la información y las relaciones entre ellas.

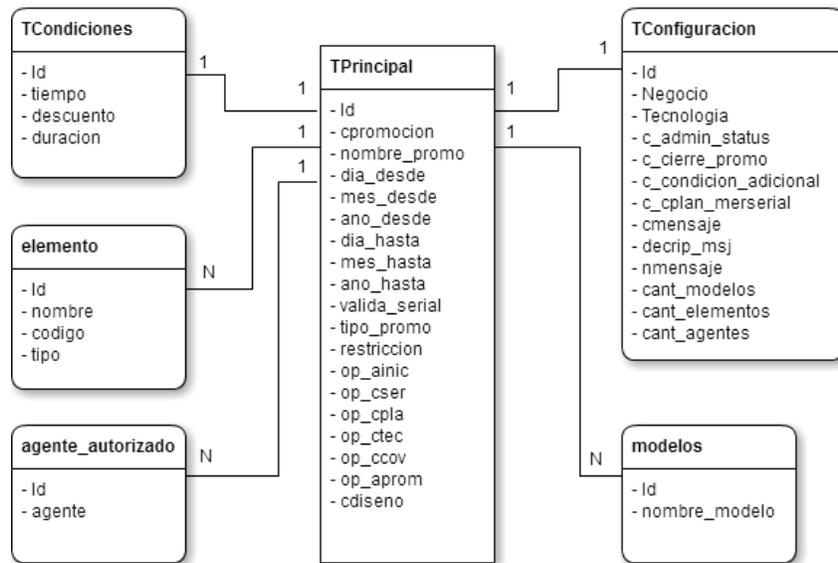


Figura 20: Modelo de Base de Datos Local.

Luego de definir las tablas y sus restricciones, se realizaron los procedimientos y las vistas necesarias para la carga y el almacenamiento de las promociones realizadas por medio de la aplicación. En la figura 21, se muestra un fragmento de código C# que permite cargar los elementos de una promoción en la aplicación.

```
public void leer_elementos(int id) {
    string consulta = "SELECT nombre,codigo,tipo FROM Promociones.dbo.elemento where Id=" + id;
    string connectionstring = "Data Source=INNOVATEC-PORTA;Integrated Security=True;Connect Timeout=15;Encrypt=False;";
    using (System.Data.SqlClient.SqlConnection conn = new System.Data.SqlClient.SqlConnection(connectionstring))
    {
        conn.Open();
        using (System.Data.SqlClient.SqlCommand myCommand = new System.Data.SqlClient.SqlCommand(consulta, conn))
        {
            System.Data.SqlClient.SqlDataReader Reader = myCommand.ExecuteReader();
            int i = 0;
            while (Reader.Read())
            {
                e.nombre_elemento[i] = Reader.GetString(0);
                e.cod_elemento[i] = Reader.GetInt32(1);
                e.tipo_elemento[i] = Reader.GetString(2);
                i++;
            }
        }
    }
}
```

Figura 21. Código de carga de promociones.

En la figura 22, se puede observar un fragmento del código C# mediante el cual se realiza la carga de algunos datos en la aplicación desde la base de datos SQL Server:

```

public void leer_tcondiciones(int id) {
    string consulta = "SELECT descuento,tiempo,duracion FROM Promociones.dbo.TCondiciones where Id=" + id;
    string connectionstring = "Data Source=INNOVATEC-PORTA;Integrated Security=True;Connect Timeout=15;Encrypt=False;
using (System.Data.SqlClient.SqlConnection conn = new System.Data.SqlClient.SqlConnection(connectionstring))
{
    conn.Open();
    using (System.Data.SqlClient.SqlCommand myCommand = new System.Data.SqlClient.SqlCommand(consulta, conn))
    {
        System.Data.SqlClient.SqlDataReader Reader = myCommand.ExecuteReader();

        if (Reader.Read())
        {
            e.descuento = Reader.GetInt32(0);
            e.tiempo = Reader.GetInt32(1);
            e.duracion = Reader.GetString(2);
        }
    }
}
}
}

```

Figura 22. Carga de datos.

En la figura 23 se muestra un fragmento de código que permite insertar datos en dicho repositorio.

```

if (c_in.c_modelos > 0) { for (int i = 0; i < c_in.c_modelos; i++) {
    try
    {
        string query_TModelos = "INSERT INTO Promociones.dbo.modelos(Id,nombre_modelo) VALUES (@ID_promo,@modelo)";
        System.Data.SqlClient.SqlCommand Command_model = conn.CreateCommand();
        Command_model.CommandType = CommandType.Text;
        Command_model.CommandText = query_TModelos;

        System.Data.SqlClient.SqlParameter param_model_1 = new System.Data.SqlClient.SqlParameter("@ID_promo", SqlDbType.Int);
        param_model_1.Value = id;
        Command_model.Parameters.Add(param_model_1);

        System.Data.SqlClient.SqlParameter param_model_2 = new System.Data.SqlClient.SqlParameter("@modelo", SqlDbType.VarChar, 10);
        param_model_2.Value = e.modelo[i];
        Command_model.Parameters.Add(param_model_2);

        Command_model.ExecuteNonQuery();

        sw.Write("Se inserta con exito en Promociones.dbo.modelos el modelo: " + i + ".\n");
    }
    catch (Exception e)
    {
        sw.Write(e.ToString());
        sw.Write("\n Se produjo un error Promociones.dbo.modelos");
    }
}
}
}

```

Figura 23. Almacenamiento de la promoción.

Para manipular las bases de datos de Oracle y SQL Server, es necesario utilizar librerías distintas dentro de Microsoft .NET. Sin embargo, las funciones diseñadas en este framework son bastante similares entre ellas.

Al finalizar las tareas del sprint backlog, fueron realizadas un conjunto de pruebas para comprobar que las promociones configuradas sean almacenadas de forma correcta en la Base Datos Local, así como fue verificado que las mismas puedan ser cargadas nuevamente para ser gestionadas.

### **Séptima iteración:**

En la última iteración, se completaron las tareas señaladas en la tabla 10, dando por culminado el desarrollo de la aplicación:

Tabla 10: Sprint Backlog de la séptima iteración.

<b>ID</b>	<b># TAREA</b>	<b>DESCRIPCIÓN</b>
15	37	Identificar Promoción a eliminar
	38	Establecer conexiones con la Base de datos SQL Server
	39	Establecer querys y parámetros a ejecutar.
16	40	Modificación de las vistas
	41	Modificación del CSS
17	42	Módulo de inicio de sesión en la aplicación

En las tareas de esta última fase, se desarrollaron las vistas y funciones requeridas para eliminar las promociones almacenadas dentro de la base de datos local. En la figura 24, se muestra la porción de código C# que permite eliminar las promociones de la base de datos SQL Server.

```

public void Elimina_Promo_BD(int id, string query) {

    string connectionstring = "Data Source=INNOVATEC-PORTA;Integrated Security=True;Connect Timeout=15;Encrypt=False;TrustServerCer
using (System.Data.SqlClient.SqlConnection conn = new System.Data.SqlClient.SqlConnection(connectionstring))
{
    conn.Open();
    try
    {
        System.Data.SqlClient.SqlCommand Command_model = conn.CreateCommand();
        Command_model.CommandType = CommandType.Text;
        Command_model.CommandText = query;

        System.Data.SqlClient.SqlParameter param_model_1 = new System.Data.SqlClient.SqlParameter("@ID_promo", SqlDbType.Int);
        param_model_1.Value = id;
        Command_model.Parameters.Add(param_model_1);

        Command_model.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }
}
}
}

```

Figura 24. Código para eliminar promociones.

Dando por terminados todos los requerimientos funcionales de la aplicación, se realizaron las modificaciones en las vistas, utilizando el framework Bootstrap en el CSS de la aplicación. En la figura 25 se puede observar una porción del CSS. Este cambio en el código permite que la aplicación también pueda ser utilizada en dispositivos móviles.

```

[ ] .content-info label{
    font-size: 14px;
}

[ ] .mb20 {
    margin-bottom: 20px;
}

[ ] .jumbotron button{
    border-radius: 4px;
    font-size: 16px;
}

[ ] .jumbotron button a{
    color: white;
    border-radius: 4px;
    text-decoration: none;
}

[ ] .btn-jumbo-small {
    font-size: 12px;
}

[ ] .input-reduce{
    width: 75px;
    display: inline;
    text-align: center;
}

[ ] .input-radio-button{
    font-size: 16px;
}

```

Figura 25: Fragmento de código CSS.

Por último, se integraron los módulos otorgados por .NET para el control de inicio de sesión. Agregando restricciones en todas las vistas para que los usuarios que no estén registrados en el sistema no tengan permitido ingresar a la aplicación.

Para finalizar el sprint, fueron realizadas un conjunto de pruebas, que comprendían consultar la Base de Datos Local para verificar que fuera eliminado todo lo referente a la promoción a eliminar. También se hizo una revisión para comprobar que el código CSS permite que la aplicación se pueda ver correctamente en cualquier navegador y sin importar en que dispositivo esté. Por último, fue verificado que todas las funcionalidades de la aplicación solo se puedan ejecutar si el usuario realizó un inicio de sesión previamente.

Al final de cada iteración, se realizaron reuniones llamadas Sprint review, que contenía una versión programada según lo establecido. Analizando los beneficios agregados y las posibles mejoras a realizar en la aplicación. En la tabla 11, se describen las funcionalidades agregadas al final cada iteración.

Tabla 11: Funcionalidades entregadas en cada iteración.

<b>ITERACIÓN</b>	<b>RESULTADO</b>
1	Página web con las vistas básicas de la aplicación y formularios para el llenado de la información referente a la promoción
2	Creación y descarga de scripts de la promoción
3	Edición de las promociones, Consulta de identificadores de forma automática
4	Ejecución de scripts de instalación
5	Ejecución de scripts de contingencia. Descarga de spools de instalación
6	Permite almacenar las promociones creadas y cargarlas para su uso
7	Permite eliminar las promociones almacenadas y restringe la aplicación a los usuarios que tengan una sesión creada.

Luego de cada Sprint review, se realizó una reunión en donde se evaluaron los aspectos a mejorar para las siguientes iteraciones. También fueron evaluados los problemas presentados durante el desarrollo y las posibles soluciones. En la última iteración, se analizaron las posibles mejoras para el desarrollo de futuros proyectos.

### **3.2. Pruebas de usabilidad**

Al finalizar el desarrollo de la aplicación, se realizaron pruebas de usabilidad con el grupo de desarrolladores de la Gerencia de Configuración de Servicios de Telefónica S.A. Dichas pruebas, consistieron en permitir al usuario utilizar la herramienta, de forma tal que evaluaran libremente como gestionar una promoción, incluyendo la configuración y ejecución de las pruebas correspondientes a un requerimiento. Luego de esto, se le suministraba el manual de usuario para su lectura. Una vez leído el manual, el usuario manejaba nuevamente la aplicación. Para finalizar, el usuario respondió un test para evaluar la usabilidad de la aplicación.

El cuestionario realizado para evaluar la aplicación, fue hecho en base a la Escala de Usabilidad del Sistema (SUS, por sus siglas en inglés, System Usability Scale), el cual fue desarrollado en 1986 como parte de la introducción de la ingeniería de usabilidad a los sistemas de oficina de Digital Equipment Co. Ltd.

