



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Envío de notificaciones SMS
a clientes bancarios de
remesas masivas**

TESIS

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Carlos Manuel Cerecedo Aguilar

DIRECTOR DE TESIS

M. I. Juan Fernando Solórzano Palomares



Ciudad Universitaria, Cd. Mx., 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A mi mamita, Paty Aguilar:

*La diferencia entre 'regañó' y 'regalo' es una letra, pero conocer el lustre que representan es un mérito que solo una persona tan sabia como tú podría transmitirme. En cada uno de los minutos que he vivido a tu lado tú siempre has dado todo para que yo sea una mejor persona. . . cada **regañó** valió la pena, y cada **regalo** tuyo es una lágrima de mi sensibilidad, porque siempre confiaste en mí como si no hubiera mañana y casi más que yo mismo. Te prometo engrandecer tu legado, y esta es la **primer evidencia**. Gracias mamita.*

A mi papá, Carlos Cerecedo Díaz:

*La persona más sensata del mundo me ha dicho siempre que tu fuiste **el mejor hombre** que se pueda hallar en el azar de la vida y sus vivencias me hacen sentirte tan cerca de mí como lo está ella todos los días, por eso decidimos amarte para toda la eternidad. Aún cuando el destino me prohibió abrazarte tantas veces cuando lo necesité, yo sé que estás orgulloso de mí, porque tu luz siempre estará para guiarme. Este **logro** es para ti, porque llegar a ser **el mejor hombre**, como tú, es algo que nunca podré **lograr**. Te amo, pa.*

A mi abuelita, Tere Vega:

*Tu **cariño eterno**, la suavidad de tus caricias y la delicia de tu comida son los **tesoros** que más extraño en tu ausencia. . . porque el único objetivo de tu vida siempre fue darnos más a aquellos que por ti vivimos y jamás pensar en ti misma, esa es la enseñanza más grande que jamás igualará ninguna otra persona. Porque te amo tanto como si te hubiese abrazado apenas ayer, fuiste el pilar más fuerte de mi **infancia** y el motor más grande que forjó mi **humildad**. Te extraño mucho abue.*

A mi segundo papá, Pepe Velasco

*Tu eres el ejemplo más claro de que la vida nunca es **justa** con las mejores personas, pero aquellos que tuvimos el privilegio de convivir contigo, sabemos que podemos renunciar a esa justicia y ser felices al seguir tus imborrables pasos. Yo seré dichoso de convertirme algún día en el magnánimo padre que fuiste conmigo, porque para mí siempre será un honor tener el linaje **Velasco** en mi educación. Si yo tuviera la oportunidad de volver a elegirte como el hombre más **importante** de mi vida, sin duda lo haría, porque tu impacto en mi persona me formó como **hombre**. Muchas gracias, Pepe.*

Tabla de contenido

1. Objetivo	3
2. Definición del problema.....	3
3. Introducción	4
4. Justificación y alcance del sistema	5
5. Marco teórico	7
6. Antecedentes del tema	10
7. Entorno actual del problema	11
8. Análisis y metodología propuesta	12
8.1. Desarrollo de metodología y equipo de trabajo.....	18
8.2. Tecnologías propuestas	22
8.3. Susceptibilidad a cambios	23
8.4. Algoritmo del programa	24
9. Solución propuesta.....	25
9.1. Requerimientos funcionales.....	27
9.2. Contexto del sistema	28
9.3. Requerimientos no funcionales	29
9.4. Riesgos.....	30
9.5. Involucrados.....	30
9.6. Casos de uso.....	31
9.7. Diagrama de casos de uso básico.....	31
9.8. Diagrama de casos de uso refinado	32
9.9. Arquitectura	33
9.10. Manejo y almacenamiento de información	34
9.11. Diseño de interfaz gráfica	35
9.12. Aplicaciones.....	38
9.13. Cronograma de trabajo.....	38
9.14. Arquitectura detallada	39
9.15. Pruebas e iteraciones.....	39
10. Participación profesional	42
10.1. Asignaturas involucradas	44
11. Resultados obtenidos.....	44
12. Conclusiones	47
13. Apéndices.....	49
14. Glosario	51
15. Referencias	54
16. Bibliografías	55

Envío de notificaciones SMS a clientes bancarios de remesas masivas

1. Objetivo

Dar solución a una necesidad en concreto de una institución bancaria. Para ello, se propondrá la definición y el diseño de un sistema que cumpla ciertas tareas específicas, dentro de los diversos sistemas del banco ya existentes. Valía

Se planteará una solución concreta y estructurada para resolver un problema específico dentro del sector bancario, enfocado a la siguiente necesidad:

Enviar notificaciones mediante mensajes de texto SMS a clientes de servicios bancarios que utilizan tarjetas de débito de remesas masivas.

A partir de un sistema bancario existente y en operación, es necesario enviar notificaciones de mensajes de texto a ciertos portadores de tarjetas de débito bajo ciertas condiciones, para cumplir con una regulación bancaria establecida por el gobierno.

2. Definición del problema

Una entidad bancaria requiere cumplir con normatividades de seguridad y regulación gubernamental recabando datos de clientes para crear y complementar su expediente personal dentro del banco. El problema se presenta cuando dichos clientes obtuvieron acceso a tarjetas de débito del banco mediante un tercero, mismo que, por diversas necesidades, asignó dichas tarjetas a clientes, sin la necesidad de presentarse en el banco a solicitarlas en persona.

Dado que los clientes ya cuentan con la tarjeta plástica de acceso y hacen uso de ésta, tienen establecido un plazo máximo de 90 naturales días para que el portador de la tarjeta se presente en la sucursal bancaria que la emitió con su documentación de identidad y se formalice su expediente.

Cuando el cliente hace uso por primera vez de su tarjeta, debe pedírsele que acuda al banco, dentro del plazo mencionado. Ante esta situación, se recomienda emplear un sistema automatizado que cumpla con todo lo especificado anteriormente, y debido a que no hay un conocimiento definido de cuándo se entregarán las tarjetas, ni tampoco a quién les serán entregadas con exactitud, y el camino más directo para concretar la comunicación y recolección de los datos necesarios con los portadores es a través del uso de los sistemas bancarios existentes.

Es indispensable atender esta problemática debido a que los organismos gubernamentales que requieren la regulación de los servicios financieros para todos los clientes de los bancos que incumplan con estas leyes, tienen la facultad legal de multar y/o suspenderles la prestación de sus servicios. Con la propuesta que aquí se plantea, se evita la duplicidad de identidades bancarias y se fomenta la integridad de los expedientes de los usuarios de servicios financieros, lo que garantiza mayor calidad en la gestión de los datos y manejo de la información.

3. Introducción

En la actualidad, la computación representa una de las áreas con mayor desarrollo tecnológico directo en el mundo. El crecimiento de la computación en los últimos 30 años provocó que el rumbo de la humanidad fuese totalmente diferente de lo que había sido hasta la primera mitad del siglo XX¹. Sectores como el automotriz, telecomunicaciones, turismo, deportes, comercio, salud, aeroespacial, televisivo, educativo y otros obtuvieron acceso a tecnologías derivadas del cómputo que provocaron que las industrias y gobiernos más importantes del mundo volcaran su interés a los resultados de este desarrollo. Así como también, orientaron la inversión de recursos para utilizar la computación en beneficio propio, lo que a la larga derivó en que muchas tecnologías hoy puedan estar al alcance de los particulares.

Con base en lo anterior, se deben identificar plenamente los diversos campos en donde está presente la computación, los cuales tienden a demandar soluciones propias a su permanente crecimiento. Es decir, el desarrollo tecnológico en el mundo fomenta y facilita numerosos ámbitos de la vida humana, brindando una mejor calidad de vida, produciendo bienes y servicios y alcanzando objetivos diversos², sin embargo, a su vez, estos desarrollos requieren ser atendidos y apoyados para garantizar factores no funcionales dentro de cada campo tecnológico, como confiabilidad, seguridad, integridad, rentabilidad y otras necesidades específicas que son detectadas conforme avanza el tiempo y crece su relación e importancia con la humanidad.

Para el planteamiento de este documento, se analizará la manera en cómo se estaría solucionando una necesidad a un problema de seguridad en una institución bancaria mediante el desarrollo de un software que establezca las condiciones propicias para mitigar el riesgo de fraudes bancarios y/o de administración de recursos en México. Este sistema estaría siendo integrado a su vez, a otro sistema bancario más grande y con múltiples objetivos, pero que carece de condiciones operativas para el control y la eliminación del riesgo mencionado.

Esta situación manifiesta claramente que la complejidad de brindar servicios bancarios al público en general, crea un enlace que puede abrir caminos para que los usuarios malintencionados de dichos servicios puedan vulnerar algunas condiciones funcionales de los

servicios que disponen, y con ello, logren tener acceso a mecanismos que faciliten la comisión de actividades ilegales y/o fraudes en contra de otros particulares o del Estado Mexicano. Por ello, es necesaria la creación de sistemas que participen en conjunto con los servicios bancarios brindados al público, y que tengan como objetivo principal incrementar las medidas de seguridad, prevenir prácticas ilegales e identificar a plenitud la identidad de los usuarios para que las instituciones financieras mantengan el control de las plataformas electrónicas que ofrecen a sus clientes, así como sus recursos y los de terceros.

4. Justificación y alcance del sistema

Las instituciones que prestan servicios financieros son organismos que tienden a permanecer como una industria con vidas extensas y actividades propias a largo plazo, además de que las relaciones que tienen con los gobiernos de su país, gobiernos internacionales y organismos en colaboración, son altamente impactantes en sus condiciones operativas y servicios disponibles. Elementos como leyes e indicadores financieros provocan que los productos y plataformas financieras permanezcan en constante cambio. Sin embargo, en la actualidad, uno de los factores más influyentes en la industria bancaria es la computación, ya que la mayor parte de la información que posee un banco, así como la distribución de sus redes de sucursales y el monitoreo permanente sobre los recursos que genera y administra, son controlados casi totalmente por sistemas de cómputo creados con estas finalidades específicas. A su vez, estos sistemas requieren de regulaciones y controles de seguridad interna y pública para evitar la comisión de cualquier actividad inadecuada basada en los lineamientos de dicha institución.

Todos y cada uno de los cambios en materias de leyes y seguridad bancaria han sido generados a lo largo de décadas, siempre que se conocen nuevos modos en los que los criminales operan, además de las estrategias que constantemente aparecen para vulnerar al banco y obtener beneficios de manera ilegal. Esto genera que la computación también sea una herramienta que facilita el establecimiento de filtros y condiciones de seguridad dentro de los mismos sistemas bancarios, ya sea para perfeccionar su funcionamiento o para minimizar fallas de seguridad de cualquier índole.

Las metas establecidas al plantear un proyecto que resuelva el problema de un requerimiento de seguridad bancaria, a través del cómputo, deben tener la característica de solucionar la(s) necesidad(es) con la mayor inmediatez posible. Esto es debido a que los sistemas de cómputo van a ser lanzados en operación plena y sí, y sólo sí, ya han superado las pruebas de funciones para las que fueron diseñados. De igual manera, en un ambiente en donde el control de calidad del software es funcional, se sabe que el producto terminado no puede tener fallas al ser instalado.

La forma en como impacta un software nuevo dentro de un ambiente con otro software ya en operación siempre debe preponderar los beneficios sobre los posibles riesgos. Y si bien, es

fundamental considerar planes de mantenimiento continuos a un software existente, se debe identificar plenamente que una actualización o mejora de dicho software es un proceso que implica cambios operativos que pueden distar de un plan de mantenimiento programado. Es aquí donde, inclusive, los proyectos de software que incrementan las medidas de seguridad deben ser considerados como un proyecto ciento por ciento nuevo, y generado desde cero. Si bien, en la mayoría de los casos se conoce y se diseña un software a partir de necesidades ya determinadas, dichas necesidades tendrían que funcionar y/o interactuar con el software previamente existente, por lo que este nuevo proyecto debe estar basado y adaptado al entorno funcional para evitar generar conflictos.

Las justificaciones concretas para este proyecto son:

- Cumplimiento con una regulación global a nivel nacional por parte de un organismo gubernamental enviada a todas las corporaciones que prestan servicios financieros al público en general.
- Enriquecimiento y actualización de la base de datos de clientes portadores de tarjetas de débito.
- Minimizar significativamente los riesgos ya detectados de aplicación de actividades ilegales y/o suplantación de identidad en cierto grupo identificado de portadores de tarjetas de débito.

El alcance específico para este proyecto justifica su diseño y creación condicionado a cubrir dos parámetros definidos:

- Identificar las cuentas vinculadas a remesas masivas, donde se verifica el requisito de actualización de datos personales, se validan los períodos establecidos para la actualización de datos personales, el envío de mensajes de texto a dispositivos móviles con recordatorios de actualización de datos personales dentro de los intervalos definidos y finalmente, la cancelación de tarjetas de débito vinculadas a cuentas cuyos clientes no actualizaron sus datos correctamente en el plazo establecido
- La integración completa y satisfactoria de este proyecto dentro del actual de sistema bancario

El estricto cumplimiento del contenido del primer punto se encontrará detalladamente descrito para cada acción en el apartado de requerimientos funcionales en el presente documento.

De igual manera, el segundo punto refiere a que el nuevo sistema, en conjunto con los cambios necesarios al entorno actual, debe tener una funcionalidad que no perjudique ninguno de los sistemas con los que se relacione.