Para Floría (2000), el propósito de esta empresa era proporcionar un test fácil de completar, puntuar y que permitiera hacer comparaciones cruzadas de los productos. Este cuestionario debe contener diez (10) oraciones en las cuales el usuario debe asignar un valor entre uno (1) y cinco (5), en el que el menor valor representa un total desacuerdo con el contenido de la frase y el mayor valor representa un acuerdo total con la misma.

La puntuación de la escala SUS se calcula sumando la contribución de cada punto, la cual tiene un valor entre cero (0) y cuatro (4). Para los puntos impares, la contribución será el valor asignado menos uno (1). Para los puntos pares, será cinco (5) menos el valor asignado. La suma obtenida es multiplicada por 2,5. El resultado de esta operación genera un número entre cero (0) y cien (100).

El test contaba con las siguientes oraciones:

1. Los títulos, menús y textos de la aplicación son legibles.
2. Los colores utilizados no permiten la percepción adecuada de la aplicación.
3. Se ofrece una organización que facilita la navegación.
4. Los descriptores de la aplicación no son precisos.
5. El diseño de la aplicación favorece su uso.
6. No pude entender la aplicación sin utilizar el manual de usuario.
7. Los elementos de ayuda disponibles en la aplicación son útiles.
8. El tiempo de respuesta de la aplicación es muy lento.
9. La aplicación ofrece los resultados deseados.

10. Necesito aprender muchas cosas para poder utilizar la aplicación.

### Resultados del test

Luego de que un grupo de cuatro (4) usuarios realizaron las pruebas de usabilidad de la aplicación, se obtuvieron las siguientes puntuaciones de los resultados:

Test Usuario #1: 100 puntos.

Test Usuario #2: 90 puntos.

Test Usuario #3: 95 puntos.

Test Usuario #4: 97.5 puntos.

En los siguientes gráficos, se muestran las respuestas seleccionadas por los usuarios a las preguntas del test de usabilidad. Las oraciones impares reflejan un comentario positivo sobre la aplicación, mientras que las pares señalan un comentario negativo sobre la usabilidad de la herramienta.

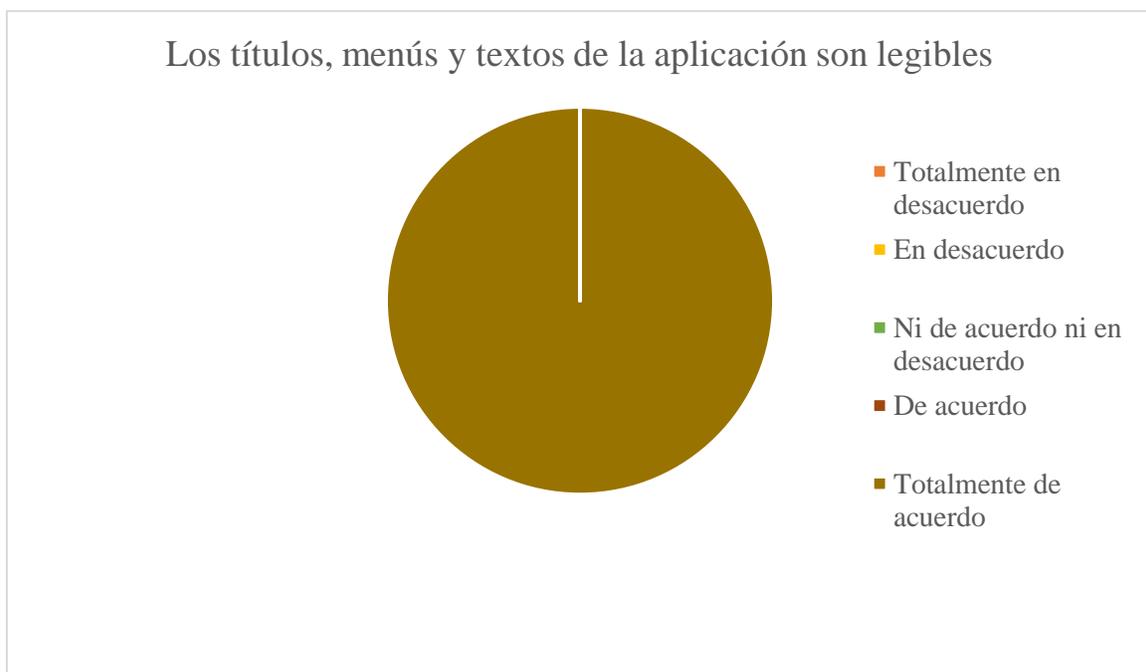
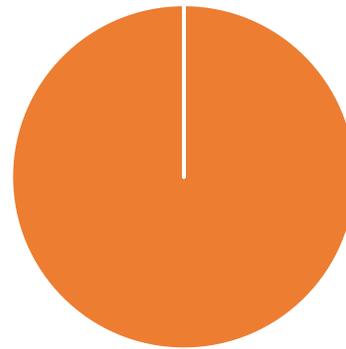


Figura 26: Test de usabilidad. Respuestas de la primera pregunta.

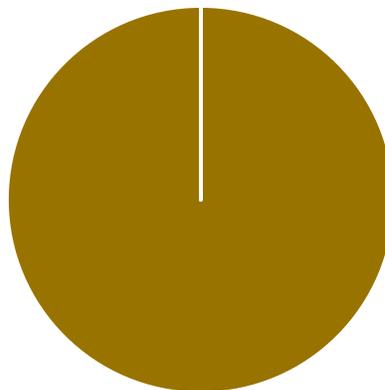
Los colores utilizados no permiten la percepción adecuada de la aplicación.



- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

Figura 27: Test de usabilidad. Respuestas de la segunda pregunta.

Se ofrece una organización que facilita la navegación



- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

Figura 28: Test de usabilidad. Respuestas de la tercera pregunta.

### Los descriptores de la aplicación no son precisos

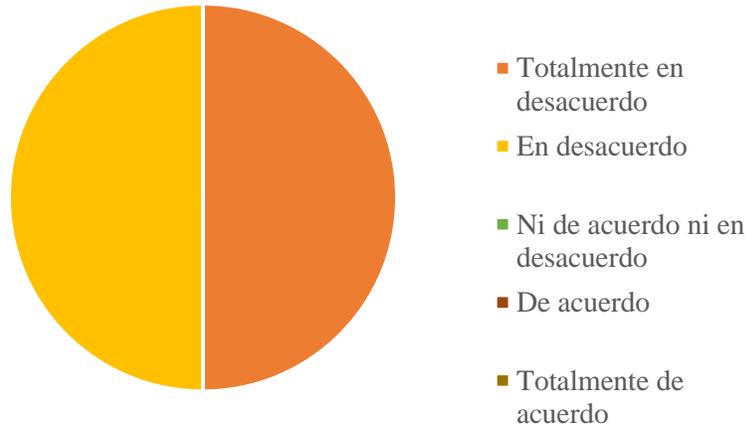


Figura 29: Test de usabilidad. Respuestas de la cuarta pregunta.

### El diseño de la aplicación favorece su uso



Figura 30: Test de usabilidad. Respuestas de la quinta pregunta.

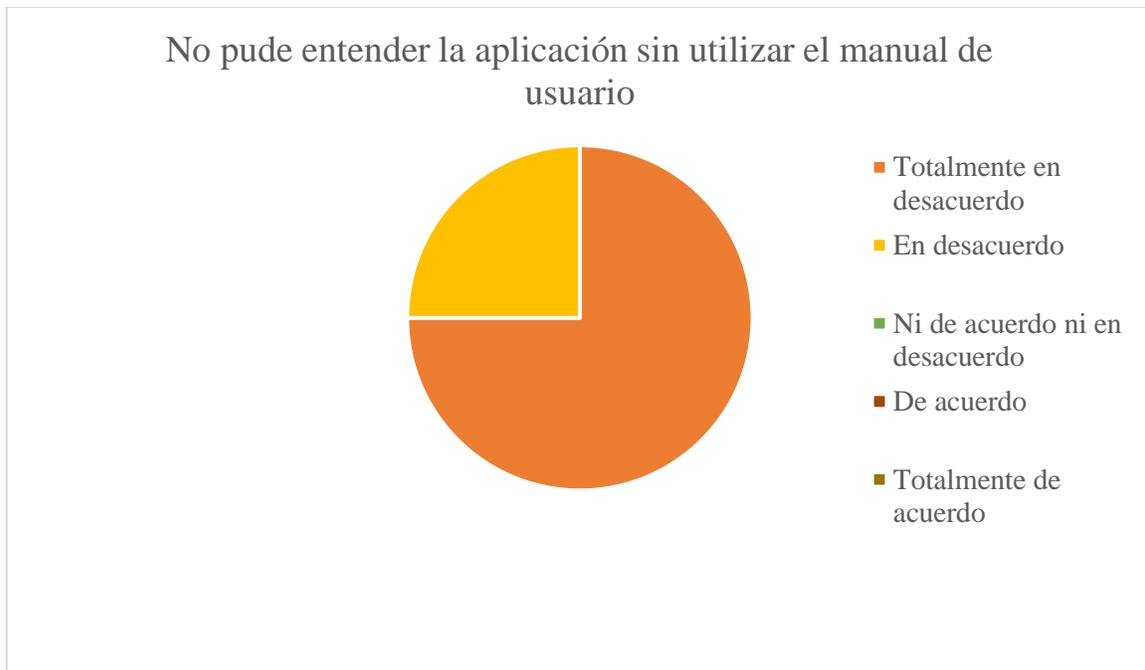


Figura 31: Test de usabilidad. Respuestas de la sexta pregunta.

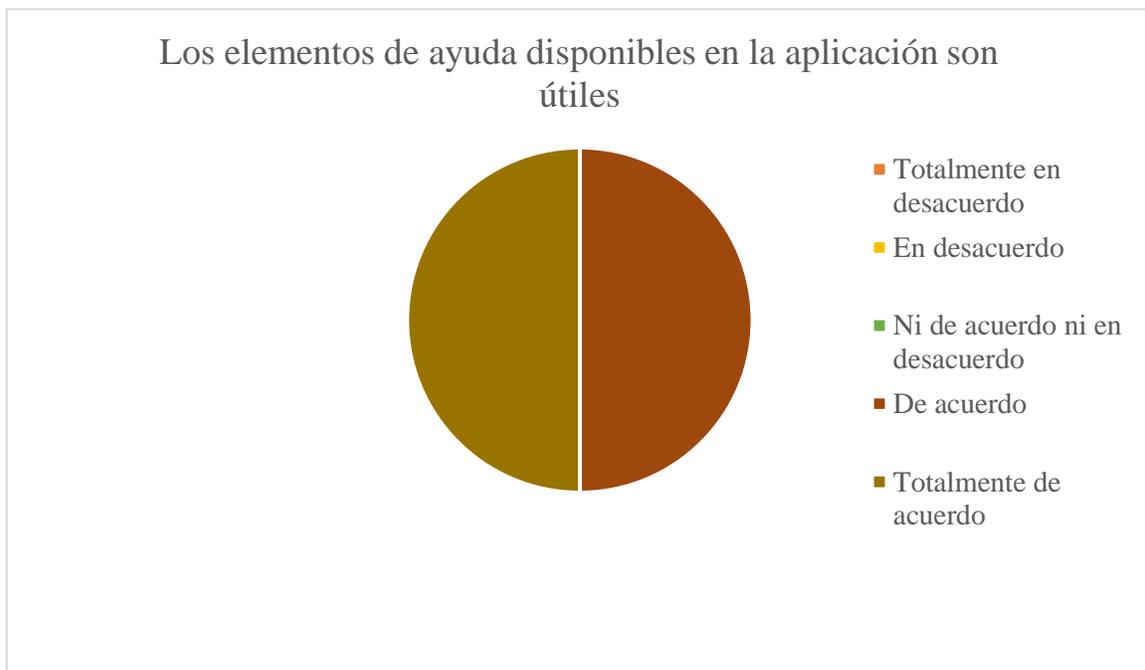


Figura 32: Test de usabilidad. Respuestas de la séptima pregunta.