5. Marco teórico

Las telecomunicaciones y los SMS

En este siglo, las telecomunicaciones han brindado a cualquier persona y organización la



Figura 1. La primera generación de telefonía móvil, en los años 80's.

posibilidad de minimizar las barreras de tiempo y distancias de manera efectiva, global y a bajo costo. La llegada de los teléfonos celulares permitió que cada portador esté comunicado permanentemente a la red mundial de información y crear un canal de acceso a datos personales que ha hecho posible crear un vínculo entre dos o más sistemas involucrados en un mismo objetivo.³

Los orígenes de las telecomunicaciones móviles dieron lugar al desarrollo de diversas tecnologías a lo largo de su evolución. Con el nacimiento de los sistemas de telefonía móvil digital en la década de los 80s y su mejora tecnológica en la década de los 90s bajo la denominación 'GSM' —acrónimo en inglés de Global System for Mobile communications y su traducción al español, sistema global para las comunicaciones móviles— se logró ampliar el panorama de telecomunicaciones que estaba limitado a la voz, y con ello implementar un segundo canal de comunicación dedicado a texto, que recibió el nombre de Short Message Service (SMS) o su traducción al español, servicio de mensajes cortos.⁴

Los SMS son cadenas de texto de hasta 160 caracteres que viajan en un canal paralelo en las redes de telefonía y dado que necesariamente son triangulados a través del operador de telefonía, tienen la ventaja de ser almacenados por dichos operadores (durante su periodo de vigencia) hasta que pueden ser recibidos por el destinatario. Inicialmente, los SMS fueron



Figura 2. El servicio de mensajes cortos fue muy popular en la primera década del siglo XXI, cuando fue liberado para comunicación directa entre usuarios.

diseñados exclusivamente para dar a conocer información sobre el servicio a los abonados de la operadora, pero años más tarde, la firma Finlandesa Nokia desarrolló un sistema que ha permitido que los mensajes sean bidireccionales, y con esto, facilitar que en las nuevas versiones de teléfonos celulares los clientes también puedan escribir y enviar mensajes de texto al operador y hacia otros usuarios de telefonía.



Figura 3. El uso de plásticos bancarios consolidó un eficiente mecanismo de disposición de efectivo en la primera mitad del siglo XIX.

Si bien, en la actualidad existen múltiples aplicaciones más modernas y basadas en Internet que se encargan de enviar notificaciones, e incluso los bancos han diseñado sus propias aplicaciones para facilitar la gestión de cuentas bancarias por parte de los clientes, el sistema de notificaciones a través de SMS es un servicio que mantiene la gran ventaja de ser compatible con cualquier teléfono celular conectado a cualquier red móvil digital y no tener costo para quién recibe el mensaje.

Las notificaciones mediante SMS por actividad bancaria son un elemento valioso en la comunicación necesaria entre una institución que brinda diversos servicios y un cliente, dado que permiten a la institución bancaria informar de cualquier tipo de movimiento en las cuentas personales de las personas. En la actualidad, los medios digitales permiten este vínculo con mayor facilidad, y, sumado a la múltiple gama de usos que tienen los teléfonos celulares, se simplifica la tarea de localizar a los clientes en cualquier momento y en cualquier lugar, porque ellos tienen una terminal con características estandarizadas que permiten establecer comunicación global inmediata.

Las remesas masivas

Una cuenta bancaria es un servicio ofrecido por un banco hacia personas físicas o personas morales que brinda la posibilidad de almacenar fondos en el banco con la finalidad de acceder a ellos posteriormente,

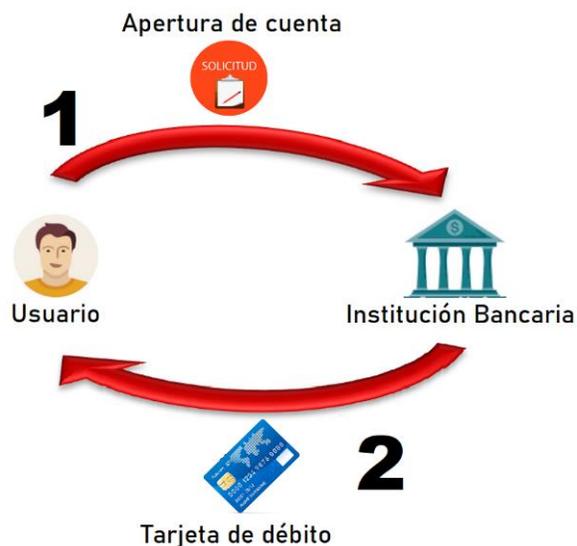


Figura 4. Procedimiento habitual de obtención de una tarjeta de débito.

utilizando diversos medios, como son ventanilla bancaria, cheques, aplicaciones electrónicas y cajeros automáticos mediante tarjetas de débito.

Habitualmente, cuando un cliente (persona física) apertura una cuenta bancaria personal, ésta se vincula a una tarjeta de débito para que pueda disponer de los fondos a través de retiros en cajeros automáticos. Sin embargo, la variedad de servicios de pago ofrecidos por parte de diversas empresas u organismos públicos gubernamentales, han obligado a diversificar el acceso a una tarjeta de débito personal a usuarios que no abrieron una cuenta bancaria de forma tradicional. Es en estos casos que surge la necesidad de crear cuentas bancarias de forma anticipada y, previamente vinculadas a tarjetas de débito predeterminadas, porque el objetivo es entregar una tarjeta de débito a un cliente y así pueda comenzar a disponer de los fondos inmediatamente. En este procedimiento, la tarjeta de débito no es entregada directamente por el banco emisor a petición expresa de un cliente, sino que inicialmente es asignada a una empresa o instancia gubernamental, posteriormente, dicha empresa asigna las tarjetas a los beneficiarios para hacerles llegar fondos periódicamente y, de esa forma, éstos puedan ser dispuestos a través de esas tarjetas de débito. Este procedimiento denomina a las tarjetas involucradas como tarjetas de “remesas masivas”.

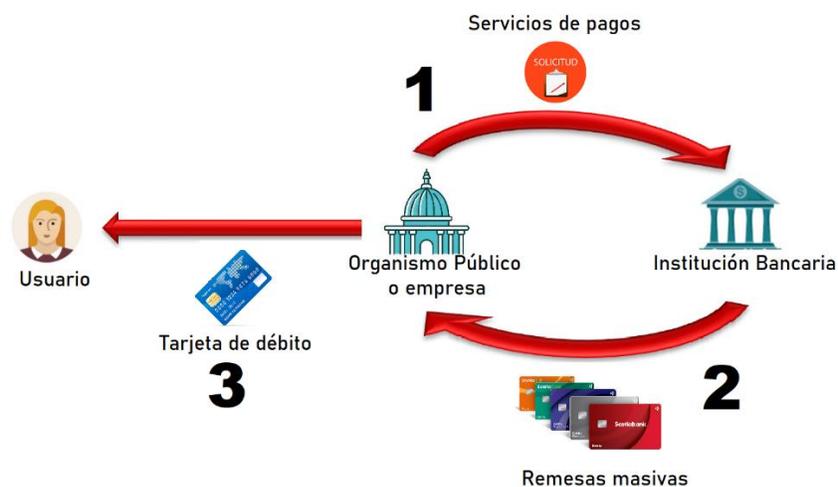


Figura 5. Procedimiento de asignación de tarjetas de débito de remesas masivas.

Cabe señalar que, por lo general, las tarjetas de remesas masivas son empleadas para entregar recursos a los beneficiarios que el organismo solicitante designe. Sin embargo, en la actualidad también son utilizadas para operaciones electrónicas, en donde el portador del “plástico” inicialmente realiza depósitos, y posteriormente dispone del dinero utilizando diversos medios electrónicos, tanto nacionales como internacionales.

6. Antecedentes del tema

Los múltiples servicios bancarios han evolucionado a lo largo de las últimas décadas con la finalidad de aumentar la eficiencia y elevar la variedad de servicios que se puede ofrecer a los clientes, ya sean personas físicas o morales. Algunos de los avances más significativos que se han logrado brindar en servicios financieros a distancia, se obtuvieron por la consolidación de las redes de telecomunicaciones y particularmente el internet. La minimización de barreras acrecentó la facilidad en que los clientes pueden interactuar con los servicios bancarios que ofrece una institución, y de esta forma, han logrado conseguir optimizar el manejo de sus fondos y más fácilmente tomar las decisiones que involucran su gestión económica.



Figura 7. En la actualidad, las operaciones electrónicas de transferencia de fondos son instantáneas, sin restricción de horarios y a nivel mundial.

Considerando que muchos de los avances en cuestión de manejo de dinero han significado operaciones remotas, movimientos o autorizaciones que no requieren la presencia física del (los) titular(es) de las cuentas en el lugar de las operaciones, se han identificado diversas formas de fraudes, evasión fiscal o actividades delictivas en general, por parte de delincuentes que aprovechan las ventajas y facilidades electrónicas que ofrecen las instituciones bancarias a sus clientes para cometer diversos delitos.



Figura 6. La banca por internet ofrece soluciones inmediatas y personalizadas a las necesidades de los clientes.

A lo largo de la última década, la Comisión Nacional Bancaria y de Valores ha aprobado una serie de regulaciones dirigidas al sector financiero que exige a las instituciones bancarias crear y/o mejorar los sistemas existentes para buscar mitigar los problemas de fraudes bancarios.⁵ Estos sistemas incluyen diferentes cambios en la administración

de los sistemas bancarios, por ejemplo, el intercambio de múltiple información entre los bancos y las entidades emisoras de identificaciones oficiales, la creación de aplicaciones personales para teléfonos inteligentes, la toma de elementos biométricos de identificación, y el envío de notificaciones en tiempo real por cada movimiento que se realice en las cuentas de los clientes.

En el caso particular de los sistemas de notificaciones, la idea principal de esta técnica es que los titulares de las cuentas bancarias estén informados en el momento exacto que una acción se presenta en sus cuentas, ya sea un abono, un cargo, actualización de datos, retenciones o en general cualquier actividad que se realice en sus cuentas. La implementación de notificaciones fue una iniciativa que redujo significativamente el índice de fraudes y acciones delictivas relacionadas con personas que son víctimas de suplantación de identidad, clonación de tarjetas, sustracción de fondos, y en general cualquier otro abuso que represente cualquier tipo de problema al cliente o inclusive, pérdidas para los bancos.

7. Entorno actual del problema



Figura 8. Los delitos informáticos relacionados con fraudes bancarios son los principales mecanismos que provocan fuga de capital.

Una entidad bancaria asigna tarjetas de débito de remesas masivas a otra institución externa al banco. Posteriormente, dichas tarjetas son entregadas a clientes que no tienen antecedentes con el banco emisor, o éstos son escasos. En este contexto, es mandatorio que el cliente se presente personalmente en una sucursal bancaria con documentación oficial vigente, que permita que el banco formalice el expediente de este usuario portador de dicha tarjeta de remesa masiva.

Se sabe que, cuando el beneficiario de la tarjeta de remesa masiva recibe el plástico, lo debe activar en un cajero automático y cambiar el NIP predeterminado por uno personalizado; es en este momento donde el portador debe proporcionar un número de celular a 10 dígitos válido en la República Mexicana. Esta información permitirá que el software que se diseñó en este proyecto contacte con este cliente para recordarle que debe acudir a una sucursal bancaria a actualizar su información personal en un lapso no mayor a 90 días naturales.

Si el portador del “plástico” realiza las actualizaciones de información personal correctamente dentro del plazo establecido, el expediente del usuario estará correctamente estructurado, y, por tanto, el software finaliza su funcionamiento. En caso contrario, se emitirán alertas periódicas para insistir en que el portador realice esta actualización de datos personales, y si no se cumple con este requisito, la tarjeta será desactivada permanentemente y el portador de la tarjeta deberá asistir a una sucursal bancaria cuando desee que la cuenta asignada sea vinculada nuevamente a otro “plástico”.

En la actualidad, la correcta validación de la identidad de una persona física constituye un objetivo primordial en los procesos de seguridad bancarios. Se sabe que en los últimos 10 años, las instituciones bancarias a nivel mundial, han invertido recursos millonarios para evitar prácticas de “pishing” mediante procedimientos que han evolucionado continuamente; se utilizan enlaces directos con instituciones de gobierno emisoras de identificaciones oficiales personales para evitar la falsificación de documentación oficial, y se crean carteras de identificación únicas que comparten los bancos entre sí, y lo más moderno es la utilización de sistemas biométricos como huellas digitales y escáneres de retina para validar la identidad personal de un cliente en casi todos los servicios bancarios. Derivado de todo esto, es claro que, cuando una persona obtiene una tarjeta de débito a través de procedimientos legales, pero no realizó concluyentemente el proceso de identificación bancaria vigente en ese momento, es indispensable que logre culminar estas acciones. Este objetivo es uno de los motivos más claros del diseño de este sistema.

8. Análisis y metodología propuesta

Al comprender que la indicación gubernamental implica acciones proactivas entre la institución bancaria y el cliente, se procede a analizar las áreas de oportunidad que se tienen actualmente para resolver el problema, ponderando que, en las herramientas existentes en el banco, ya haya un mecanismo que pueda resolver las necesidades planteadas.

Al visualizar que no hay un proceso operativo y práctico para resolver estos problemas, se analizaron algunos caminos previos que resultaron ser inviables. Como la suspensión de las remesas masivas, o la realización de una campaña pública en donde se solicite a los portadores de dichas tarjetas cumplir con estas reglamentaciones. Sin embargo, los riesgos, la inconveniencia y la vigencia de los productos bancarios activos, demuestran que estas alternativas deben ser descartadas.

Una vez establecido todo lo anterior, se optó por resolver el problema a través de la creación de un sistema de cómputo que permita cubrir las necesidades ya identificadas, depuradas y concretas. A pesar de que ya se conocen dichas necesidades, éstas todavía se describen a alto nivel, es decir, con lenguaje humano y sin un camino fijo de actividades computacionales que permitan su desarrollo y solución. Por ello, es mandatorio que dichas acciones se puedan precisar y reducir a un camino fijo y ordenado para, posteriormente, definir las funciones específicas que entenderá y ejecutará el sistema.

Es por ello, que se plantean y discuten diversas propuestas de creación de software, estudiando su factibilidad, duración, presupuesto y otros aspectos. Finalmente, se opta por revisar con expertos los 4 mejores métodos de manera formal y estructurada que estarían solucionando las necesidades por diversas vías; se presentan las propuestas en un cuadro que también incluye un breve análisis comparativo de ventajas y desventajas que permitirá validar la factibilidad previa al diseño del software.

***Método:**

Propuesta	Descripción	Ventajas	Desventajas
1	Diseñar un software que permita interactuar en cajeros automáticos. La idea es, que cuando el portador del "plástico" utilice un cajero automático, el cajero notifica al cliente que debe pasar a sucursal bancaria de inmediato para realizar el trámite, de lo contrario no podrá efectuar ninguna operación.	Se garantizará la obtención de los datos requeridos igualando el procedimiento del banco, y los portadores no podrán disponer de los fondos hasta que cumplan con el requerimiento. La integración con el sistema de cajeros es de una sola consulta	Los clientes podrían acudir a cajeros automáticos ubicados en puntos donde no hay sucursal, y/o en horarios inhábiles. Además, la disposición de los fondos obedece a respetar el interés de la institución que asignó la tarjeta y ésta no se puede transgredir.
2	Crear un sistema para vincular información del portador de manera automática a un expediente bancario, basándose en la información personal que pueda recabar la entidad que entregó los plásticos	Los clientes no tienen que participar en ello. Se realizará de forma sistematizada.	Las leyes de protección de datos personales no permiten estas acciones. Asimismo, es posible que haya datos faltantes y/o inconsistencias en los datos. De igual manera, no se podrán obtener datos biométricos.
3	Diseñar un servicio web que permita realizar el trámite de alta formal de los clientes desde internet	El cliente podrá realizar el trámite desde la comodidad de su domicilio, y cuando tenga disponibilidad de hacerlo. Sólo se requiere una notificación en cajero automático	La confiabilidad de la obtención de documentos todavía no está garantizada cuando se realiza por internet. De igual manera, es necesario validar dicha documentación de forma inmediata en diversos niveles dentro del banco. La validación de datos demanda interactuar con muchos sistemas y abrir canales para ello.
4	Diseñar un software que notifique mediante alertas con mensajes SMS al portador de la tarjeta que tiene un plazo máximo para acudir en persona al banco a realizar los trámites requeridos. De lo contrario, el "plástico" se desactivará al exceder dicho período.	Se le concede el plazo máximo establecido al cliente para acudir a sucursal bancaria. El cliente dispondrá de sus fondos de manera normal y sin limitaciones durante este período.	Se le permitirá al cliente interactuar con el plástico con todos los beneficios del sistema bancario que le corresponden durante el plazo de actualización de datos personales

Analizando a profundidad los detalles de las propuestas realizadas y luego de evaluar detalladamente la relación de las ventajas con respecto a las desventajas, el comité encargado de hacer cumplir la regulación gubernamental resuelve que solución más factible es la propuesta #4.

Esta resolución está basada en afirmar y reconocer que la creación de este software es viable, cumple con las condiciones internas del banco, y fomenta el compromiso de los portadores de los plásticos para convertirse en clientes formales del banco; concepto que termina siendo el objetivo primordial de esta reglamentación. Asimismo, los plazos estipulados coinciden con los que señaló el gobierno para cumplir las condiciones, y se podrá aprovechar como valor agregado el hecho de que el software que se diseñará será un sistema ligero, de bajo impacto y poco invasivo, ya que estará inmerso y basado en diversos sistemas que el banco ya tiene operando en la actualidad.

El objetivo a grandes rasgos, de la propuesta #4 establece que será necesario definir un sistema que involucre el aprovechamiento de sistemas bancarios funcionales existentes y utilizar sus servicios en un sistema nuevo cuyo objetivo es notificarle al portador del “plástico” acerca de la necesidad de acudir al banco a presentar documentación oficial dentro de un plazo máximo de 90 días naturales, y con ello regularizar su expediente dentro del banco prestador del servicio.

El sistema cumplirá con diversas acciones y objetivos delimitados que se detallarán a lo largo de este documento, entre los que destacan que deberá ser capaz de identificar la clasificación del “plástico”, para determinar si pertenece a una remesa masiva o fue asignada a través del procedimiento habitual; en caso de ser de remesa masiva, se solicitará que el portador de la tarjeta proporcione un número telefónico móvil y con ello se enviarán alertas periódicas mediante mensajes SMS al número telefónico de dicho portador, para recordarle que es necesario su presencia en sucursal bancaria. Estas notificaciones serán enviadas tres veces: la primera vez, en el momento del alta del número telefónico. La segunda ocasión, 30 días naturales después del alta, y la tercera vez a los 60 días naturales luego del alta. En caso de que el cliente sí acuda a sucursal bancaria a realizar esta actualización de datos, las alertas por este cumplimiento se detienen permanentemente. En caso de que el portador no acuda a realizar esta actualización y el plazo se venza, la tarjeta será inhabilitada, y para recuperar acceso a los fondos de la cuenta bancaria, será necesario que el cliente acuda a restaurar el plástico en una sucursal bancaria. Naturalmente, el proceso de reemplazo de tarjeta conllevará la actualización de datos personales como cumplimiento del procedimiento de formalización del expediente de dicho cliente.

Como ya se ha mencionado con anterioridad, el proyecto propuesto será implementado sobre el sistema bancario ya existente, y afectará servicios de aplicativos que ya operan brindando acciones hacia otros sistemas. Ello representa que se necesitará de diversas modificaciones de dichos aplicativos, y considerando esto, el diseño de este software nuevo

no será a partir de cero totalmente. Se utilizará un proceso de modificación de dichos sistemas y dados los procedimientos para la creación y las modificaciones planteadas, se concluyó que el mecanismo más favorable para alcanzar los objetivos que pretende este nuevo software es conocido como SDLC —System Development Life Cycle— en español, Ciclo de vida de desarrollo de sistemas. Esto se seleccionó por la versatilidad que posee para funcionar como marco de trabajo predefinido, cuyo contenido facilitó las condiciones, procedimientos y herramientas que conjuntan diversas metodologías de software que pueden favorecer a este tipo de circunstancias. **(Ver apéndice #1)**

Las fases de trabajo serán las siguientes:

Levantamiento de requerimientos: Es un proceso mediante el cual se recabarán todas las especificaciones funcionales que debe cumplir el nuevo software. Se realizará a través de varias entrevistas con el (los) departamento(s) que requieren este software. De igual manera, se debe establecer que las condiciones determinadas en dichas entrevistas deben estar dentro del marco de referencia que se establece en las necesidades que demanda la regulación gubernamental que se va a cubrir. En este caso fue un comité quién estableció dichas necesidades y por parte del equipo de este proyecto se responsabilizó al “Business Analyst” y al “Project Manager” para el levantamiento formal de los requerimientos de esta fase.

Diseño: Fase de definición de todo el proceso de trabajo subsecuente para la elaboración del software. Aquí se delimitaron todos los mecanismos sistemas y herramientas que se tienen y las que se necesitan. Así como los algoritmos, y procedimientos detallados que deben cumplir los desarrolladores. Esta fase gira completamente en torno a establecer las maneras en que el software cumplirá los requerimientos funcionales.

Construcción o implementación del software: En esta fase todos los desarrolladores crean el software a partir de los entornos de programación, y generan los cambios a través de la codificación de su respectivo sistema para cumplir con cada requerimiento del software.

Integración: Una vez que todos los involucrados terminaron con el desarrollo en su subsistema, se realiza una implementación global para validar el funcionamiento, características y detectar errores en la unificación.

Pruebas: Este nivel tiene como finalidad validar que el software funcione adecuadamente y que cumpla con los requerimientos establecidos, además de detectar errores. Está dividido en dos fases: la fase de pruebas unitarias en un entorno ideal, en donde el software se instala en un sistema con un ambiente controlado y se realizan pruebas específicas con condiciones ideales. La segunda etapa consiste en generar múltiples condiciones de inestabilidad con respecto a la prueba anterior, buscando simular las circunstancias operativas reales que el

software va a experimentar, como son rendimiento, saturación, pérdida de conexión, fallo de sistemas involucrados, expectativas de otros problemas posibles e incluso, vulnerabilidades.

Iteraciones e incrementos. Esta fase consiste en repetir el ciclo de construcción, integración y pruebas en múltiples ocasiones cuando la etapa de pruebas finaliza. El objetivo es que al finalizar las pruebas haya una retroalimentación basada en los resultados de las pruebas para corregir problemas y nutrir a los desarrolladores para mejorar el software continuamente. No hay una cantidad establecida de iteraciones que realizar, todo se basa en la calidad del software y la nulificación de errores.

Release (instalación). Solamente se llega a esta fase cuando todas las fases previas han sido cubiertas satisfactoriamente o en su defecto, un responsable del proyecto avala esta etapa con inconsistencias previamente detectadas.

Cabe señalar que en el proceso global del proyecto se emitirán recomendaciones para el mantenimiento del software una vez lanzado y durante el tiempo de cierre, así como sugerencias de monitoreo de componentes específicos durante su ejecución en las primeras horas, para buscar fallas esperadas y/o puntos débiles. Esto con la finalidad de reportar a las áreas correspondientes los posibles conflictos con los aplicativos participantes, y minimizar el impacto en la plataforma global del banco. Esta información será incluida en la documentación de entrega que se conservará en el expediente histórico de contenidos de la programación de los sistemas del banco.

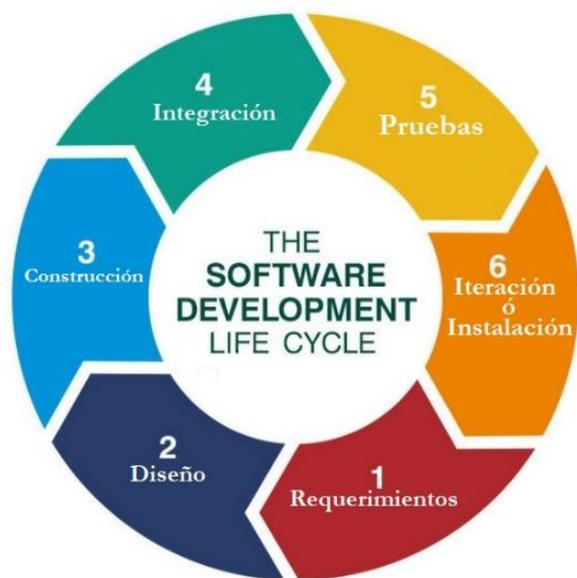


Figura 10. La versatilidad que brinda SDLC, facilita que sea adaptable a las necesidades de la empresa que lo aplica y/o a las condiciones particulares de cada proyecto.

Dada la propuesta de diseño que se obtuvo, se detectó qué aplicativos existentes en concreto serán impactados por el nuevo software, tanto en el proceso del ciclo de creación, como en las diversas actividades que debe cumplir el software de este proyecto para su correcta operación, una vez instalado:

Base de datos: El almacenamiento de información será gestionado mediante un “data warehouse” en la nube. Donde “TERADATA” es el manejador de bases de datos con el que ya se trabaja actualmente. Esta base de datos contiene el perfil completo de los clientes (entre otra mucha información) así como diferentes estatus con respecto a los servicios contratados con el banco.

Servicio de mensajería: El sistema dentro del banco es llamado “TELUS”. La operación del servicio de mensajería consiste en contratar una empresa externa al banco para realizar las actividades de notificación a través de mensajes de texto SMS. “TELUS” es la terminal que se conecta con esta empresa externa y opera con un sistema automatizado que únicamente requiere el texto que debe contener el mensaje y el número de destinatario. La empresa externa al banco fue contratada para múltiples finalidades que tienen que ver con notificaciones SMS, y tiene mucho tiempo cumpliendo estas funciones; su actividad no será exclusiva para este proyecto. La comunicación a este servicio consiste en enviar un archivo único activador en formato “XML” para disparar la acción de enviar el mensaje. Tanto el contenido del mensaje, como información del destinatario, son plantillas creadas previamente con texto predeterminado.

Sistema de alta/baja de tarjetas: Es una plataforma que permite gestionar permanentemente el “status” en tiempo real de sus todas las tarjetas de crédito y débito que tienen los clientes del banco. Únicamente recibe condiciones a manera de instrucción de otros sistemas bancarios, y sus respuestas impactan directamente en el manejador de base de datos enviando exclusivamente “datos booleanos” para activar o desactivar una tarjeta específica. Trabaja a la par con “Teradata”.

Microservicio de tiempo/plazos: Es un sistema especializado que brinda conteo de tiempos para muchas acciones que realizan los numerosos sistemas del banco. Activa y/o genera eventos cuando dichos plazos se cumplen. La finalidad es éste, es la de contabilizar y administrar diversos tiempos a petición de otros sistemas. Una vez que se vencen los plazos solicitados, envía de regreso un registro de este evento de vuelta al sistema solicitante para que cumpla su objetivo.

Sistema de cajero automático: este elemento representa uno de los enlaces existentes que tiene el cliente con los sistemas bancarios. En este caso, genera una conexión entre el portador de una tarjeta bancaria válida y los sistemas bancarios autorizados para el nivel usuario, mismos que están delimitados con los servicios que el cliente tiene contratados y la información que la base de datos posee.

Estos son los enlaces existentes entre los sistemas bancarios, y podrían cambiar para cumplir los efectos que este proyecto establece para la creación de este software.