Figura 33: Test de usabilidad. Respuestas de la octava pregunta.



Figura 34: Test de usabilidad. Respuestas de la novena pregunta.



Figura 35: Test de usabilidad. Respuestas de la décima pregunta.

Estos resultados permiten deducir que la aplicación muestra una buena usabilidad para los usuarios a los que está destinada, permitiendo que los mismos puedan realizar todas sus funcionalidades sin dificultades.

Luego de exponer todo el proceso aplicado mediante la metodología Scrum, la arquitectura utilizada y las pruebas de usabilidad realizadas, se considera finalizado el desarrollo de la aplicación GesPromos. En el siguiente capítulo se muestran los resultados obtenidos en este Trabajo Especial de Grado.

## CAPITULO IV RESULTADOS

En el siguiente capítulo, se realiza una descripción de la aplicación realizada, explicando mediante diagramas el comportamiento de cada módulo de la aplicación. También se analizan las vistas y las funcionalidades que otorga cada una de ellas.

### 4.1. Presentación de la Aplicación – GesPromos.

Se desarrolló una aplicación web para gestionar la configuración de promociones de prepago en una empresa de telefonía, consultando identificadores disponibles en la Base de datos de la mencionada empresa y almacenando las configuraciones realizadas para su uso o recuperación posterior. Esta herramienta recibió el nombre de GesPromos.

Al incorporar la aplicación a la Gerencia de Configuración de servicios de Telefónica S.A., el flujo mediante el cual se desarrollan las promociones de prepago es modificado, tal como se muestra en la figura 36:

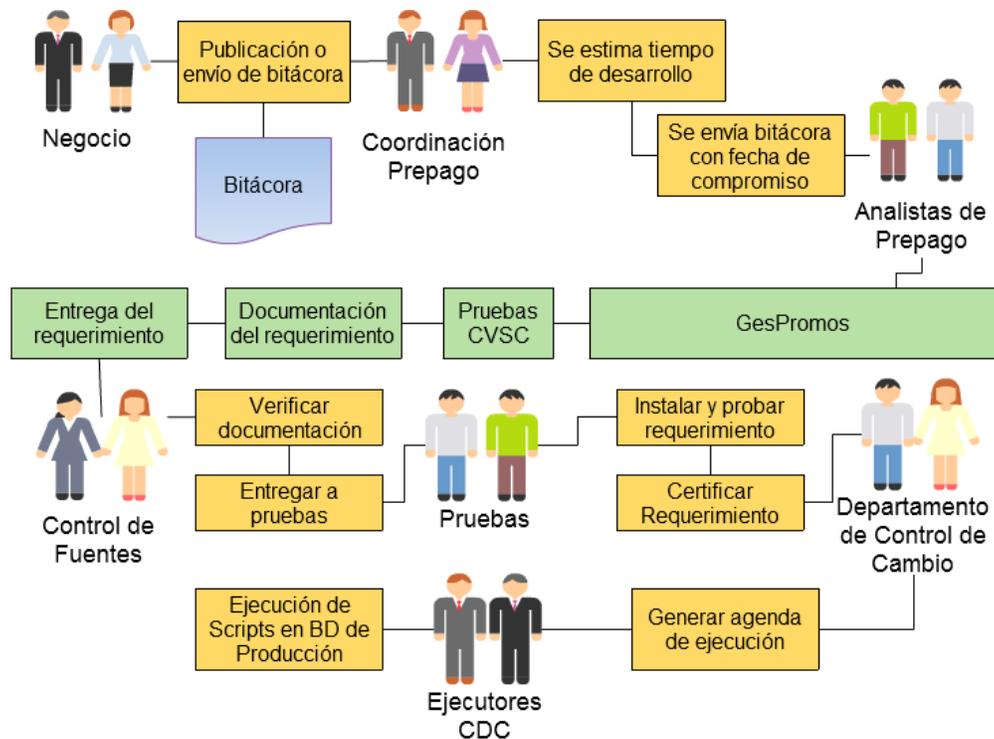


Figura 36: Diagrama de flujo para la configuración de promociones mediante GesPromos.

## 4.2. Módulos de la Aplicación

GesPromos se encuentra dividida en dos módulos: creación y gestión de promociones. El primer módulo permite crear y almacenar una promoción, el segundo permite cargar o eliminar una promoción existente. De forma adicional, se presentan dos vistas con información general sobre la creación de promociones y una que permite al usuario iniciar sesión en la aplicación.

### 4.3.1. Vistas básicas de la aplicación

#### Pantalla de inicio

Al acceder a la aplicación, se obtiene una vista general de la aplicación y sus funciones. Para poder acceder a ellas se requiere que el usuario inicie sesión en la aplicación. Es importante destacar que solo existe un rol de usuario. Por este motivo, todos los usuarios tendrán acceso a la misma información.

Luego de iniciar sesión, se puede ingresar a los módulos de creación y gestión de promociones. Para ingresar a las vistas del manual de configuración y lista de errores comunes no es necesario autenticarse en la aplicación.

En la figura 37, se observa la vista de la pantalla de inicio cuando el usuario no ha iniciado sesión.



Figura 37. Pantalla de Inicio

En la figura 38, se observa la vista de la pantalla de inicio cuando el usuario ha iniciado sesión.



Figura 38. Pantalla de Inicio luego de autenticarse.

Gracias al uso de Bootstrap, se puede acceder a la aplicación desde un dispositivo móvil, en la figura 39, se muestra la pantalla de inicio vista desde un Smartphone:



Figura 39: Pantalla de inicio desde un dispositivo móvil.

### **Manual de Configuración**

En esta vista, se muestra al usuario el diseño de las tablas que son manipuladas en la base de datos de Oracle. Se indican los nombres de las tablas, y columnas, los tipos de datos utilizados y los posibles valores establecidos para la configuración de las promociones. Esta es una página para mostrar información que puede requerir el usuario al momento de diseñar los scripts de sus promociones.

En la figura 40, se observa las distintas pestañas desplegables para cada tabla de la base de datos de Oracle.



## Tablas para la configuración de Promociones en PREPAGO

Una promoción es la forma de modificar la facturación o tasación de un concepto, a través de: descuentos, redefinición de tarifas, otorgamiento de cupos/contadores. Pueden tener asociado un cargo mensual por el disfrute de la promoción.

La configuración de una nueva promoción para Prepago se realiza en la Base de Datos IVR, en ella se ingresa toda la información necesaria para se otorgue a los clientes los elementos promocionales y el tiempo de duración de los mismos. Estos elementos promocionales pueden ser bonos o descuentos sobre la renta de algún servicio. La información necesaria para la configuración de las promociones es definida por el negocio y suministrada a través de Bitácoras.

A continuación, se muestran las tablas a configurar al momento de desarrollar una promoción:

### TCONFIG\_PROMOCION

#### TOPERACIONXPROMOCION - Ocultar

Configuración de las operaciones por las cuales se otorga la promoción.

COLUMNA	TIPO DE DATO	NULO	VALOR POR DEFECTO
C PROMOCION	NUMBER(6,0)	NO	

Figura 40. Manual de Configuración.

### Lista de errores comunes en pruebas

Cuando el usuario realiza la configuración de una promoción, debe realizar unas pruebas adicionales para garantizar el buen funcionamiento de la misma. Existen casos en los que las pruebas pueden fallar por los datos establecidos en tablas ajenas a las promociones. En esta vista, mostrada en la figura 41, se le indican al usuario las acciones a tomar en caso que sus pruebas presenten alguno de estos errores.



## Errores comunes en pruebas de desarrollo

A continuación, se enumeran algunos errores que suelen ser comunes en el proceso de pruebas de una promoción, además de la posible solución a dicho problema.

Pruebas en CVSC de Desarrollo:

CVSC El ICCID se encuentra asignado a otra cuenta en Prepago o PostPago.

Solución: Hay que buscar en la Base de Datos de CIES en test un serial adecuado, debido a que en desarrollo no están actualizados.

CVSC - Verificar los servicios que se desean eliminar y/o activar ya que existe alguna inconsistencia (-651).

Solución: Verificar los servicios que se le están asociando al plan, porque puede venir alguno que no debe estar. Verificar que los SVA que se muestran en Desarrollo corresponden a los servicios de producción o testing, ya que por lo general los ambientes no están homologados.

ORACLE – No se pudo obtener el tipo de serial (-20200).

CVSC Desarrollo presenta este error con algunos seriales de equipos. Se debe introducir otro serial o probar con otro equipo.

CVSC No se ha encontrado descripción para el error -20586. Contacte al administrador del sistema (-20586).

No se ha encontrado solución para este error en desarrollo. Se presentó al activar dos promociones

Figura 41. Lista de errores comunes.

### Inicio de Sesión

En esta vista de la aplicación, el usuario ingresa su correo electrónico y contraseña. De no realizar esta acción, no podrá acceder a los módulos de GesPromos. El formulario para ingresar estos datos se muestra en la figura 42:



## Iniciar sesión.

Ingrese su usuario para acceder a la aplicación.

Correo electrónico

Contraseña

¿Recordar cuenta?

Iniciar sesión

[Registrar como nuevo usuario](#)

[¿Ha olvidado su contraseña?](#)

© 2016 - Configurador de Promociones - Telefónica Venezuela

Figura 42: Inicio de Sesión.

### 4.2.2. Módulo de creación de promociones

Este módulo permite realizar una promoción desde cero, a través de varias vistas y formularios que solicitan información e indican las acciones a realizar. En la figura 43, se muestra el diagrama de flujo para la creación de promociones:

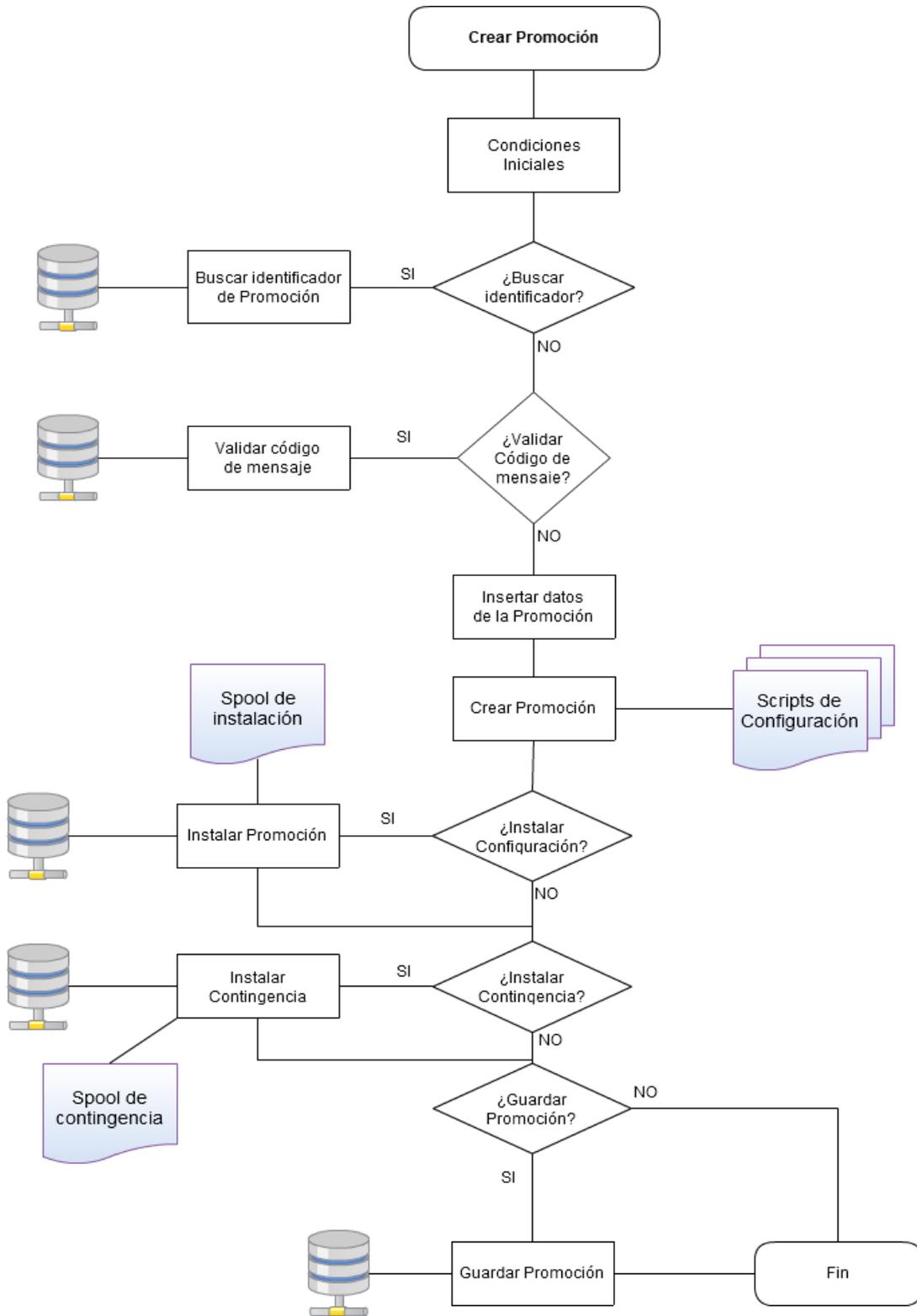


Figura 43. Diagrama de flujo – Crear Promoción.

El primer paso para crear una promoción es indicar las condiciones iniciales:

### **Condiciones Iniciales**

Para dar inicio a la configuración de la promoción, primero deben determinarse ciertos parámetros de la promoción. Dichos parámetros se pueden enumerar de la siguiente forma:

- **Buscar código de promoción:** Si el usuario solicita el código, la aplicación busca de forma automática un identificador disponible. En caso contrario, el usuario puede indicar un código de forma manual, para aquellos casos en los que deban utilizar un código determinado de forma previa.
- **Elementos de la promoción:** El usuario determina si la promoción contiene elementos promocionales. De ser así, indica cuantos elementos contiene. Puede indicar de cero hasta veinte elementos.
- **Modelos de la promoción:** El usuario indica si la promoción tiene modelos asociados. Si no tiene, la misma funciona para cualquier modelo. Si tiene, puede indicar hasta doscientos.
- **Agentes de la promoción:** El usuario debe indicar si la promoción tiene asociados agentes autorizados. Puede indicar entre cero y doscientos agentes.
- **Código de mensaje:** El usuario puede consultar si un código específico se encuentra disponible en la base de datos. De ser así, la aplicación enviará el código a la configuración.

En la figura 44, se solicita al usuario que indique las condiciones iniciales para configurar su promoción.

## Condiciones Iniciales de la Promoción

¿Desea recibir un código de promoción de forma automática? Puede ser modificado más adelante.

Buscar código de promoción

¿La promoción requiere agregar elementos?



¿La promoción requiere agregar modelos específicos?



¿La promoción requiere agregar agentes autorizados?



¿Desea consultar un código de mensaje específico?



Siguiente

Figura 44. Condiciones iniciales.

## Configuración

Una vez definidas las condiciones iniciales, la GesPromos solicita al usuario determinar toda la información perteneciente a la promoción. Luego de enviar el formulario con los datos solicitados, la aplicación genera automáticamente un script de configuración de la promoción y un script con su respectiva contingencia. Este código es generado de forma dinámica según lo solicitado por el usuario. En las figuras 45 y 46, se muestra la vista de configuración:

## Configurar Promoción

Código de la promoción:\*

1001

Nombre de la promoción:\*

Fecha de Inicio:\*

 /  / 

Fecha de Cierre:\*

 /  / 

Líneas de Negocio:\*

Fijo

Móvil

Tipo de Promoción:\*

Captación

Fidelización

Operaciones Permitidas:

AINIC - Activación inicial

CSER - Cambio de Serial

CPLA - Cambio de Plan

CTEC - Cambio de Tecnología

CCOV - Cambio de Coverage

APROM - Activación de promoción

Tecnología:\*

GSM

CDMA

Restricción de la Promoción:\*

Obligatoria

Opcional

Se Valida el serial del equipo

Planes asociados a la promoción:\*

0

Figura 45: Configuración de la promoción. Primera parte.

**Elementos de la promoción:**

Nombre del elemento

Código del elemento

Tipo de elemento

Feature  Línea gratis  Bono de saldo  Cargo  Bono

Feature  Línea gratis  Bono de saldo  Cargo  Bono

Duración de la promoción:

Porcentaje de descuento sobre el plan a migrar:

%

Días

Mes(es)

Configuración del mensaje:\*

Código del mensaje:\*

Descripción del mensaje:\*

Mensaje a enviar:\*

Condiciones de la promoción:

ADMIN\_ESTATUS

CIERRE\_PROMO

CONDICION\_ADICIONAL

CPLAN\_CMERC\_CSERIAL

**Equipos de la promoción:**

Debe añadir un código referente a cada modelo o grupo de modelos asociados a la promoción:

De no añadir ningún modelo, la promoción será compatible con cualquier modelo

**Agentes Autorizados:**

Debe añadir el código referente a cada agente autorizado para la promoción:

De no añadir ningún agente autorizado, la promoción será compatible con cualquiera

Crear Promoción

Figura 46: Configuración de la promoción. Segunda parte.

## Gestión de la Promoción

En esta vista, se le proporcionan al usuario una serie de funcionalidades para que el usuario realice la gestión que considere necesaria. En la figura 47, se muestra la vista para la gestión de la promoción.

### Gestión de la Promoción

La promoción ha sido configurada. A continuación puede descargar los archivos correspondientes a la promoción. Si lo desea, puede ejecutar los scripts en la Base de Datos de Desarrollo. No olvide guardar la promoción si desea reutilizarla en futuras configuraciones.



Figura 47. Gestión de una Promoción

Para gestionar la promoción, el usuario puede realizar las siguientes acciones:

#### Editar Promoción

Esta vista es idéntica a la configuración de la promoción, con la diferencia de que se cargan de forma automática todos los valores que el usuario estableció para la promoción. Permite al usuario editar los valores establecidos y gestionar de nuevo su promoción.

#### Descargar Configuración

Permite al usuario descargar el script generado para la promoción. Este archivo es el resultado más importante de toda la configuración.

En la figura 48 se puede observar un fragmento de código almacenado en el script:

```
spool t2_config.txt

-----TCONFIG_PROMOCION

UPDATE tconfig_promocion
  SET npromocion = 'Promocion de prueba de desarrollo',
  finicio = TO_DATE ('12/02/2016 00:00:00', 'DD/MM/YYYY HH24:MI:SS'),
  ffin = TO_DATE ('31/03/2016 23:59:59', 'DD/MM/YYYY HH24:MI:SS'),
  qmax_asignaciones = 1,
  qdias_preaviso = 0,
  inhabilitado = 'S',
  ivalida_serial = 'N',
  ivalida_doc_id = 'N',
  itipo_promocion = 'C',
  fmax_otorgamiento = TO_DATE ('31/03/2016 23:59:59', 'DD/MM/YYYY HH24:MI:SS'),
  iunidad_asignacion = 'V',
  irestriccion = 'O',
  qdias_gracia = 0,
  ivalida_listas = NULL,
  iaprovisiona_aa = 'S'
 WHERE cpromocion = 1002;

-----TTIPO_SERVICIOXPROMOCION

INSERT INTO ttipo_servicioxpromocion(cpromocion, iservicio) VALUES (1002, 'C');

-----TTIPO_CENTRALXPROMOCION

INSERT INTO ttipo_centralxpromocion(cpromocion, ctipo_central) VALUES (1002, 'G');

-----TTECNOLOGIAXPROMOCION

INSERT INTO ttecnologiaxpromocion(cpromocion, ctecnologia) VALUES (1002, 'GSM');
```

Figura 48. Archivo t2\_config.sql

## Descargar Contingencia

Permite al usuario descargar un script de contingencia generado para la promoción. En la figura 49, se muestra un fragmento del código en lenguaje Oracle SQL.

```

spool cont_t2_config.txt

DELETE FROM TDINAMICOXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM TAGENTEXXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM TMODELOXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM PSP.TROLXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM TCONDICIONXPROMOCION_DET where CCONDICION_PROMO in
( select CCONDICION_PROMO from TCONDICIONXPROMOCION where CPROMOCION = 1002 );
DELETE FROM TCONDICIONXPROMOCION where CPROMOCION = 1002;
DELETE FROM TSMC_EXPIRA_PROMO where CPROMOCION = 1002;
DELETE FROM TMENSAJEXCONFIG_PROMOCION WHERE CMENSAJE = 1955;
DELETE FROM TDETALLE_MENSAJE WHERE CMENSAJE = 1955;
DELETE FROM TMENSAJE WHERE CMENSAJE = 1955;
DELETE FROM TFORMA_APLICACION WHERE CPROMOCION = 1002;
DELETE FROM TELEMENTOXPRIMO WHERE CPROMOCION = 1002;
DELETE FROM TDISENOXXPROMOCION where cpromocion = 1002;
DELETE FROM TTECNOLOGIAXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM TTIPO_CENTRALXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM TTIPO_SERVICIOXPROMOCION WHERE CPROMOCION = 1002;
DELETE FROM TOPERACIONXPROMOCION WHERE CPROMOCION = 1002;
UPDATE TCONFIG_PROMOCION
  SET IHABILITADO = 'N'
  WHERE CPROMOCION = 1002;
spool off;

```

Figura 49. Archivo cont\_t2\_config.sql

## Ejecución de la instalación

Al utilizar esta funcionalidad, la aplicación ejecuta el script generado en la base de datos SQL. Esta función instala la promoción y crea un archivo que almacena si la misma fue ejecutada de forma correcta. Este archivo es llamado spool\_t2\_config.txt.

## Descargar Spool de Configuración

Luego de ejecutar el archivo de instalación, el botón destinado para esta funcionalidad desaparece, dando paso a dos nuevas opciones: una para ejecutar el script de contingencia y otra que permite descargar el archivo generado, tal como se muestra en la figura 50:

## Gestión de la Promoción

El script de configuración ha sido ejecutado en la Base de Datos de Desarrollo. A continuación, puede descargar el spool generado. Si así lo desea, puede instalar la contingencia y deshacer los cambios.



Figura 50: Gestión de la promoción.