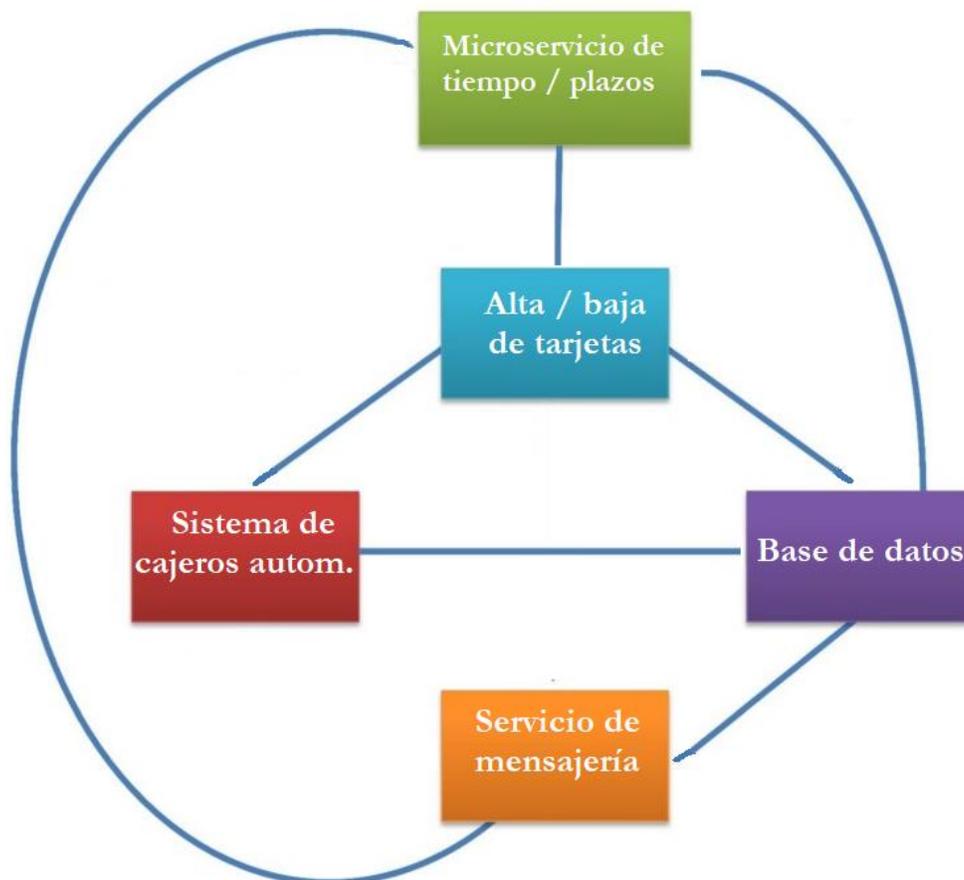


Figura 9. Un esquema de los enlaces entre aplicativos requeridos para este proyecto. Estas comunicaciones ya estaban establecidas con anterioridad para otras finalidades.

8.1. Desarrollo de metodología y equipo de trabajo

Las metodologías de desarrollo de software son marcos de trabajo diseñados con la finalidad de facilitar la estructura, planeación, etapas y orden de evolución en todos los procesos que involucran el acto de crear un software con altos estándares de calidad, desde la concepción de este, hasta su existencia formal y puesta en marcha dentro del entorno para el que fue pensado.

Derivado de lo anterior, es necesario identificar que las metodologías de desarrollo tendrán adaptaciones y modificaciones funcionales con base en las múltiples condiciones que permean en la empresa o con los responsables de la creación del software. Es por ello, que es

muy importante aclarar que las fases de desarrollo que se establecen en “SDLC” podrán estar sujetas a diversos cambios y/o ajustes con el fin de presentar de la mejor manera con el mejor software posible. Las actividades esperadas a lo largo de este ciclo de vida se describen a continuación:

- **Recabar antecedentes y documentación para la definición de características.** Se considera que, en esta fase, ya hubo una designación previa del equipo de trabajo que estará encargado de dirigir el proyecto, y estas personas no son las mismas que estarán encargadas específicamente de desarrollar el software. En esta etapa ya se realizaron los nombramientos de las personas que desempeñarán los roles de “Project manager”, el “Business Analyst”, arquitecto, el “tech lead” e integrador. Mismos que participarán continuamente en las actividades previas a las fases de desarrollo formal del software, como son la definición, diseño de algoritmo, aprobación de metodología, entrevistas con los “stakeholders”, definición de requerimientos funcionales, entre otras necesidades. Dichas actividades serán tomadas en cuenta para establecer y delimitar las diversas necesidades y habilidades que se requieren en el equipo de desarrollo.
- **Selección de desarrolladores.** El equipo de dirección del proyecto deberá seleccionar y comprobar las necesidades del proyecto, porque, con base en esto, se estará identificando a cada uno de los desarrolladores que trabajarán el desarrollo completo del software.
- **Desarrollo formal del software.** Una vez que los desarrolladores hayan sido seleccionados, se establecerá un breve tiempo para que se lleven a cabo juntas de consolidación de la estructura del software y su comprensión en general, para que los desarrolladores comprendan detalladamente cada necesidad que deberán cubrir. Posterior a ello, se procede a realizar el desarrollo del código que cumplirá con el software, recordando que habrá juntas de seguimiento para verificar los avances y constatar que el avance de todos los desarrolladores es parejo y correcto, según todos los plazos establecidos.
- **Período de pruebas y estabilidad.** Una vez que se culminó el desarrollo de todo el software, es necesario que se realicen las pruebas pertinentes. Entornos de testeo de software como “SIT” y “UAT” son las mejores formas de identificar tanto la calidad del software, así como las posibles fallas.

- **Generación de interfaz.** Las actividades en esta etapa tienen como finalidad convertir los procesos creados anteriormente para cumplir cada requerimiento funcional en interfaces compatibles con otros sistemas y entornos, así como verificar la integración correcta con ambientes o plataformas adyacentes con los que se tenga contacto durante su instalación o subsecuente operación.
- **Visto bueno y liberación (release).** Se llega a esta etapa una vez que se ha cumplido satisfactoriamente con todos los demás procesos y se considera que no es necesario cumplir más posibles iteraciones. En este punto, tanto el equipo de trabajo como los “stakeholders” ya están de acuerdo en el software obtenido, después de los períodos de pruebas, y ha quedado autorizado el lanzamiento pleno dentro del entorno existente. En este momento, cualquier proceso de creación y/o desarrollo de software ha sido culminado y se procede a realizar la inmersión de los componentes nuevos en el entorno en donde estará operando. Es probable que haya detección de “issues” que deberán ser corregidos en etapas de mantenimientos posteriores. Es altamente recomendable que los involucrados en la implementación realicen respaldos de seguridad previo a la instalación de este software.
- **Documentación y definición de entregables.** Ocasionalmente esta etapa puede estar considerada dentro de la fase anterior, pero requiere de un tiempo separado para dedicarlo a ella, debido a que es indispensable como buena práctica la depuración de los documentos y concretización de elementos entregables, adicionales al puro software como tal. Esto es debido a que se requieren registros y bitácoras de la creación del software, así como su documentación operativa para eventos futuros. Para los futuros eventos de mantenimiento, actualizaciones, corrección de errores y/o reinstalación en nuevos ambientes es indispensable realizar copias de seguridad, respaldos y evaluar los registros que se obtuvieron previamente y posteriormente a la instalación, para incluirlos en toda la documentación.

Es muy importante mencionar que estas 7 fases son diferentes a las 6 etapas que se contemplan en “SDLC” debido a que estas fases son de actividades por cumplir, y no están describiendo el ciclo de vida del desarrollo de un software, mientras que las etapas de “SDLC” son procesos definidos que se deben cumplir directamente dedicados a las tareas de desarrollo del software. En cualquier caso, las necesidades de la institución bancaria afectan directamente cada procedimiento descrito, con la finalidad de ajustarse a sus propias prácticas, políticas, normas, estatutos y/o cualquier otra condición que lo requiera.

En el entendido de que el equipo de trabajo se define por la conjunción de los roles que demanda la ejecución correcta de la metodología, así como los recursos humanos con los que cuenta el personal bancario disponible para atender las distintas fases, a continuación, se describe dicho equipo, añadiendo la participación de un integrador de sistemas como rol de soporte y validación.

Equipo de trabajo:

“Stakeholders”. Son los interesados en la creación de este software. Para este caso en particular, son los integrantes del comité que fueron entrevistados para la definición de los requerimientos funcionales. Si bien, no se consideran plenamente parte del equipo de trabajo porque no tendrán ninguna actividad práctica-funcional, si es altamente recomendable tener una línea directa de comunicación con ellos para depurar cualquier duda y/o interpretación de detalles que puedan surgir durante los procesos de desarrollo. Por lo general, el responsable de contactar con este grupo es el “Business Analyst”.

1 “Project Manager”. Es el responsable de todo el proyecto. Es la persona que facilita herramientas, comunicación, programa reuniones de avance, gestiona recursos y analiza riesgos, entre otras tareas. Tiene la capacidad y responsabilidad de identificar las correctas habilidades que se requieren en los desarrolladores.

1 “Business Analyst”. Es el encargado de comunicarse con los dueños el proyecto porque comprende y conoce las necesidades que solicitan. Es el canal más importante de conversión de necesidades de alto nivel hacia definición de requerimientos y casos de uso. También es el enlace entre los usuarios y el equipo que trabaja en el proyecto.

2 ingenieros de pruebas (“Tester”): Son los responsables de realizar todas las pruebas del software que los desarrolladores generaron. Ellos ejecutan el software generado y validan las operaciones realizadas por el software con todo el demás sistema bancario. Identifican y reportan errores de integración y ejecución.

1 arquitecto de sistemas: Es la persona que desarrolla el algoritmo general del proyecto. Posee información de diferentes niveles para el acceso a componentes necesarios durante el desarrollo del sistema. Detecta los sistemas que estarán involucrados en el proyecto, analiza el costo-beneficio de todas las actividades y supervisa el impacto de la participación del nuevo software en cada sistema involucrado.

1 líder de tecnologías (“Tech Lead”) analiza y abstrae los conceptos y funcionalidades que contiene la propuesta de solución al proyecto con respecto al algoritmo de operación para definir las tecnologías necesarias para la puesta en marcha del proyecto. De ser necesario propone cambios en el sistema en conjunto con los desarrolladores.

1 integrador de sistemas: Esta persona se encargará de dar soporte, verificar el cumplimiento de metodologías, avalar y/o cuestionar las buenas prácticas que se lleven a cabo, así como validar la unificación de piezas y componentes con respecto a los estándares del banco. También verifica la calidad de la documentación y entregables.

5 desarrolladores (uno por cada sistema involucrado): Son los encargados de realizar las modificaciones necesarias a los aplicativos que participan en el nuevo software para este caso en particular. De igual manera, escriben el código requerido para el programa que se está diseñando. Ellos pueden modificar líneas de código existentes para adaptar aquello que existe con respecto a los nuevos cambios, o en su defecto, crear código nuevo según sea el caso.

- Se requiere un desarrollador para el microservicio de tiempo/plazos.
- Se requiere un desarrollador para alta/baja de tarjetas.
- Se requiere un desarrollador para el sistema de cajeros automáticos.
- Se requiere un desarrollador para la base de datos.
- Se requiere un desarrollador para la conectividad con el servicio de mensajería.

8.2. Tecnologías propuestas

Dado que al finalizar la creación de este software se espera que el lanzamiento e instalación de este sean poco inmersivos, se descartará la utilización de paqueterías de software nuevo, de alto costo o que requiera la contratación de licencias con costo extra, para minimizar el presupuesto asignado

Considerando lo anterior, y además el hecho de que este sistema debe operar de manera nativa dentro de un entorno ya existente, como lo es el sistema bancario vigente, se utilizarán todas aquellas las tecnologías propias de cada aplicativo involucrado, así como también se recibió el visto bueno del arquitecto para utilizar “Java” como lenguaje de programación predominante en el diseño, así como C++ como lenguaje para algunos módulos en específico, tomando en cuenta los motivos mencionados. De igual manera, se sabe que los programadores responsables involucrados de cada aplicativo en el que habrá impacto de este software, ya trabajan bajo los lenguajes con los que está constituido cada uno de ellos, y estarán abriendo los canales de comunicación para establecer el enlace correspondiente que necesitará este software.

Base de datos: El software manejador es “TERADATA”, basado en lenguaje SQL.

Alta/baja de tarjetas de débito. Es un sistema clasificado como “backend”. Basado en lenguajes C++ y “php”.

Sistema de cajeros automáticos. La relación con el cliente para este y cualquier otro servicio de software hacia el portador de un “plástico”, es nivel “frontend”. La relación con este software y la activación del algoritmo son a nivel “backend”. Basado en “PHP”, C++ y “Javascript”.

Microservicio de tiempo / plazos. También es un sistema clasificado como “backend”. Basado en múltiples desarrollos “JAVA”.

Servicio de envío de mensajes “SMS”: Opera recibiendo paquetes en formato “XML”. Sus actividades posteriores a ello no impactan al banco ni al software dado que es un organismo externo al banco.

De igual manera, se definió que las siguientes tecnologías, “frameworks”, lenguajes, sistemas y/o paquetería son requeridos, y se menciona una breve explicación de su participación en este proyecto:

CORBA: Tecnología requerida para procesar desarrollos multilenguajes.

“J2EE: Java™ 2 Platform, Enterprise Edition”. Será requerido para capas “backend.”

“JDBC: Java™ 2 Database Conectivity”. Será requerido para capas “backend”, en particular, la base de datos.

“J2SE: Java Platform, Standard Edition”: Será requerido para capas “backend”.

“LINUX”: El servidor donde se alojará el software opera sobre en distribución de Linux.

“WINDOWS XP”: El software operativo de los cajeros automáticos funcionan sobre “Microsoft Windows XP”.

“ADOBE FLEX”: Será utilizado para algunas particularidades de Web Services.

“LIGHSPARK”: Software para interpretar archivos “SWF”.

8.3. Susceptibilidad a cambios

Los cambios en los requerimientos funcionales están condicionados a la operatividad de los procesos de cada fase de la metodología de desarrollo que se proponga. Dado que los tiempos de entrega ya están determinados y la participación de los involucrados representaría atención adicional a este desarrollo, en todo momento que se presente un cambio se volverá a cotizar el presupuesto asignado, además de la factibilidad de lo que se solicite, mismo que deberá ser presentado por escrito y con el visto bueno del “Project owner” y del “Bussiness Analyst” asignados a este desarrollo.