En la figura 51 se muestra un fragmento del spool generado por la aplicación al ejecutar el script de instalación en la base de datos de Oracle:

```
INSERT INTO ttipo_servicioxpromocion(cpromocion, iservicio) VALUES (1001, 'C');
1 fila(s) afectada(s).

INSERT INTO ttipo_centralxpromocion(cpromocion, ctipo_central) VALUES (1001, 'G');
1 fila(s) afectada(s).

INSERT INTO ttecnologiaxpromocion(cpromocion, ctecnologia) VALUES (1001, 'GSM');
1 fila(s) afectada(s).

INSERT INTO toperacionxpromocion(cpromocion, coperacion) VALUES (1001, 'AINIC');
1 fila(s) afectada(s).

INSERT INTO toperacionxpromocion(cpromocion, coperacion) VALUES (1001, 'APROM');
1 fila(s) afectada(s).

INSERT INTO tdisenoxpromocion(cpromocion, cdiseño, cservicio)
SELECT 1001, cdiseño, cservicio
FROM tdiseño
WHERE cdiseño IN (292);
1 fila(s) afectada(s).
```

Figura 51. Spool de instalación.

## Ejecución de la contingencia

Al utilizar esta funcionalidad, la aplicación ejecuta el script de contingencia en la base de datos SQL. Esta función elimina todos los datos de la promoción de la base de datos y crea un archivo que almacena los resultados de esta ejecución. Este archivo es llamado `spool_cont_t2_config.txt`. Al ejecutar esta opción, dicha funcionalidad desaparece, dando paso a un nuevo botón que permite descargar el archivo generado, tal como se muestra en la figura 52.

## Gestión de la Promoción

El script de contingencia ha sido ejecutado en la Base de Datos de Desarrollo. A continuación, puede descargar el spool generado. No olvide guardar su configuración si desea utilizarla en otro momento.



Figura 52: Gestión de la promoción.

## Descargar Spool de Contingencia

Permite al usuario descargar el archivo generado al ejecutar la contingencia en la base de datos de Oracle. En la figura 53 se muestra un fragmento del spool generado por la aplicación.

```

>SQL: DELETE FROM TMODELOXPROMOCION WHERE CPROMOCION = 1001;
216 fila(s) afectada(s).

>SQL: DELETE FROM PSP.TROLXPROMOCION WHERE CPROMOCION = 1001;
2 fila(s) afectada(s).

>SQL: DELETE FROM TCONDICIONXPROMOCION_DET where CCONDICION_PROMO
in ( select CCONDICION_PROMO from TCONDICIONXPROMOCION where CPROMOCION = 1001 );
6 fila(s) afectada(s).

>SQL: DELETE FROM TCONDICIONXPROMOCION where CPROMOCION = 1001;
4 fila(s) afectada(s).

>SQL: DELETE FROM TSMC_EXPIRA_PROMO where CPROMOCION = 1001;
1 fila(s) afectada(s).

>SQL: DELETE FROM TMENSAJEXCONFIG_PROMOCION WHERE CMENSAJE = 1953;
1 fila(s) afectada(s).

>SQL: DELETE FROM TDETALLE_MENSAJE WHERE CMENSAJE = 1953;
1 fila(s) afectada(s).

```

Figura 53. Spool de contingencia.

## Guardar Promoción

Para finalizar la gestión de la promoción, el usuario tiene la opción de almacenar los datos de la configuración en la base de datos de SQL Server. Al realizar esta acción, la aplicación regresa a la pantalla de inicio.

### 4.2.3. Módulo de gestión de promociones

Este módulo permite gestionar las promociones que fueron guardadas anteriormente. Esto otorga al usuario la posibilidad de descargar nuevamente scripts ya configurados, crear nuevas promociones a partir de una anterior y eliminar promociones que ya no considere necesarias. En la figura 54 se muestra el diagrama de flujo de GesPromos para configurar una promoción a partir de una almacenada en la base de datos Local:

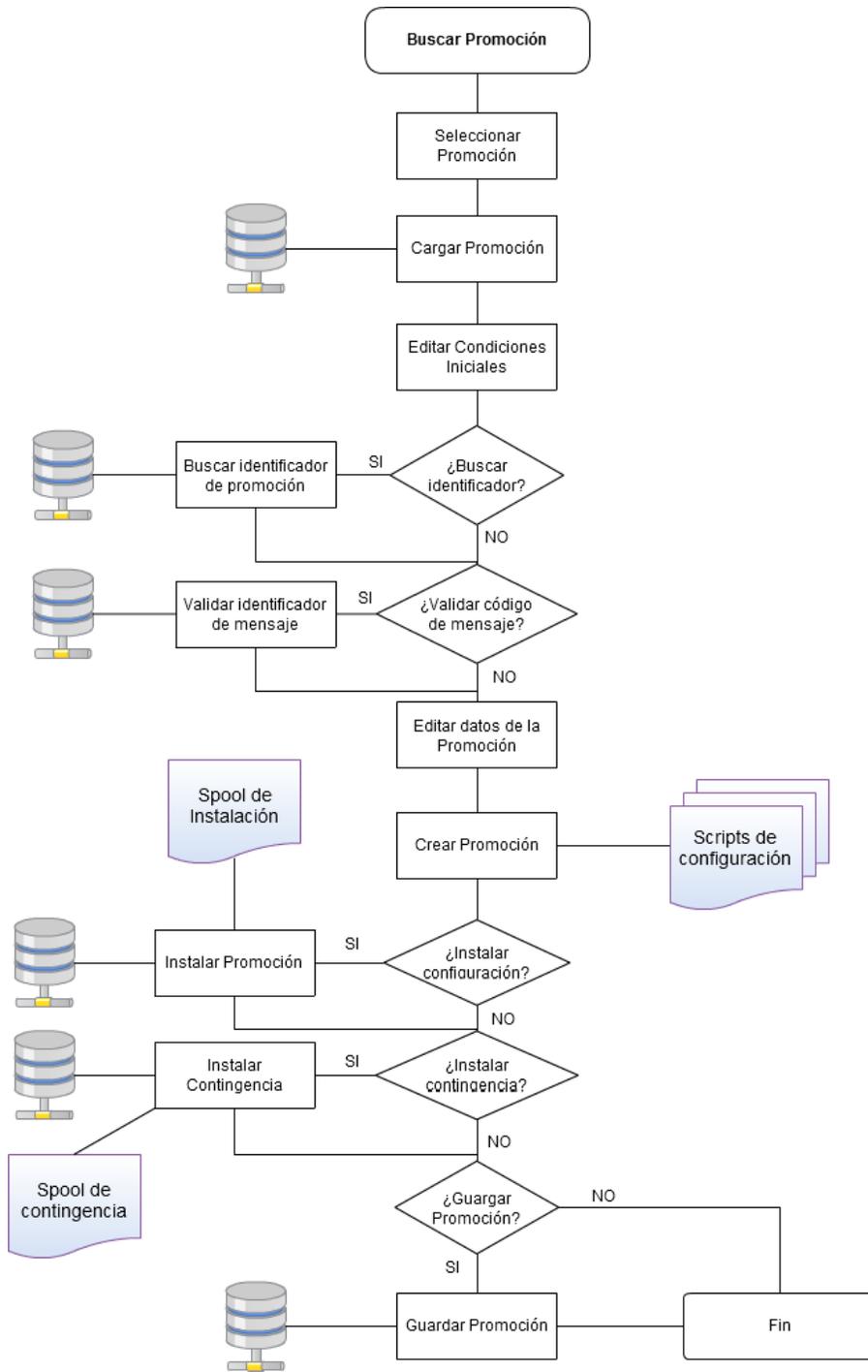


Figura 54. Diagrama de flujo – Buscar Promoción.

En la figura 55, se describen los pasos a seguir al momento de eliminar una promoción de la aplicación.

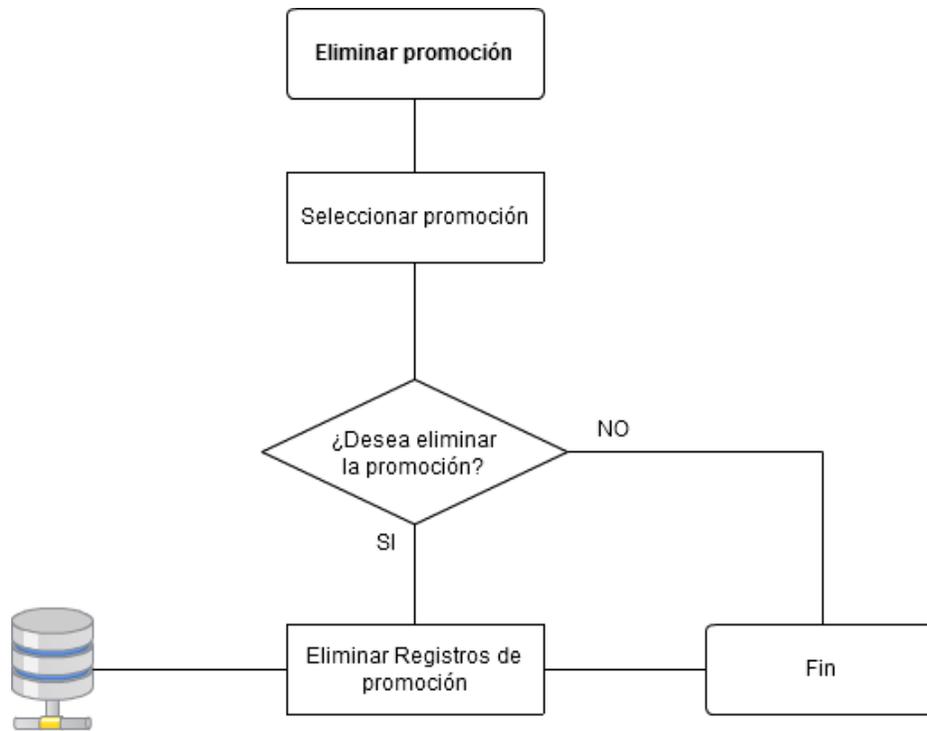


Figura 55. Diagrama de flujo – Eliminar Promoción.

### Gestionar Promoción

Esta vista permite al usuario ver las promociones almacenadas en SQL Server. A partir de esto, se pueden buscar las promociones para ser reutilizadas o se pueden eliminar de la base de datos. En la figura 56 se muestra la vista de la gestión de promociones:

## Gestión de Promociones

Existen 3 promociones disponibles.

Código de promoción	Nombre de la Promoción
1001	Promocion de prueba 1001
1004	Promocion de prueba 1003
1002	Promocion de prueba 1002



Figura 56. Gestión de promociones almacenadas.

### Buscar promoción: Seleccionar de una lista

Al ingresar a esta vista, el usuario puede seleccionar de una lista que promoción desea cargar. En la figura 57, se muestra la vista para seleccionar las promociones:

### Cargar Promoción

Seleccione la promoción que desea cargar:

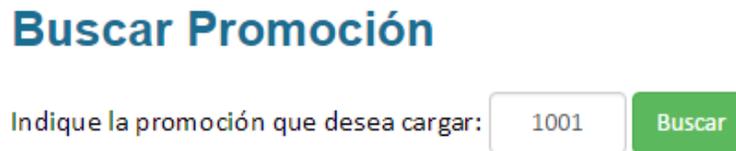
Selecciona	Código de promoción	Nombre de la Promoción
<input checked="" type="radio"/>	1001	Promocion de prueba 1001
<input type="radio"/>	1004	Promocion de prueba 1003
<input type="radio"/>	1002	Promocion de prueba 1002



Figura 57. Buscar Promoción.

## Buscar por código de promoción

Esta funcionalidad permite al usuario buscar una promoción a partir de un identificador de promoción. En la figura 58 se muestra la vista de la aplicación.



The screenshot shows a web interface with the title "Buscar Promoción" in blue. Below the title, there is a text label "Indique la promoción que desea cargar:" followed by a text input field containing the number "1001" and a green button labeled "Buscar".

Figura 58. Buscar promoción por código.

Si GesPromos no encuentra la promoción solicitada por el usuario, se le envía una notificación de error, tal como se muestra en la figura 59.



The screenshot shows the same "Buscar Promoción" interface. Above the search input, there is a blue error message: "El código de promoción no existe. Por favor intente nuevamente." The search input field now contains the number "1500" and the "Buscar" button is still present.

Figura 59. Buscar promoción por código – Error.

## Editar Condiciones Iniciales

Luego de seleccionar una promoción, se cargan los datos para ser reutilizados. El primer paso consiste en cargar las condiciones iniciales. El usuario debe decidir si necesita conservar estos datos o efectuar algún cambio. Puede buscar un nuevo identificador de promociones o utilizar el código guardado. De la misma forma, debe decidir si utilizar el código de mensaje que fue almacenado para la promoción. Esta vista es idéntica a la carga de condiciones iniciales del módulo de creación de promociones.

## Configuración

Al llegar al punto de configuración, llegamos a la vista de edición del módulo de creación de promociones. Sin embargo, esta es cargada con todos los datos recibidos de la base de datos de SQL Server.

## Eliminar Promoción

En esta vista se muestran todas las promociones almacenadas en la base de datos SQL Server. Al seleccionar una de estas promociones, la misma es eliminada del repositorio. Una vez eliminada no podrá ser recuperada. Luego de realizar esta acción, GesPromos retorna al módulo de gestión. En la figura 60, se muestra la vista para eliminar promociones.

### Eliminar Promociones

Seleccione las promociones que desea eliminar:

Selecciona	Código de promoción	Nombre de la Promoción
<input checked="" type="radio"/>	1001	Promocion de prueba 1001
<input type="radio"/>	1004	Promocion de prueba 1003
<input type="radio"/>	1002	Promocion de prueba 1002

Eliminar

Figura 60. Eliminar promociones.

Mediante el uso de esta herramienta, el equipo de desarrollo de la Gerencia de Configuración de servicios de Telefónica S.A consigue simplificar la gestión de las promociones a realizar, evitando incluso que el usuario tenga que programar una solución. GesPromos también otorga a sus usuarios la capacidad para recuperar las promociones ya realizadas anteriormente, ya sea para eliminarlas, obtener nuevamente los archivos de descarga o para utilizarlos como base de un nuevo requerimiento asignado.

## CONCLUSIONES

Las tecnologías de desarrollo de software son utilizadas cada día con mayor fuerza en el mercado, ya que son una gran herramienta para administrar la información y cumplir con las necesidades que se presenten dentro de las organizaciones que así lo requieran.

El hecho de utilizar una aplicación que permita automatizar un conjunto de procesos de una organización, brinda ciertas ventajas estratégicas en cuanto a costos, tiempo y esfuerzo, mejorando la calidad y el desarrollo, optimizando ciertas operaciones y la ejecución de tareas específicas.

Mediante el uso de las tecnologías señaladas en este documento, se ha desarrollado una aplicación web que permite gestionar el proceso de configuración de las promociones de la plataforma prepago de Telefónica S.A, con el objeto de otorgar a los desarrolladores de la empresa una herramienta que facilite el desarrollo de sus requerimientos, reduciendo el esfuerzo y el tiempo invertido en cada uno de ellos. El uso de esta aplicación reduce los errores de configuración y los tiempos de entrega.

Al desarrollar una herramienta para optimizar un proceso, también se rediseña la forma de llevar a cabo el conjunto de tareas que lo conforman, agregando y modificando aquellos factores que puedan mejorar el método de trabajo para procedimientos específicos.

Este trabajo de investigación puede tomarse como un punto de partida para futuros proyectos dentro de Telefónica S.A. Mediante el uso de las mismas tecnologías, se pueden desarrollar nuevas aplicaciones para la gestión de otro tipo de requerimientos realizados dentro de la Gerencia de Configuración de Servicios de la empresa.

## **RECOMENDACIONES**

Luego de cumplir con los objetivos establecidos en este Trabajo Especial de Grado, se indican las siguientes recomendaciones, que pueden seguir mejorando los procesos realizados por la Gerencia de Configuración de Servicios de Telefónica S.A:

1. Estudiar las funcionalidades obtenidas para esta aplicación y como podrían aplicarse en el desarrollo de otros requerimientos que se realizan de forma regular dentro del departamento.
2. Aumentar el alcance de la aplicación para configurar cualquier elemento que se desee otorgar a los clientes de la plataforma.
3. Estudiar el uso de las tecnologías de desarrollo aplicadas en este Trabajo Especial de Grado para ser implementadas en la plataforma de postpago.
4. Considerar el uso de metodologías de desarrollo ágiles para la gestión de los requerimientos recibidos en el departamento.

## REFERENCIAS

Calle, H. (2012). *Sistemas Gestores de Base de Datos*. Recuperado el 04/08/2015, de <http://es.slideshare.net/HernanOkamura/sistemas-de-gestores-de-base-de-datos-13332504>.

Camarena, J., Trueba, A., Martínez, M. y López, M. (noviembre-febrero, 2012). Automatización de la codificación del patrón modelo vista controlador (MVC) en proyectos orientados a la web. *Revista Ciencia Ergo Sum*. Vol. 19, núm. 3. 239-250.

Charte, F. y Serrano, J. *Microsoft visual studio .NET* (Primera Edición). Madrid, España: ANAYA. (2002).

Corporación Oracle (2015). *Oracle para empresas en crecimiento*. Recuperado el día 01/08/2015 de <http://www.oracle.com/lad/solutions/midsize/index.html>

Davenport, T. y Prusak, L. (1999). *Dato e información*. Recuperado el día 10/07/2015 de <https://es.scribd.com/doc/12699281/Dato-e-Informacion>

Elmasri, R. y Navathe, S. *Fundamentos de sistemas de bases de datos* (Tercera Edición). Madrid, España: Pearson. (2002)

Floría, A. (2000). Usabilidad y Diseño Centrado en el Usuario. Recuperado el día 28/04/2016 de <http://www.sidar.org/recur/desdi/traduc/es/visitable/nuevos/CuestCon.htm>

Fowler, M. (2005). *The new methodology*. Recuperado el día 15/04/2015 de <http://www.martinfowler.com/articles/newMethodology.html>

Gallego, M. (2014). *Tecnologías de desarrollo de aplicaciones web*. Recuperado el día 09/07/2015 de <http://es.slideshare.net/micahgallego/tema2-tecnologas-de-desarrollo-de-aplicaciones-web>

Gracia, J. (2006). *Gestión de proyectos con scrum*. Recuperado el día 08/04/2015 de <http://www.ingenierosoftware.com/equipos/scrum.php>

Gutiérrez, C. (2008). *Sistema gestor de base de datos Oracle*. Recuperado el día 01/08/2015 de [www.dcc.uchile.cl/~clgutier/Oracle1.ppt](http://www.dcc.uchile.cl/~clgutier/Oracle1.ppt)

Hernández, O. (2012). *Ciclo de vida scrum*. Recuperado el día 02/08/2015 de <http://analisisydise.blogspot.com/2012/04/ciclo-de-vida-rup-el-proceso-unificado.html>

Luján, S. (2002). *Programación de aplicaciones web: historia, principios básicos y principios web. (Primera Edición)*. Madrid, España: Club Universitario

Mesia, A. y Moncada M. (2012). *Manual de Configuración de Promociones en Base de Datos IVR. (Versión 1.0)*. Caracas. Telefónica S.A.

Microsoft (2015). *Información general acerca de .NET framework*. Recuperado el día 01/07/2015 de [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)

Microsoft (2015). *Introducción a .NET framework*. Recuperado el 01/07/2015 de [https://msdn.microsoft.com/es-es/library/hh425099\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/hh425099(v=vs.110).aspx)

Microsoft (2016). *Introducción al lenguaje C# y .NET Framework*. Recuperado el día 13/01/2016 de <https://msdn.microsoft.com/es-es/library/z1zx9t92.aspx>

Microsoft (2016). *Microsoft SQL Server*. Recuperado el día 11/04/2016 de <https://msdn.microsoft.com/es-es/library/bb545450.aspx>

Moreno, A. (2000). *Diseño e implementación de un léxico computacional para lexicografía y traducción automática*. Recuperado el día 08/07/2015 de <http://elies.rediris.es/elies9/index.htm>

Palacio, J. (2014). *Gestión de proyectos scrum manager*. Recuperado el día 10/04/2015 de <http://www.safecreative.org/work/1504243922697-gestion-de-proyectos-scrum-manager>

Pinto, M. (2004) *Bases de datos*. Recuperado el día 08/07/2015 de [http://www.mariapinto.es/e-coms/bases\\_datos.htm](http://www.mariapinto.es/e-coms/bases_datos.htm)

Ramos, D. (2009) Microsoft SQL Server. Recuperado el día 11/04/2016 de <http://www.monografias.com/trabajos73/microsoft-sql-server/microsoft-sql-server2.shtml>

Rivas, M. (2010). *Ventajas y desventajas de las aplicaciones web*. Recuperado el día 01/08/2015 de <http://www.vidamrr.com/2010/03/ventajas-y-desventajas-de-las.html>

Ruiz, E. (2001). *Lenguajes de programación. Conceptos y paradigmas*. Recuperado el día 10/07/15 de Ruiz, G. et. Al. *Proyecto de investigación adaptative software development*. Universidad Latina de Costa Rica. Recuperado el día 07/04/15 de <http://www.academia.edu/5002829/ASD>

Telefónica S.A. (2015). Quienes somos. Recuperado el día 17/06/2015 de [http://www.telefonica.com.ve/venezuela/telefonica\\_vzla\\_quienes.asp](http://www.telefonica.com.ve/venezuela/telefonica_vzla_quienes.asp)

## MANUAL DE USUARIO

La configuración de una nueva promoción de prepago se debe realizar en la Base de Datos IVR, en ella se debe configurar toda la información necesaria para que la misma otorgue al cliente todos los elementos promocionales solicitados en bitácora, los cuales pueden ser bonos, descuentos sobre la renta de algún servicio o el otorgamiento de una línea gratis para los nuevos clientes. El tiempo y las condiciones para que el cliente disfrute de estos elementos son definidos por el negocio.

El siguiente manual tiene como finalidad explicar cómo realizar una promoción de prepago de Telefónica S.A. en la aplicación GesPromos. Por medio de esta herramienta, el desarrollador puede realizar la configuración y la gestión de la promoción sin la necesidad de configurar los scripts solicitados por el negocio. Sin embargo, debe estar en capacidad de identificar en la bitácora todos los elementos y las condiciones solicitadas por el negocio.

Por medio de GesPromos, es posible obtener los identificadores principales de la promoción, configurar los scripts indicando los datos principales y realizar las pruebas de los mismos en la Base de Datos IVR de desarrollo, incluyendo las evidencias del hecho. A continuación, se realiza una descripción detallada de las funcionalidades que otorga la aplicación:

En su pantalla de inicio, se pueden encontrar tres (3) secciones: Configuración, Prepago y Ayuda.