En caso de que dichas evaluaciones de factibilidad no sean favorables, se declararán estas peticiones de cambios en anexos que se incluirán en la documentación final, con el objetivo de que sean aplicados al momento de realizar las tareas de mantenimiento y/o actualizaciones futuras.

8.4. Algoritmo del programa

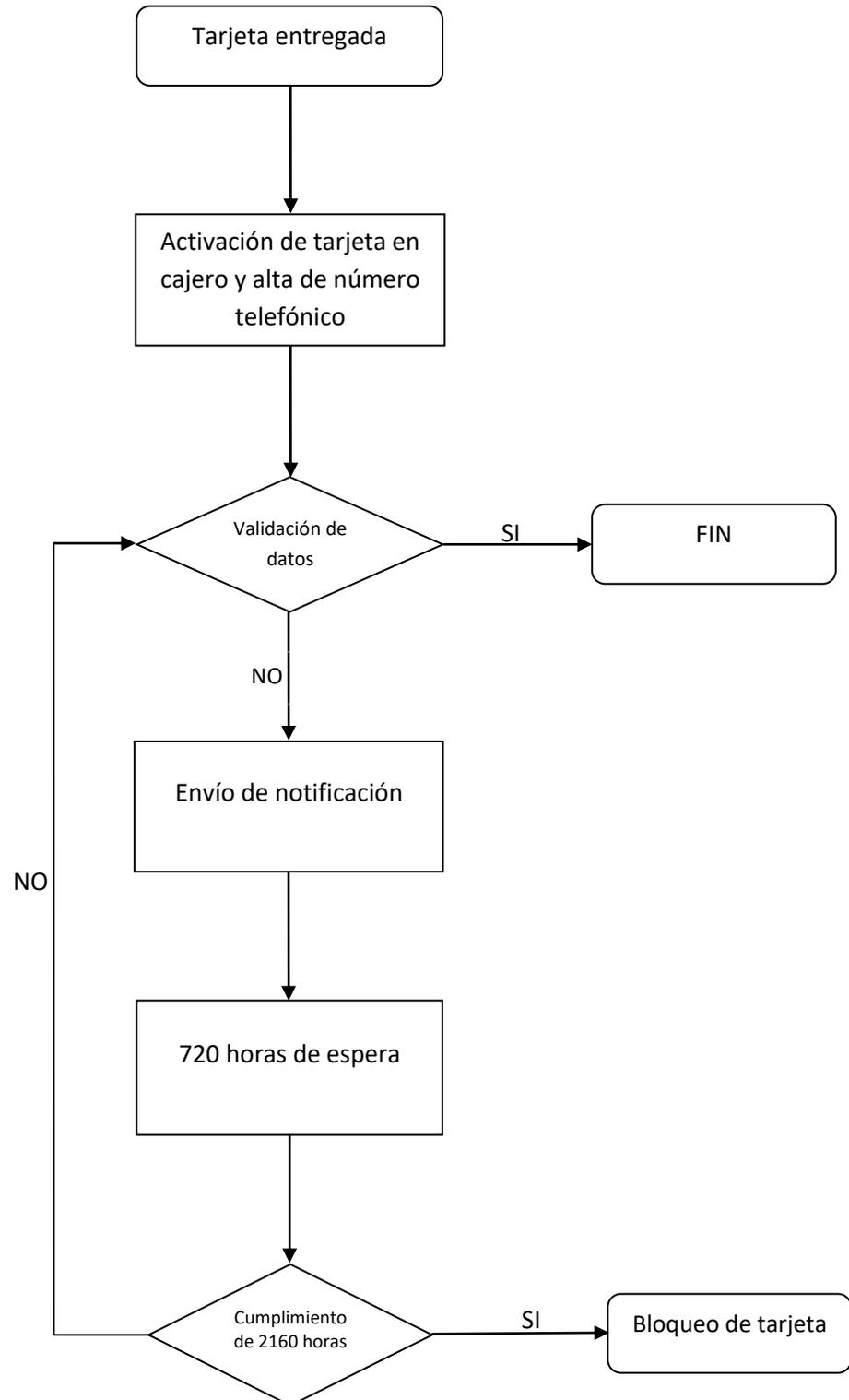


Figura 11. Este algoritmo representa la secuencia concreta de acciones que debe cumplir el software con base en cada evento presentado.

El algoritmo se creó a partir de la identificación del método operativo del software, incluye los requerimientos funcionales del sistema y el orden en que deben fluir. Debe estar incluido y en la documentación formal. Cabe señalar que, se descartan eventos de altas de remesas y tarjetas como sucesos previos a la entrega de la remesa a la institución responsable, así como actividades posteriores a una posible cancelación por incumplimiento de actualización de datos personales. Esto debido a que dichos eventos y acciones de respuesta ya están activos en otros sistemas.

9. Solución propuesta

Se establece que el nombre clave de desarrollo de este software será **DecoTurf**. Mismo que podrá ser cambiado por algún otro que el departamento de TI de la entidad financiera determine.

Se tomará como referencia el análisis y explicación detallada del algoritmo para establecer los siguientes eventos en concreto:

Tarjeta entregada: Si bien, dentro de los procesos operativos del banco, el evento “tarjeta entregada” comprende numerosas actividades previas a que el portador del “plástico” lo presente en un cajero automático para su activación, aquí se deberá identificar con claridad que las tarjetas participantes en este software serán solamente las tarjetas de débito de remesas masivas. Esto se podrá conocer en el momento en que las cuentas bancarias sean creadas y dichos “plásticos” sean catalogados y vinculados a esas nuevas.

Dicho lo anterior, el disparador de esta fase deberá ser el sistema de alta de una remesa masiva, porque al momento que una remesa masiva sea entregada a una institución externa al banco, todas las tarjetas de débito que contiene ya deben estar activas y correctamente vinculadas a una cuenta bancaria.

Activación de tarjeta en cajero y alta de número telefónico: Corresponde, necesariamente a la primera actividad que el portador de una tarjeta de débito realiza (sea o no sea de remesa masiva). Es necesario cambiar el NIP predeterminado en un cajero automático para acceder a las funciones habituales que la cuenta de bancaria ofrece con su “plástico” correspondiente.

Es en este paso el portador del “plástico” será notificado de las condiciones de uso de la cuenta bancaria y su correspondiente tarjeta de débito; es por ello, que se le indicará que debe finalizar el alta de su expediente actualizando sus datos personales en la institución bancaria emisora de la cuenta, presentándose con una identificación oficial en un plazo máximo de 90 días naturales a partir de esta fecha. Finalmente, para hacerle llegar recordatorios de esta normatividad, se le solicitarán los diez dígitos de un número telefónico

móvil válido en la República Mexicana para enviarle mensajes de texto “SMS” reiterando esta información. Y que, de no hacerlo, la tarjeta de débito será desactivada permanentemente.

Validación de datos: Esta fase corresponde al evento en el que el portador de la tarjeta acude a una sucursal bancaria con identificación oficial para cumplir con el objetivo de verificar/validar/actualizar los datos personales ante la institución bancaria. El cliente cumplirá con los requisitos vigentes al momento de su presencia en el banco, y dicha información será almacenada en la base de datos del banco y administrada de manera normal como se realizaría con un cliente que no corresponde a tarjeta de remesa masiva.

Dado que este evento es el objetivo central de la creación de este software, su cumplimiento será el parámetro que determine si el software finaliza su operación o continúa para realizar las actividades de envío de notificaciones mediante “SMS”. Es decir, si el cliente valida correctamente sus datos personales en sucursal, el software no enviará notificaciones. De lo contrario, el software realizará las iteraciones pertinentes de enviar notificaciones y validar los datos personales, hasta que el cliente cumpla con la actualización de sus datos o se termine el plazo estipulado y se proceda a la cancelación del “plástico” involucrado.

Envío de notificación: Esta actividad será realizada cuando el software valide que la tarjeta de débito correspondiente no cuenta con un expediente correctamente estructurado en la base de datos. Se enviará un máximo de 4 mensajes de texto SMS, bajo las siguientes condiciones:

- El primer mensaje será enviado inmediatamente que el cliente active su tarjeta de débito, cambie su NIP predeterminado por uno personalizado y en esa misma sesión de cajero automático asigne un número telefónico móvil válido en la República Mexicana. En este primer mensaje no se valida el “status” de expediente bancario en la base de datos.
- El segundo mensaje dependerá de dos condicionantes, la primera es que se hayan cumplido 720 horas posteriores al envío del primer mensaje de texto, y segunda, que el portador del “plástico” no haya validado sus datos todavía. En caso de que la segunda condicionante esté cubierta, el software termina su función por completo.
- El tercer mensaje será una iteración del segundo mensaje, solamente cambiando las 720 horas de espera con respecto al primer mensaje, por 1440 horas (que representan 60 días).
- El envío del cuarto mensaje debe cumplir con las condicionantes que poseen el segundo y el tercer mensaje, cambiando las horas de espera por 2160 (que representan 90 días de plazo máximo) La diferencia con respecto a los mensajes previos, es que no estará invitando al cliente a actualizar sus datos, sino que estará informando del bloqueo definitivo de la tarjeta de débito por la omisión de la actualización de datos personales

720 horas de espera: Corresponde al intervalo que se establece entre una notificación y otra. Son 30 días naturales. Es el tiempo de espera para volver a validar la base de datos para verificar si el cliente ha cubierto su actualización de datos para dar por finalizadas las tareas del software, realizar otra iteración o proceder a la cancelación de la tarjeta involucrada, según sea el caso.

Cumplimiento de 2160 horas: Es un contador de tiempo que genera una condición que validará si el plazo máximo para actualizar datos personales. Define las siguientes acciones:

- Si la base de datos fue actualizada correctamente, procede a la finalización del software
- Si la base de datos no fue actualizada correctamente, envía una notificación informando del bloqueo definitivo de la tarjeta de débito, junto con la instrucción de acudir a sucursal bancaria para actualización de datos bancarios y reasignación de un “plástico” nuevo. Procede a la finalización del software

Bloqueo de tarjeta: Esta acción se presenta cuando se cumplieron todas las fases del algoritmo, y no se validó la actualización de los datos personales del cliente, en conjunto con el vencimiento del plazo máximo de 2160 horas

9.1. Requerimientos funcionales

- “C1” – El software “DecoTurf” deberá ser capaz de identificar aquellas tarjetas de débito que pertenecen a remesas masivas y excluir las tarjetas con otra clasificación.
- “C2” – Almacenar el número telefónico proporcionado por el portador del plástico.
- “C3” – Validaciones periódicas de los datos que se solicitan actualizar.
- “C4” – “TELUS” debe enviar los mensajes SMS a petición de DecoTurf.
- “C5” – Es necesario emitir una instrucción de bloqueo de tarjeta al vencer el plazo estipulado.

Características principales del sistema

El sistema se encargará de gestionar las notificaciones mediante mensajes SMS que debe recibir un cliente en su teléfono móvil para recordarle la necesidad de acudir a sucursal bancaria para actualizar sus datos personales por ser portador de una tarjeta de débito de remesa masiva.

El sistema deberá coordinar acciones con otros aplicativos del banco ya existentes, con el objetivo de recabar el número telefónico para el envío de las notificaciones, validar periódicamente que si el cliente ya actualizó u información o aún no lo hace, establecer

plazos para el envío de éstos recordatorios, suspender el envío de dichos mensajes cuando el cliente cumpla con la actualización de sus datos, y finalmente, suspender el uso de la tarjeta de débito involucrada a través del bloqueo del plástico cuando el cliente no actualizó sus datos y el período se haya vencido.

9.2. Contexto del sistema

Actualmente, el banco no cuenta con una aplicación, programa, software ni solución computacional que se encargue directamente de cumplir plenamente las funciones que persigue “DecoTurf”. Por lo que la creación de “DecoTurf” no será considerada una actualización, mejora, sustitución, cambio de versión ni ningún tipo de cambio renovación ni relevo de un software por otro.

De igual manera, se debe contemplar que el entorno operativo actual del banco ya cuenta con diversas herramientas y aplicativos que serán modificados para que brinden los múltiples servicios que se requieren en “DecoTurf”.

Por último, si bien, existe un servicio de notificaciones “SMS” a clientes operando actualmente en el banco, este servicio no está vinculado a ningún software que tenga la obligación de cubrir los requerimientos que “DecoTurf” busca.

Declarado lo anterior, se establece que “DecoTurf” es un software cuya creación es indispensable, necesaria y cuyos requerimientos funcionales no están cubiertos por ningún otro software actual del banco. Su diseño se creará a partir de cero y con parámetros propios.

Descripción de los actores

Actor #1: Componentes de sistema

Dado que este software tiene como objetivos principales el cumplimiento de diversas acciones que dependen exclusivamente de validaciones de diversos parámetros de sistema, es el propio sistema quién será el disparador de diversas funciones que debe realizar. Las acciones que debe realizar el sistema derivado de instrucciones del propio sistema son las siguientes:

- Almacenar los datos de las tarjetas de débito de remesas masivas. Acción que será cumplida cuando el sistema de alta de tarjetas de débito haya definido ciertos lotes de estos “plásticos” como remesas masivas.
- Validación de el correcto status de los datos de identificación de un cliente. Esta acción está sujeta a la verificación de activación de tarjeta, disponibilidad del número

telefónico válido para envío, y confirmación del cumplimiento de los plazos preestablecidos

- Generación de la orden de envío de notificación. Recordando que este software no es quien enviará las notificaciones SMS, sino que crea la instrucción de que el servicio encargado realice dicha acción. La orden de envío es generada a partir de la condición de que no haya un expediente correctamente actualizado en la base de datos en conjunto con el vencimiento de cualquiera de los plazos establecidos.
- Generación de orden de bloqueo de tarjeta. Este software no tiene facultad de bloqueo de tarjetas, esa actividad corresponde al aplicativo directamente encargado de ellos. Aquí sólo se genera la orden de bloqueo de tarjeta de débito luego de cumplir con los requisitos de tiempo de actualización vencido y la inexistencia de un expediente de cliente actualizado.
- Finalización de acciones. El software terminará acciones con respecto al cliente portador de un plástico de remesa masiva cuando se cumpla una de las siguientes acciones:
 - Se valida que la actualización de datos ha sido completada
 - Se generó la orden de bloqueo de tarjeta

Actor #2: El Cliente

En las actividades correspondientes al software, el cliente portador de la tarjeta únicamente cumple la acción de proporcionar el número telefónico móvil válido en la República Mexicana en el momento en que activa la tarjeta de débito en un cajero automático.