Configurador de Promociones    Inicio    Acerca de    Iniciar sesión

# GesPromos

**Configuración**  
Desarrolle de forma rápida y sencilla la configuración de promociones de telefonía prepago o seleccione una configuración realizada previamente.  
Es necesario iniciar sesión para acceder a los servicios de Gestión de Promociones.

**Prepago**  
Se recomienda revisar el manual de configuración de promociones, con la finalidad de comprender mejor cada una de las opciones a programar.  
Manual de Configuración

**Ayuda**  
Puedes consultar los errores comunes que se presentan en las pruebas de desarrollo de Base de Datos y la herramienta CVSC.  
Pruebas de desarrollo

Para acceder a la sección de Configuración, el usuario debe estar registrado en el sistema. De lo contrario, no podrá realizar ningún tipo de gestión sobre las promociones de prepago. La contraseña debe contener letras mayúsculas, minúsculas, números y símbolos especiales.

Configurador de Promociones 

## Iniciar sesión.

Ingrese su usuario para acceder a la aplicación.

---

**Correo electrónico**

**Contraseña**

¿Recordar cuenta?

[Registrar como nuevo usuario](#)

[¿Ha olvidado su contraseña?](#)

---

© 2016 - Configurador de Promociones - Telefónica Venezuela

Luego de iniciar sesión, la pantalla de inicio permite tener acceso al módulo de configuración:

Configurador de Promociones Inicio Acerca de [mangialomini@gmail.com](#) Cerrar sesión



## Configuración

Desarrolle de forma rápida y sencilla la configuración de promociones de telefonía prepago o seleccione una configuración realizada previamente.

## Prepago

Se recomienda revisar el manual de configuración de promociones, con la finalidad de comprender mejor cada una de las opciones a programar.

## Ayuda

Puedes consultar los errores comunes que se presentan en las pruebas de desarrollo de Base de Datos y la herramienta CVSC.

En la sección de Prepago, se puede acceder al manual de configuración de promociones. En esta parte se indican cuáles son las tablas que deben ser llenadas para configurar una promoción de prepago. Para cada una de ellas se indican cuáles son sus columnas, el tipo de dato que las conforman y cuáles son los posibles valores aceptados.



## Tablas para la configuración de Promociones en PREPAGO

Una promoción es la forma de modificar la facturación o tasación de un concepto, a través de: descuentos, redefinición de tarifas, otorgamiento de cupos/contadores. Pueden tener asociado un cargo mensual por el disfrute de la promoción.

La configuración de una nueva promoción para Prepago se realiza en la Base de Datos IVR, en ella se ingresa toda la información necesaria para se otorgue a los clientes los elementos promocionales y el tiempo de duración de los mismos. Estos elementos promocionales pueden ser bonos o descuentos sobre la renta de algún servicio. La información necesaria para la configuración de las promociones es definida por el negocio y suministrada a través de Bitácoras.

A continuación, se muestran las tablas a configurar al momento de desarrollar una promoción:

### TCONFIG\_PROMOCION

#### TOPERACIONXPROMOCION - Ocultar

Configuración de las operaciones por las cuales se otorga la promoción.

COLUMNA	TIPO DE DATO	NULO	VALOR POR DEFECTO
CPROMOCION	NUMBER(6,0)	NO	

En la sección de configuración, podemos encontrar dos módulos: Crear y Gestionar Promoción. El primero nos permite diseñar una promoción desde cero. El segundo nos permite cargar una promoción almacenada, ya sea para utilizarla de nuevo o para crear una nueva a partir de esta.

## **Crear Promoción**

Cuando accedemos al módulo de Crear Promoción, el primer paso consiste en indicar las condiciones básicas a configurar. Se deben indicar las siguientes acciones:

- **Buscar Promoción:** permite seleccionar un identificador de forma automática. Si decide no utilizar esta opción, deberá indicarlo de forma manual en el siguiente paso.
- **Elementos:** es necesario identificar en la bitácora cuantos son los elementos que se desean configurar. Pueden ser bonos, cargos, bonificaciones de saldo, SVA o una línea gratis. Pueden añadirse hasta veinte (20) elementos por promoción.
- **Modelos:** una promoción puede ser creada para modelos específicos. De ser así, puede indicar hasta doscientos (200). De lo contrario, no debe seleccionar esta opción.
- **Agentes autorizados:** algunas promociones solo son válidas para vendedores específicos, De ser así, debe indicar el número de agentes que tendrá la promoción, teniendo como límite doscientos (200).
- **Código de mensaje:** Puede verificar si un identificador de mensaje se encuentra disponible. Puede realizar esta acción colocando el número que desea consultar. Si se encuentra disponible, este identificador aparecerá en el siguiente paso. En caso contrario, aparecerá un cero (0).

## Condiciones Iniciales de la Promoción

¿Desea recibir un código de promoción de forma automática? Puede ser modificado más adelante.

Buscar código de promoción **Seleccionar para solicitar un código de promoción**

¿La promoción requiere agregar elementos?

**Seleccionar para agregar los elementos**

¿La promoción requiere agregar modelos específicos?

**Seleccionar para agregar los modelos**

¿La promoción requiere agregar agentes autorizados?

**Seleccionar para agregar los agentes**

¿Desea consultar un código de mensaje específico?

**Seleccionar para consultar un código de mensaje**

**Siguiente paso**  **Siguiente**

En el siguiente paso, se debe llenar un formulario para indicar todas las condiciones de la promoción. Debe realizarse un análisis previo de la bitácora para conocer cuáles son los parámetros que debe llevar el requerimiento.

Se deben indicar las siguientes condiciones:

- Código de promoción: si fue seleccionada la opción de buscar promoción en el paso previo, un identificador aparecerá de forma automática. De lo contrario, debe indicarse el número deseado.
- Nombre de la promoción: se indica el nombre que debe llevar la promoción según es indicado en la bitácora.
- Fecha de inicio: se indica la fecha en que comienza la promoción:
- Fecha de cierre: se indica el día en el que finaliza la promoción.

## Configurar Promoción

Código de la promoción: \*  ← **cpromocion**

Nombre de la promoción: \*

← **Nombre indicado en bitácora**

Fecha de Inicio: \*  /  /  ← **dd/mm/aa**

Fecha de Cierre: \*  /  /  ← **dd/mm/aa**

- Líneas de negocio: se debe seleccionar si la promoción es para telefonía fija o móvil.
- Tecnología: se debe indicar si el tipo de tecnología es GSM o CDMA.
- Tipo de promoción: se debe indicar si es de captación o fidelización.
- Restricción de la promoción: se debe indicar si es de carácter obligatorio u opcional.

Líneas de Negocio: \*

Fijo

Móvil

Tipo de Promoción: \*

Captación

Fidelización

Tecnología: \*

GSM

CDMA

Restricción de la Promoción: \*

Obligatoria

Opcional

- Operaciones permitidas: una promoción permite realizar varios tipos de operaciones para ser otorgada. Las opciones son las siguientes
  - AINIC (Activación inicial): se selecciona en las promociones de captación.
  - CSER (Cambio de serial): para aquellas promociones donde el cliente deba cambiar de equipo.

- CPLA (Cambio de plan): esta opción debe ser seleccionada cuando el cliente deba cambiar de plan para recibir los beneficios de la promoción.
- CTEC (Cambio de tecnología): debe ser seleccionado si se requiere un cambio de tecnología para activar la promoción.
- CCOV (Cambio de coverage): la promoción debe requerir un cambio de cobertura para activar esta opción.
- APROM (Activación de promoción): se selecciona esta opción si es necesario que la promoción sea otorgada de forma automática cuando el cliente cumpla las condiciones.
- Valida serial: se selecciona si es necesario verificar que el cliente ha cambiado de equipo.
- Planes asociados a la promoción: se indica el plan que disfrutará el cliente para recibir la promoción.

**Operaciones Permitidas:**

 AINIC - Activación inicial

CSER - Cambio de Serial

CPLA - Cambio de Plan

CTEC - Cambio de Tecnología

CCOV - Cambio de Coverage

 APROM - Activación de promoción

 Se Valida el serial del equipo

Planes asociados a la promoción: \*



**Código del plan destino**

- Elementos de la promoción: en esta sección se debe introducir la cantidad de elementos indicados en el paso anterior. Cada uno contiene:
  - Nombre: se debe indicar el nombre sugerido en bitácora.
  - Código: se indica el identificador del elemento.
  - Tipo: este campo indica cual es el tipo de beneficio que recibe el cliente. Puede ser un bono, un SVA, una bonificación de saldo, un cargo o una línea gratis (para el caso de las promociones de captación).

Elementos de la promoción:

Nombre del elemento

Código del elemento

Tipo de elemento

Nombre indicado en bitácora

0

Feature  Línea gratis  Bono de saldo  Cargo  Bono

Nombre indicado en bitácora

0

Feature  Línea gratis  Bono de saldo  Cargo  Bono

- Duración de la promoción: se indica si la promoción dura días o meses y la cantidad.
- Porcentaje de descuento sobre el plan a migrar: se puede establecer un descuento sobre los SVA del plan que está recibiendo el cliente para recibir la promoción. Debe indicar un número entre cero (0) y cien (100)

Duración de la promoción:

2

Porcentaje de descuento sobre el plan a migrar:

0

%

Días

Mes(es)

Duración: 2 meses

- Código del mensaje: si el código indicado en el paso previo está disponible, el mismo aparecerá de forma automática en este campo. De lo contrario, debe indicar un nuevo identificador.
- Descripción del mensaje: se debe colocar un nombre relacionado a la promoción que se está configurando.
- Mensaje a enviar: se debe escribir el mensaje indicado en bitácora. Este no puede superar los 129 caracteres.
- Condiciones de la promoción: se indican que condiciones deben ocurrir para que el cliente deje de recibir los beneficios otorgados:
  - Admin\_status: el cliente pierde la promoción si cae en un estatus determinado.
  - Cierre\_promo: el cliente deja de recibir el beneficio si se cierra la promoción.
  - Condicion\_adicional: se pierde la promoción si el cliente no cumple la condición de la tabla tdinamicoxpromocion
  - Cplan\_Cmerc\_Cserial: el cliente pierde los beneficios si realiza un cambio de plan, de equipo o mercado.

Configuración del mensaje:\*

Código del mensaje:\*

1954

← Código del mensaje

Descripción del mensaje:\*

Nombre referente a la promoción

Mensaje a enviar:\*

Mensaje indicado en bitácora

Condiciones de la promoción:



ADMIN\_ESTATUS



CIERRE\_PROMO



CONDICION\_ADICIONAL



CPLAN\_CMERC\_CSERIAL

- Equipos de la promoción: se debe indicar el nombre del modelo. En este caso, puede existir un modelo que se presenta en distintos colores o versiones y todas se encuentren incluidas. Por este motivo, se recomienda indicar el nombre del modelo en mayúsculas, con el símbolo ‘%’ para iniciar, finalizar y utilizando el mismo como espacio. Ejemplos: %MOTO%G%, %IPHONE%5%, %SAMS%.

Equipos de la promoción:

Debe añadir un código referente a cada modelo o grupo de modelos asociados a la promoción:

De no añadir ningún modelo, la promoción será compatible con cualquier modelo

% MOTO% G%

% IPHONE% 5%

% SAMS%

- Agentes autorizados: se debe indicar el código de vendedor para cada agente autorizado de la promoción.