9.3. Requerimientos no funcionales

Debe generar el archivo de notificación en “formato xml” para que el servicio de envío de mensajes sea capaz de interpretarlo correctamente y generar la notificación “SMS” correctamente

Verificar que cada notificación “SMS” sea recibida por el cliente dentro de las primeras 12 horas posteriores a la generación del archivo xml.

“DecoTurf” deberá ser capaz de identificar las tarjetas de remesas masivas existentes y vigentes hasta el momento de su puesta en marcha, así como las nuevas tarjetas de remesa masiva que sean activadas posteriormente.

El software “DecoTurf” debe ser capaz de operar con una cantidad indefinida de tarjetas de remesas masivas, así como atender el permanente crecimiento de este volumen. La recomendación es que sea capaz de atender los diversos procesos para un mínimo de 20 millones de tarjetas de débito pertenecientes a remesas masivas.

El software debe tener una integración plena con cada aplicativo del banco, con el objetivo de minimizar el impacto operacional que genere una vez que en su “release” sea instalado formalmente.

“DecoTurf” deberá tener una integración plena con las terminales bancarias de los cajeros automáticos que están en la red de propiedad directa del banco involucrado. Sin embargo, es altamente deseable que “Decoturf” también tenga presencia y participación completa en cajeros automáticos de cualquier otra red aceptada para operaciones con el banco involucrado.

Es deseable que los registros de las tarjetas que terminaron toda actividad relacionada con los procesos del software “DecoTurf” sean depurados. Sin embargo, el status de las acciones y los registros de los eventos ejecutados por este software serán conservados en archivos de control de actividades para consultas futuras.

9.4. Riesgos

- Se sabe que la compartición de información sensible y confidencial a terceros representa el riesgo de que dicha información deje de ser controlada completamente por el banco y pueda ser difundida de formas no autorizadas por los lineamientos del banco. Esto puede presentarse cuando el software genere reportes de envío de notificaciones hacia la empresa encargada de brindar el servicio de mensajes “SMS”. Esto debe ser perfectamente bien contemplado en la Ley Federal de Protección de datos personales en posesión de los particulares, así como en el contrato de prestación de servicios que se establezca con dicha empresa.
- El riesgo operativo que se puede presentar es que el software que se propone se vislumbra hacia el momento del “release” en el proceso de instalación en el entorno de sistemas funcionales actuales del banco, ya que la adhesión de este software en este entorno podría generar cualquier tipo de problemas en la operatividad normal con la que se cuenta actualmente (inestabilidad, congelamientos, problemas de integridad, conflicto de versiones, etcétera).

9.5. Involucrados

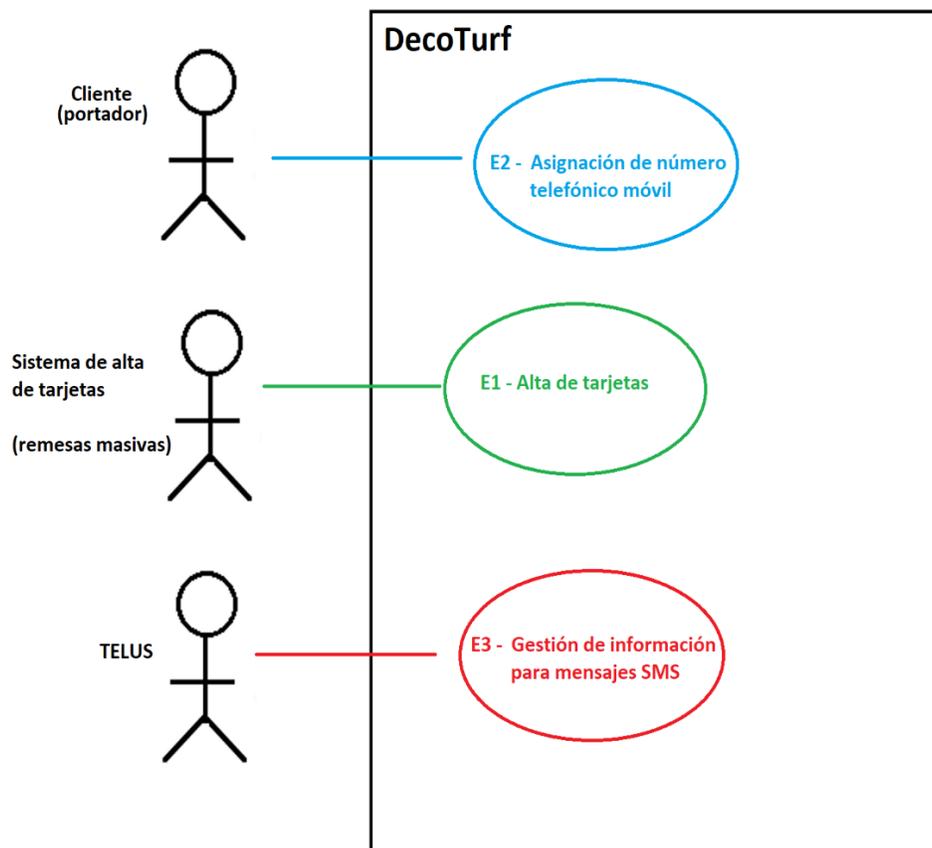
Actor (rol)	Involucrado primario	Involucrado secundario
Crear remesas masivas	Sistema de alta de remesas masivas	Gestoría de tarjetas de débito
Cliente	Varios	
Envío de mensajes SMS	TELUS	

9.6. Casos de uso

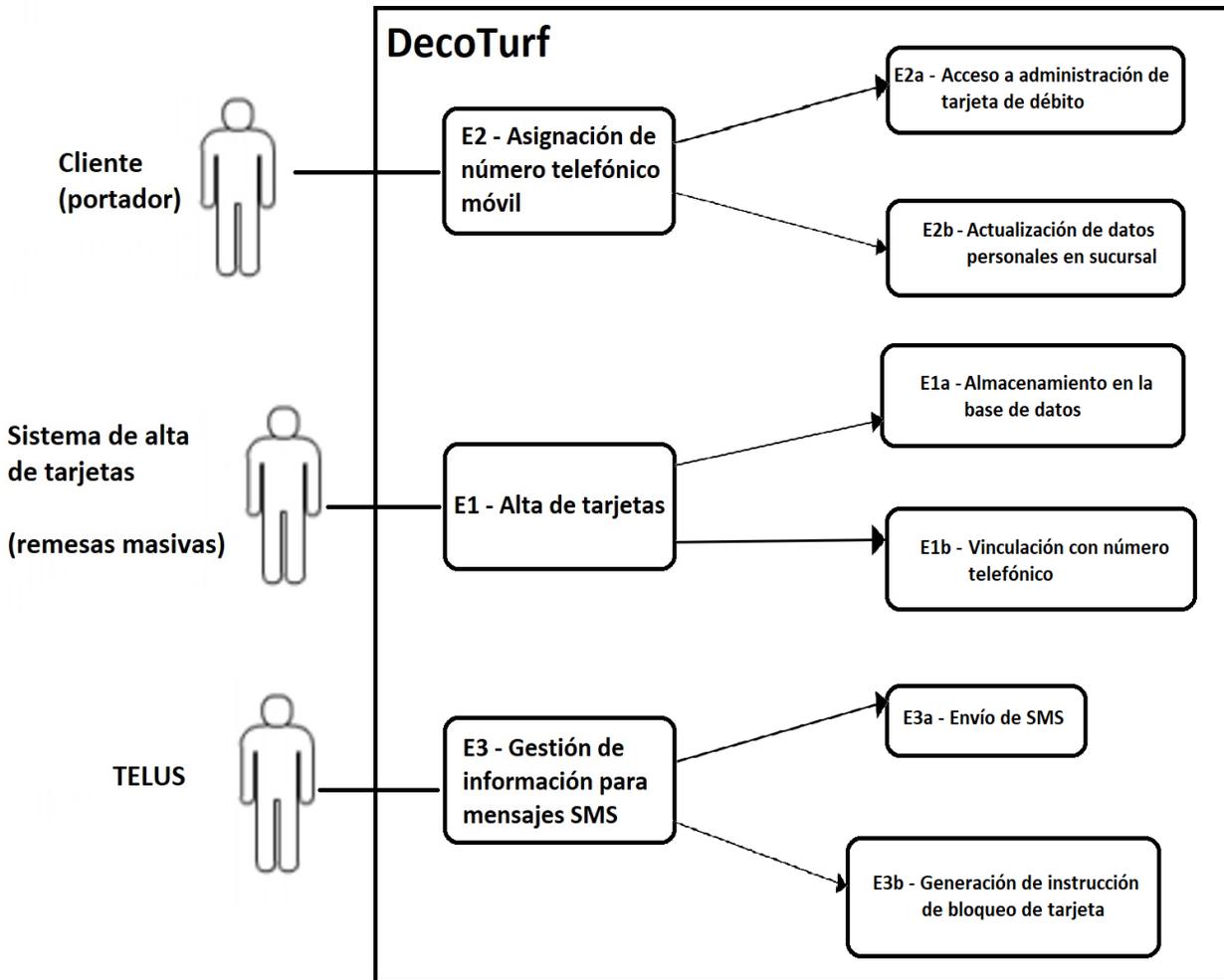
Caso de uso	Prioridad	Núm.	Descripción
Alta / Identificación de tarjetas válidas para "DecoTurf"	E	1	Solamente las tarjetas pertenecientes a remesas masivas activan "DecoTurf".
Un cliente proporciona un número telefónico móvil.	E	2	Un cliente recibe información acerca de los requerimientos legales y por ello asigna un número telefónico móvil para vincularlo a su tarjeta de débito.
"TELUS" recibe / gestiona peticiones desde "DecoTurf" para enviar mensajes "SMS"	E	3	Es necesario que "TELUS" mantenga conexión permanente con "DecoTurf" para recibir pedidos de envíos de mensajes "SMS".

*E=esencial, R=recomendado

9.7. Diagrama de casos de uso básico



9.8. Diagrama de casos de uso refinado

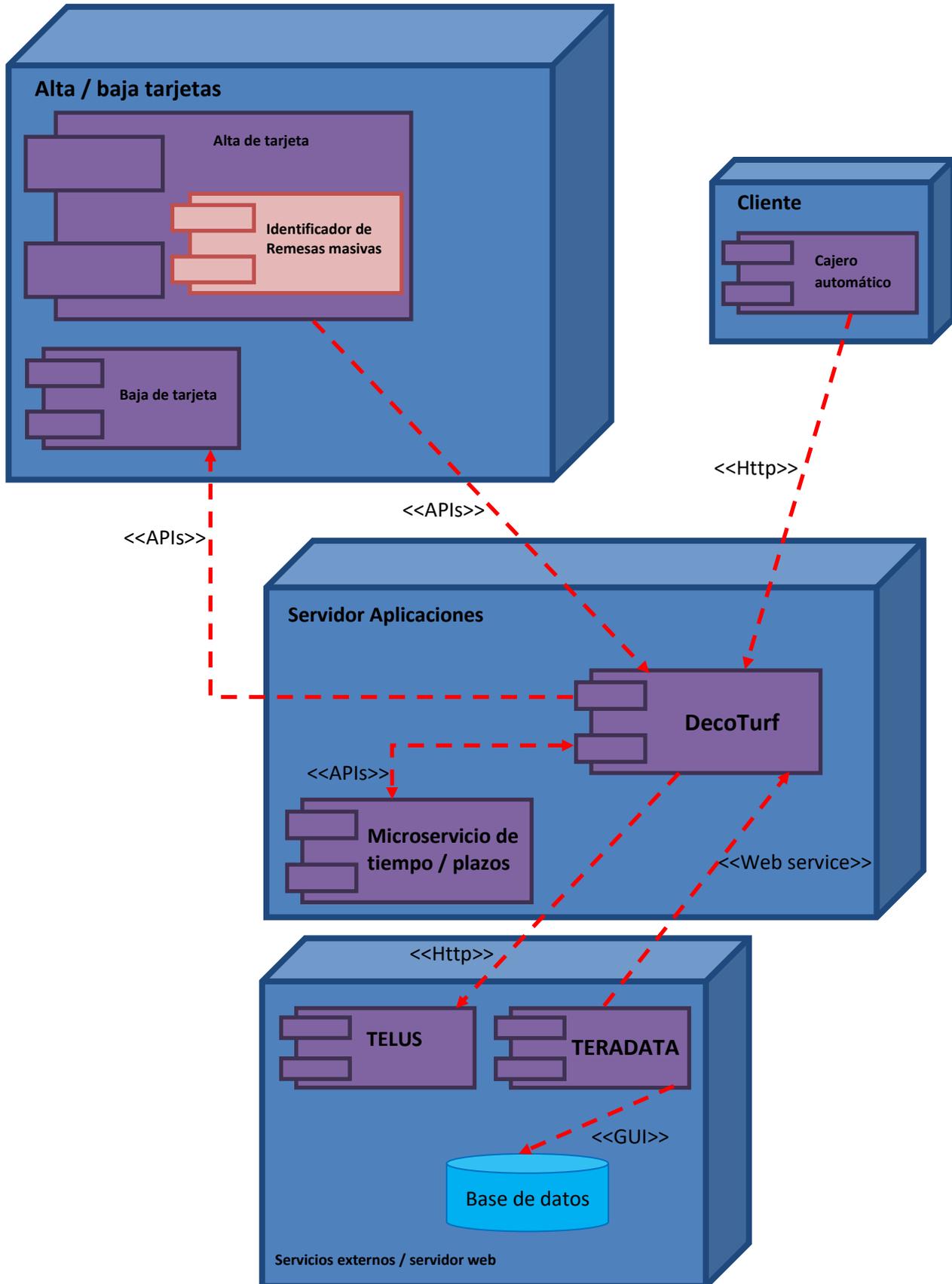


Relación con requerimientos funcionales

Requerimiento funcional	Descripción
C1	E2a
C2	E1a
C3	E3
C4	E3a
C5	E3b

9.9. Arquitectura

Posible arquitectura (arquitectura general) Diagrama de Deployment



9.10. Manejo y almacenamiento de información

A continuación, se describen algunos de los contenidos globales de información que serán manejados por “DecoTurf”:

Información requerida	Información obtenida	Información generada
<ul style="list-style-type: none">• Número de cuenta bancaria• Número de tarjeta de débito• Nombre del titular de la cuenta• Fecha de cambio de NIP	<ul style="list-style-type: none">• Número telefónico móvil de 10 dígitos• Fechas de envío de mensajes SMS	<ul style="list-style-type: none">• Instrucción de envío de mensaje “SMS”• Instrucción de bloqueo de tarjeta de débito

Los requerimientos para establecer un almacenamiento de información a través de la base de datos tienen las siguientes características:

- Conectividad con Teradata a través de los “Web Services” que ya operan actualmente. Será necesario abrir un puerto en el servidor para realizar este enlace.
- Las necesidades que requiera “DecoTurf” con la información existente en tablas de bases de datos actuales, será únicamente con fines de consultas. No se otorgará ningún privilegio para cambios.
- “DecoTurf” si tendrá privilegios para crear y administrar una tabla única en el manejador de base de datos, “Teradata”; misma que será diseñada y dedicada exclusivamente para los objetivos que persiga este software. Si bien, esta tabla será considerada como la base de datos de DecoTurf, no tendrá relación alguna con otra tabla que también se requiere para consultas.

Se anticipa que podrá haber dos tipos de clientes que pueden recibir una tarjeta de remesa masiva: uno es aquel que ya tiene antecedentes de cualquier índole como cliente del banco, mientras que otro posible cliente que nunca ha tenido contacto con el banco y que debe ayudar a crear el expediente completo. En cualquier caso, “DecoTurf” no tiene privilegios de alterar la base de datos del banco en pro de los objetivos buscados. El momento de dar de alta información del perfil formal del cliente será cuando acuda a sucursal bancaria y sea el ejecutivo de cuenta quién realice estos procedimientos.

Dada la naturaleza de operación que se plantea para “DecoTurf”, y las condiciones antes mencionadas, no se propone ni se requiere la creación de una base de datos de operación completa. Únicamente se requieren privilegios para la creación de una sola tabla; mientras que la base de datos del banco notificará (en coordinación con los plazos de notificación) si hay o no hay actualización de datos personales. Lo anterior, significa que se planificarán consultas periódicas para obtener información de una sola tabla de la base de datos de clientes del banco.

La propuesta de la tabla que se requiere crear está a la izquierda, mientras que se espera que la tabla existente de clientes tenga una estructura como la que se supone del lado derecho. **(Ver apéndice #2)**

tarjeta_decoturf		
numero_tarjeta	numeric(18,0)	PK
fecha_activación	date	
número_telefónica	numeric(18,0)	
primer_notificación	date	
segunda_notificación	date	
tercer_notificación	date	
cuarta_notificación	date	
fecha_bloqueo	date	

Esta es la única tabla de base de datos que administrará DecoTurf. No participa en la base de datos del banco ni tiene relación con ninguna otra tabla de ninguna otra base de datos

cliente		
num_cliente	numeric(18,0)	PK
nombre	varchar(50)	
ap_paterno	varchar(50)	
ap_materno	varchar(50)	
fecha_nacimiento	date	
curp	varchar(50)	
identificación	varchar(50)	
dirección	varchar(50)	
teléfono	numeric(18,0)	
correo_electrónico	varchar(50)	

Una tabla como esta es la que se busca actualizar. Pertenece a la base de datos del banco, y una vez que sea vinculada a la cuenta y tarjeta de remesa masiva, podrá interactuar normalmente como cuenta de débito simple.

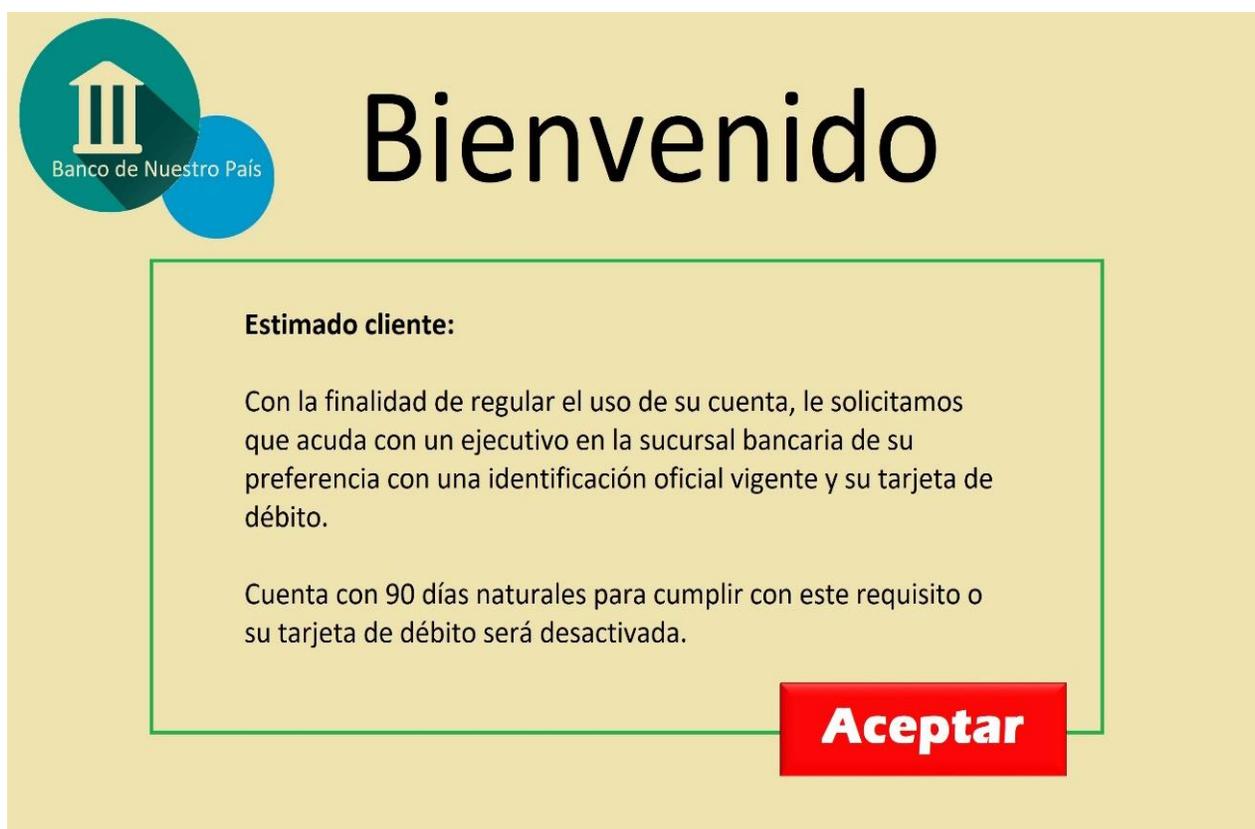
9.11. Diseño de interfaz gráfica

Se considera que un gran parte del diseño y operación de “DecoTurf” será automatizado y a manera de sistema de perfil “backend”. Es por ello que no se requiere una interfaz gráfica dedicada a la interoperatividad entre el usuario y el sistema, o alguno de los aplicativos que involucran a “Decoturf”. Esto se debe a que, las relaciones que tienen los usuarios humanos con aplicaciones que tengan alcances y/o relaciones con “DecoTurf” ya están establecidas y predeterminadas con anterioridad para otros sistemas, y naturalmente, ya cuentan con interfaces gráficas dedicadas que no estarían requiriendo una alteración preponderante para el correcto funcionamiento de este software. Incluso, se concretiza que las conectividades necesarias con “TELUS”, el sistema de alta y baja de tarjetas, la base de datos, y el microservicio de tiempos/plazos, se mantendrán con enlaces de la programación que permitan la funcionalidad. Ninguno de estos sistemas ni enlaces tendrá un diseño de interfaz gráfica dedicada.

Basado en lo anterior, la única participación humana en los procesos de “Decoturf”, se presentará de manera irruptora en el protocolo de activación de una tarjeta de débito en un cajero automático, con el objetivo de notificar las necesidades de actualización de datos al cliente, así como de asignación de un número telefónico para la ejecución de “DecoTurf” y la correcta activación de la tarjeta de débito de remesa masiva.

Dada esta circunstancia, se proponen las siguientes pantallas de información y solicitud de datos para el cliente portador del plástico al momento de utilizar por primera vez el cajero automático:

La primer pantalla perteneciente a DecoTurf se presenta inmediatamente después de que se cumple la operación obligada que establece el cajero automático como protocolo para la activación de una tarjeta de débito de primer uso, que es la asignación de un NIP personalizado. A partir de este punto, se confirma que la tarjeta de débito de remesa masiva ha quedado activada satisfactoriamente porque el NIP predeterminado ha sido modificado por el NIP personalizado. Al finalizar esto, aparece:



Descripción de botones:

- El botón aceptar conduce a la siguiente pantalla.

*Cabe aclarar que el botón físico de “CANCELAR” en el teclado del cajero automático puede cancelar el proceso, evitar ir a la siguiente pantalla, y expulsar la tarjeta, pero ello no suprime el alta del nuevo NIP personalizado.

Luego de dar a conocer dicha información, la pantalla siguiente procede a solicitarle al cliente el número telefónico móvil a 10 dígitos válido en la República Mexicana para enviar las notificaciones “SMS”. La propuesta de esta pantalla es la siguiente:

Descripción de botones:

- Los botones físicos del teclado numérico del cajero permiten introducir los dígitos del número telefónico solicitado.
- El botón **“aceptar”** almacena la información proporcionada por el usuario y conduce a la pantalla principal de operaciones
- El botón **“cancelar”** no almacena ningún número telefónico y finaliza la sesión del usuario en el cajero automático, entregando la tarjeta insertada.

*Es mandatorio ingresar un número telefónico móvil para enviar las notificaciones. De lo contrario, no se podrá acceder a la pantalla principal de operaciones en ese momento, ni en ninguna sesión de cajero automático.

9.12. Aplicaciones

Si bien, “DecoTurf” es un software diseñado como un elemento con presencia a nivel “backend” y por ello no se considera que tendrá alcances / accesos por diversos medios, es necesario identificar cada subsistema para relacionarlo con el respectivo caso de uso que deberá cumplir.

Subsistema	Descripción	Caso de uso
Alta / baja de tarjetas	Identificar tarjetas de remesas masivas / Bloqueo de tarjetas	E1, E5
Cajero automático	Almacenar número telefónico móvil	E2
“TELUS”	Envío de mensajes “SMS”	E4
Teradata	Almacenar número telefónico móvil / Eliminación de registros obsoletos / Verificación de datos personales	E2, E3, R1
Microservicio de tiempo / plazos	Bloqueo de tarjeta	E5

9.13. Cronograma de trabajo

Se contempla un tiempo aproximado de 6 meses y 3 semanas para cubrir todas las fases de desarrollo y realizar el release formal del software.