Agentes Autorizados:

Debe añadir el código referente a cada agente autorizado para la promoción:

De no añadir ningún agente autorizado, la promoción será compatible con cualquiera

Código del agente (cvendedor)

Código del agente (cvendedor)

Siguiente paso



Crear Promoción

Luego de llenar el formulario con todos los datos solicitados, se procede a crear la promoción.

En el siguiente paso, comienza la gestión de la promoción. Son permitidas las siguientes acciones:

- Descargar scripts de configuración de la promoción.
- Descargar scripts de contingencia.
- Editar Datos: permite modificar los datos ingresados en el paso anterior, en caso de ser necesario.
- Instalar promoción en BD: esta acción permite ejecutar los scripts de configuración en la Base de Datos IVR de Telefónica. Luego de ejecutar la instalación aparecerán nuevas opciones para la gestión de la promoción.
- Guardar y Salir: permite almacenar la promoción configurada para ser utilizada en futuros desarrollos.

## Gestión de la Promoción

La promoción ha sido configurada. A continuación puede descargar los archivos correspondientes a la promoción. Si lo desea, puede ejecutar los scripts en la Base de Datos de Desarrollo. No olvide guardar la promoción si desea reutilizarla en futuras configuraciones.



Al finalizar la instalación de la promoción, el botón determinado para esta opción desaparece. A su vez, aparecen dos acciones nuevas para realizar:

- Descargar Spool Instalación: permite obtener un registro con los resultados de la instalación en la base de datos IVR de Telefónica.
- Instalar contingencia en BD: al realizar esta acción, el script de contingencia es ejecutado en la base de datos IVR de Telefónica.

## Gestión de la Promoción

El script de configuración ha sido ejecutado en la Base de Datos de Desarrollo. A continuación, puede descargar el spool generado. Si así lo desea, puede instalar la contingencia y deshacer los cambios.



Luego de instalar la contingencia, se puede descargar el registro con las evidencias de la ejecución de los scripts. Con esta acción se da por terminada la gestión de la promoción. De no estar conforme con los resultados, puede editar los datos de la promoción y comenzar el proceso nuevamente.

## Gestión de la Promoción

El script de contingencia ha sido ejecutado en la Base de Datos de Desarrollo. A continuación, puede descargar el spool generado. No olvide guardar su configuración si desea utilizarla en otro momento.



### Gestionar Promociones

En el segundo módulo de configuración, se pueden buscar o eliminar las promociones disponibles en la base de datos de GesPromos. Para iniciar la búsqueda, se cuenta con una vista previa de los requerimientos almacenados. Para cargar una promoción, la misma puede seleccionarse de una lista o buscarla mediante su identificador.

## Gestión de Promociones

Existen 3 promociones disponibles.

Código de promoción	Nombre de la Promoción
1001	Promocion de prueba 1001
1004	Promocion de prueba 1003
1002	Promocion de prueba 1002

Cargar Promociones 

Buscar por Código 

Eliminar Promociones 

- Cargar Promoción: esta opción muestra una lista con las promociones almacenadas. Se puede seleccionar una de ellas y proceder a su edición.

### Cargar Promoción

Seleccione la promoción que desea cargar:

Selecciona	Código de promoción	Nombre de la Promoción
<input checked="" type="radio"/>	1001	Promocion de prueba 1001
<input type="radio"/>	1004	Promocion de prueba 1003
<input type="radio"/>	1002	Promocion de prueba 1002

Cargar Promoción

- Buscar Promoción: si desea buscar una promoción directamente por su código, puede indicar el identificador en esta sección.

## Buscar Promoción

Indique la promoción que desea cargar:

Luego de obtener la promoción que se desea editar, GesPromos se redirige al módulo de crear promociones. Se realiza la configuración desde cero, con la diferencia que los valores de la promoción solicitada son cargados de forma automática en los formularios.

Es importante indicar en las condiciones iniciales si se desean reutilizar los identificadores o buscar unos disponibles. Si no se cambia ningún dato, se obtendrán los mismos scripts de configuración. De lo contrario, se pueden obtener nuevos archivos a partir de requerimientos anteriores.

- Eliminar promoción: permite eliminar una promoción de la base de datos de GesPromos, seleccionando en una lista el identificador de la misma.

## Eliminar Promociones

Seleccione las promociones que desea eliminar:

Selecciona	Código de promoción	Nombre de la Promoción
<input checked="" type="radio"/>	1001	Promocion de prueba 1001
<input type="radio"/>	1004	Promocion de prueba 1003
<input type="radio"/>	1002	Promocion de prueba 1002

Luego de realizar la configuración y gestión de pruebas de una promoción, la misma debe ser probada en otra aplicación llamada CVSC. Sin embargo, esta aplicación puede indicar que la

promoción presenta errores. En muchos casos estos errores se presentan en tablas ajenas a las configuradas para este tipo de requerimientos. Por este motivo, en la sección de ayuda (se encuentra en la página principal) se muestran los errores más comunes que pueden registrarse en la aplicación de pruebas.



## Errores comunes en pruebas de desarrollo

A continuación, se enumeran algunos errores que suelen ser comunes en el proceso de pruebas de una promoción, además de la posible solución a dicho problema.

Pruebas en CVSC de Desarrollo:

CVSC El ICCID se encuentra asignado a otra cuenta en Prepago o PostPago.

Solución: Hay que buscar en la Base de Datos de CIES en test un serial adecuado, debido a que en desarrollo no están actualizados.

CVSC - Verificar los servicios que se desean eliminar y/o activar ya que existe alguna inconsistencia (-651).

Solución: Verificar los servicios que se le están asociando al plan, porque puede venir alguno que no debe estar. Verificar que los SVA que se muestran en Desarrollo corresponden a los servicios de producción o testing, ya que por lo general los ambientes no están homologados.

ORACLE – No se pudo obtener el tipo de serial (-20200).

CVSC Desarrollo presenta este error con algunos seriales de equipos. Se debe introducir otro serial o probar con otro equipo.

CVSC No se ha encontrado descripción para el error -20586. Contacte al administrador del sistema (-20586).

No se ha encontrado solución para este error en desarrollo. Se presentó al activar dos promociones

## MANUAL DE TECNOLOGÍA

En la actualidad, existe una gran cantidad de tecnologías, herramientas y librerías que permiten crear aplicaciones web, otorgando cada vez más funcionalidades a los desarrolladores. El siguiente manual tiene como finalidad señalar las tecnologías utilizadas para llevar a cabo el desarrollo del Trabajo Especial de Grado: **Aplicación web para la gestión de los procesos de configuración de promociones en una empresa de Telefonía.**

La aplicación desarrollada con el nombre de GesPromos, fue desarrollada en su totalidad en el entorno de desarrollo de Visual Studio 2015, agregando además un conjunto de tecnologías que permitieron añadir las funcionalidades establecidas para cumplir con el proceso de configuración y gestión de las promociones solicitadas por Telefónica S.A.

A continuación se describen en detalle cuales fueron las tecnologías utilizadas, su versión y por qué fueron escogidas para realizar el desarrollo de la aplicación:

- Microsoft Visual Studio Community 2015. Versión 14.0.24720.00 Update 1.  
Versión instalada: Community.

La aplicación web fue desarrollada por medio de este entorno de desarrollo creado por Microsoft, aprovechando las funcionalidades y herramientas que el mismo ofrece, entre las cuales destacan:

- ✓ Asistencia en tiempo real al momento de escribir el código.
- ✓ Contextualiza el código de forma automática, proporcionando mayor orden en el desarrollo.
- ✓ Servicio de ayuda automático para comprender el código utilizado.
- ✓ Solucionador de errores mediante acciones rápidas.
- ✓ Refactorización del código desarrollado.
- ✓ Integración con otras herramientas web (HTML5, CSS3, ASP.NET, etc.).

Esta aplicación fue seleccionada debido a que es la edición más moderna de Visual Studio, además de ser gratuita para desarrolladores individuales e investigación académica.

- Microsoft .NET Framework. Versión 4.5.2

Microsoft Visual Studio Community 2015 mantiene un control sobre las versiones del Framework .NET, las cuales se encuentran integradas a la aplicación. Por medio de Razor, fueron desarrolladas las vistas y otras funcionalidades. Fue seleccionado por solicitud de Telefónica S.A.

- Visual C# 2015 00322-20000-00000-AA489. Microsoft Visual C# 2015

Visual Studio permite desarrollar aplicaciones bajo tres (3) lenguajes: Visual Basic, C++ y C#. Este último fue seleccionado para llevar a cabo las funcionalidades del proyecto, gracias a las similitudes que presenta con respecto a Java y C++ y la facilidad de aprendizaje a partir de ellos.

- Administrador de paquetes NuGet 3.3.0

Se encuentra integrado a Visual Studio y con la finalidad de administrar los paquetes instalados en la aplicación, los cuales permitieron agregar tecnologías adicionales al proyecto, tales como bootstrap, jquery y SQL Server.

- bootstrap v3.3.6

Este Framework permitió dar el estilo deseado a las vistas de la aplicación, además de adaptarla a cualquier navegador web, incluyendo teléfonos móviles y tablets.

- JQuery. v1.10.2

Algunas funcionalidades del lado del cliente fueron desarrolladas bajo esta biblioteca de javascript. Fue seleccionado para agregar algunas funcionalidades del lado del cliente.

- ASP.NET Web Frameworks and Tools 2012.2 v4.1.41102.0
- ASP.NET Web Frameworks and Tools 2013 v5.2.30624.0
- ASP.NET y herramientas web 2015.1 (Beta8) v14.1.11106.0

Estos frameworks y herramientas web se encuentran integrados dentro de Microsoft Visual Studio Community 2015, con la finalidad de administrar las funcionalidades de .NET.

- SQL Server Data Tools v14.0.50616.0
- Microsoft SQL Server Data Tools

Para agregar una base de datos local de SQL Server, fue agregado este paquete adicional, que permite crear y administrar un repositorio desarrollado en este lenguaje. Fue seleccionado este motor de base de datos para seguir con los estándares de Telefónica S.A, los cuales utilizan este sistema y el de Oracle.

- Oracle

Para gestionar las promociones configuradas, se realiza una conexión con la base de datos de Telefónica S.A, la cual posee las siguientes características:

- ✓ Oracle9i Enterprise Edition Release 9.2.0.8.0 - 64bit Production
- ✓ PL/SQL Release 9.2.0.8.0 – Production
- ✓ TNS for Solaris: Version 9.2.0.8.0 - Production
- ✓ NLSRTL Version 9.2.0.8.0 – Production

## CUESTIONARIO DE USABILIDAD DE GESPROMOS

El presente cuestionario tiene como finalidad conocer su opinión con respecto a la usabilidad de GesPromos. La cual se refiere a la efectividad, eficiencia y satisfacción con la que la aplicación puede ser comprendida, utilizada y ser atractiva al usuario.

Se agradece seleccionar la puntuación que corresponda a su opinión de acuerdo a la siguiente escala:

Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
1	2	3	4	5

N°	Ítem	1	2	3	4	5
1	Los títulos, menús y textos de la aplicación son legibles					
2	Los colores utilizados no permiten la percepción adecuada de la aplicación.					
3	Se ofrece una organización que facilita la navegación					
4	Los descriptores de la aplicación no son precisos					
5	El diseño de la aplicación favorece su uso					
6	No pude entender la aplicación sin utilizar el manual de usuario					
7	Los elementos de ayuda disponibles en la aplicación son útiles					
8	El tiempo de respuesta de la aplicación es muy lento					
9	La aplicación ofrece los resultados deseados					
10	Necesito aprender muchas cosas para poder utilizar la aplicación					