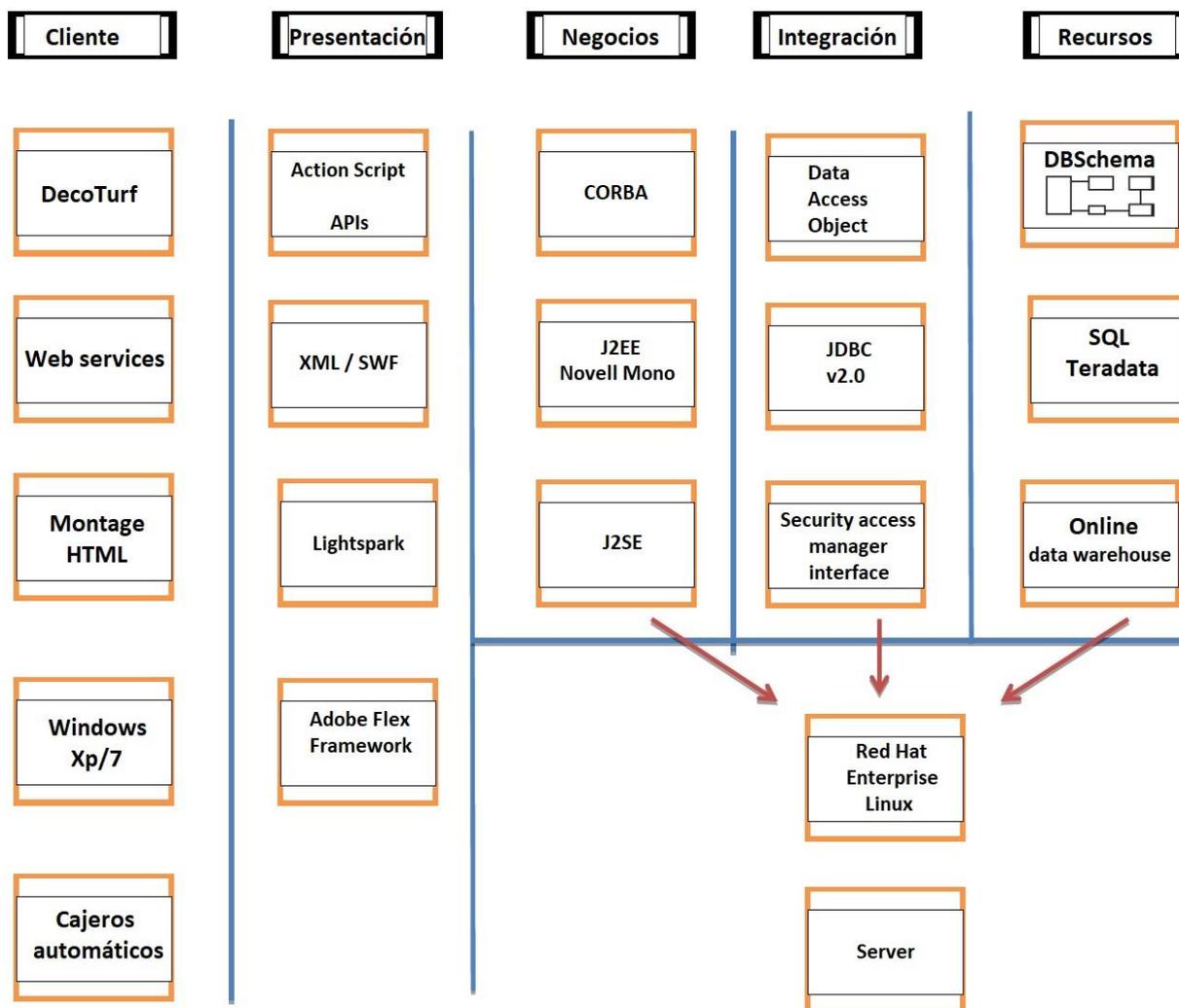
A este tiempo, se le deberá considerar la extensión por cuestiones de iteraciones, y un posible período para corrección de “issues” (si los hay) una vez lanzado, y finalmente, el período de cierre de documentación una vez que los “issues” hayan sido atendidos.

De igual forma, se considera que la documentación final, en conjunto con los entregables, incluirán todos los elementos necesarios para los procesos de calidad posteriores a la creación del software, entre ellos, bitácoras, registros, consideraciones, riesgos detectados y cualquier información útil para facilitar acciones futuras, como pueden ser mantenimiento,

las actualizaciones, corrección de errores, instalación en nuevos ambientes e incluso la posibilidad de cambiar tecnologías involucradas.

9.14. Arquitectura detallada

Propuesta de arquitectura detallada, con nomenclatura de modelado de 5 capas



9.15. Pruebas e iteraciones

La fase de pruebas de “Decoturf” permitirá conocer e identificar el avance tangible y real del funcionamiento del software, así como la detección del cumplimiento de las funciones y necesidades planteadas en fases previas. Asimismo, las iteraciones posteriores consecuentes a cada ciclo pruebas, arrojarán información referente a metas alcanzadas, y posibles problemas.

Inicialmente, para las primeras fases de pruebas, se sabe que el software debe estar operando en ambientes simples, controlados y bajo observaciones de reacción y comportamiento para asegurar que las ejecuciones de funciones, instrucciones y comandos pueden cumplir los diversos objetivos correctamente, además de ser, simultáneamente seguros en la duración de sus actividades. A estas pruebas se les conoce como pruebas unitarias, debido a que suelen ser procesos independientes en donde se examinan piezas y/o componentes de manera individual, evaluando que su comportamiento sea correcto y cumplan con las funciones para las que fueron diseñadas según su programación autónoma.

La fase de pruebas para el software global y su objetivo formal, está comprendida en dos procesos o etapas que servirán para separar las actividades técnicas de ejecución de aquellas actividades demostrativas que tienden a ser más concretas, superficiales y más avanzadas porque están orientadas a ser presentadas más a manera de evidencia.

Las evaluaciones comienzan ejecutando las pruebas de integración de sistemas de software (“Software integration testing” o solamente “SIT”), mismas que podrán exhibir que las piezas individuales exitosas de las pruebas unitarias ya operan de manera correcta, así como también hacer constar que se están cumpliendo requerimientos funcionales, casos de uso, diagramas de flujo, algoritmos y elementos a nivel estructural planeados para el software.

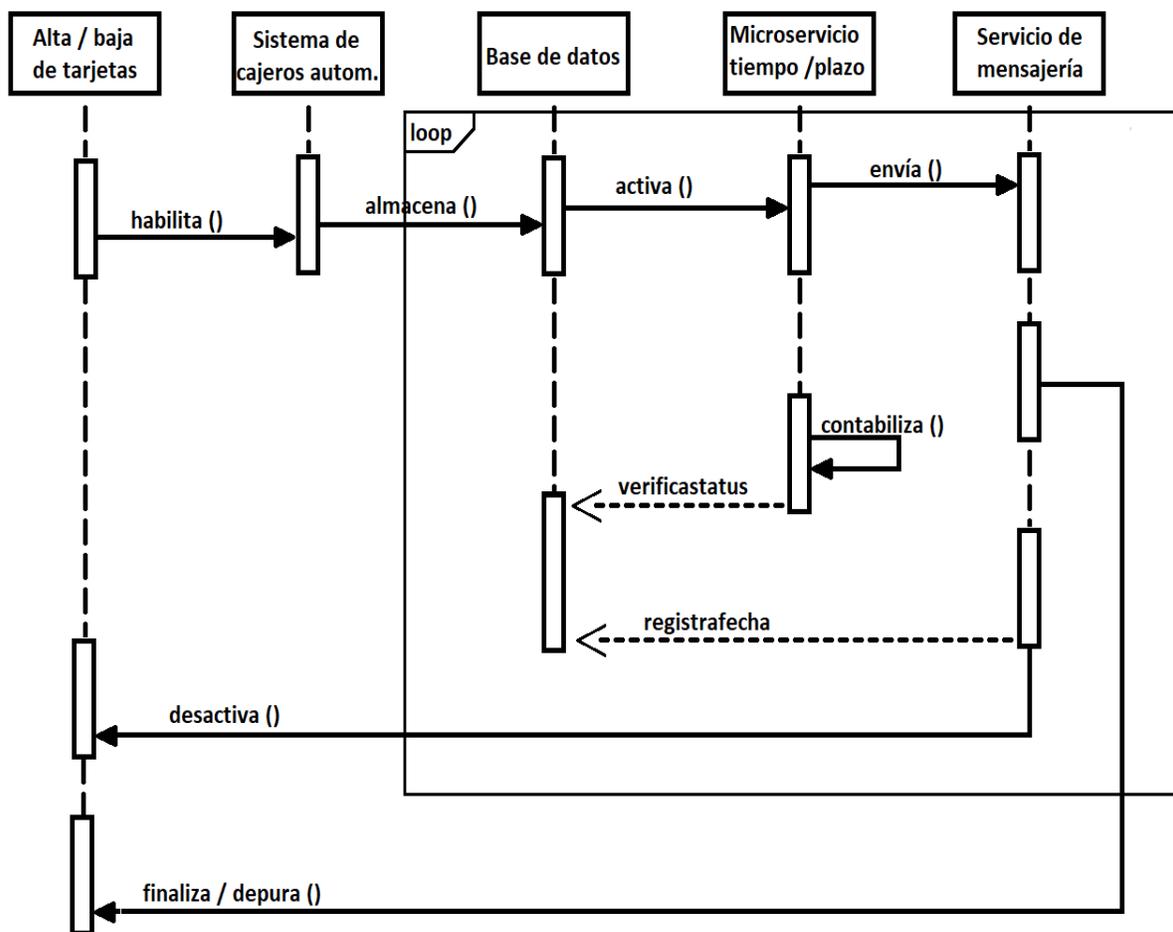
Las pruebas a nivel técnico, incluyen la necesidad de realizar exámenes específicos conforme el tipo de software que se está creando, es decir, si se considera que DecoTurf es un software que involucra relaciones con múltiples sistemas y plataformas, así como conectividad con servidores, bases de datos y administración de peticiones simultáneas, es imperativo realizar pruebas particulares de resistencia (endurance) y de estrés (stress) para demostrar y advertir el comportamiento del software en situaciones de alta demanda, así como también conocer los límites máximos que puede sostener prevaleciendo su estabilidad.

Una vez que dichos análisis y evaluaciones han sido superadas satisfactoriamente, es común que se preste atención para la generación de una interfaz gráfica, en proyectos en donde la participación de un usuario final es amplia y determinante, ya que la obtención de las capas de interacción con el usuario son las últimas actividades que se trabajan, dado que en este momento se ha aceptado la operación del software a nivel estructural, y se procederán con las pruebas de demostración para los “stakeholders”.

Cuando se tiene la presencia de los interesados, “stakeholders”, y en este caso, el comité que solicita la creación de “DecoTurf”, se realizan las pruebas de aceptación de usuario (“User Acceptance Testing” o solamente “UAT”). El objetivo con estas pruebas es demostrar que el software ha llegado a los últimos momentos de su fase de desarrollo, superó pruebas “SIT”, ya cuenta con interfaz gráfica y se considera que se ha completado en más de un 90%. A estas alturas, ya se tiene una estructura y diseño capaz de cumplir con todos los

requerimientos y en realidad sus cambios y perfeccionamientos en una posible nueva iteración, tendrían que ser mínimos y de nulo impacto.

En el siguiente diagrama de secuencias, se puede apreciar de forma clara cada paso ordenado de manera sucesiva conforme cada evento avanza:



Lo que expresa este diagrama corresponde a eventos ordenados según el avance de cada suceso, y la forma en cómo se estructuran con relación a cada sistema involucrado. Si bien, este es un diagrama que puede ser creado a manera de sustento para la creación del código base del proyecto, también se utiliza como un parámetro esquemático de los eventos que se van a estar presentando en las pruebas, y cómo debe impactar una acción previa/posterior a la siguiente, y en caso de errores, es más sencillo identificarlos en una fase de pruebas dedicada a ello.

En los períodos de pruebas, se pueden contemplar diversas acciones que involucren aspectos fuera de los lineamientos originales del software, como limitarse a realizar exámenes de

requerimientos funcionales y estabilidad. En el caso de institución bancaria, y dado el grado de valor que representan sus activos informáticos, hay rutinas orientadas a poner a prueba elementos de seguridad y de integridad. Esto significa que se realizan acciones que comprometen la seguridad básica y avanzada del software de reciente creación, así como la estabilidad de su operación, a través de intentos de vulneraciones en su integridad. Pruebas tan sencillas como el ingreso de datos incorrectos o inválidos en un campo que solicita información, o tan complejas como la contratación de un “hacker” que busque corromper el sistema, suelen ser algunas de las prácticas más comunes en una etapa de pruebas que integra estos requisitos como parte del proceso de pruebas generales par aun software nuevo de cualquier nivel por designación general de la institución bancaria.

La fase de pruebas está conformada por la serie de procesos y actividades que demuestran los niveles de calidad del software, aspectos de eficiencia, solidez, compatibilidad, respuesta, cumplimiento de requerimientos y diversos parámetros cuantitativos que definen si el sistema cuenta con la robustez necesaria para culminar el proceso de desarrollo o, si es necesario regresar a la línea de producción para perfeccionar aquello que lo requiera.

Si dicho retroceso se presenta, será necesario repetir el ciclo de fases en la línea de producción para pulir el desarrollo, y con ello, repetir pruebas que permitirán medir nuevamente la calidad de software. A este proceso de repetición cíclica se le conoce como iteración, mientras que a los ajustes en pro del software se les denomina incremento. Es por ello que las metodologías que tienen la apertura de presentar estas mejoras cíclicas son conocidas como iterativas e incrementales, dado que suman gradualmente los resúmenes de resultados de las pruebas realizadas, aportando mejoras en beneficio global del software.

10. Participación profesional

La propuesta y los procedimientos de resolución que planteé en este problema están completamente vinculados a una carrera afín a las tecnologías de la información, puesto que inicialmente el banco involucrado ya cuenta con un sistema informático que diariamente cumple diversas funciones que facilitan las tareas de los departamentos que lo conforman y brinda diversos servicios financieros a los clientes, así como enlaces con múltiples instituciones asociadas al sector bancario que permiten expandir los servicios a nivel mundial. Este es el motivo fundamental del por qué se opta por realizar la resolución de estas necesidades a través del diseño y la creación de un sistema de cómputo capaz de administrar cada necesidad y poder proyectarla en una solución viable.

Con base en la recopilación de información y el diseño de la propuesta de solución, pude identificar que el contenido de los temarios presentados en el plan de estudios de mi carrera son un cúmulo de herramientas que son ampliamente requeridas y aplicadas diariamente en la gestión de proyectos informáticos, dado estudié asignaturas que me permitieron trabajar con tecnologías y metodologías modernas y vigentes. Desde la capacidad de profundizar en el

análisis y la abstracción que me brindaron las asignaturas de la división de ciencias básicas, hasta la computación aplicada plenamente a la resolución de problemas y diversas necesidades, dado que continuamente es necesario conocer las condiciones que establece la arquitectura de un sistema en red para la interconexión de diversos sistemas, así como el diseño y análisis de una base de datos o la implementación y aplicación de las metodologías de desarrollo de software. Asimismo, es importante la presencia de conocimientos en lógica de programación para diversos lenguajes y el conocimiento de diversos sistemas operativos.

Determiné la creación de un problema real que involucrase los conocimientos de la elaboración de un proyecto completo de software y las diversas disciplinas que lo respaldan porque mi módulo de salida es Ingeniería de Software y ahí conocí todas las posibilidades disponibles para la creación de sistemas informáticos para la solución de problemas.

Para ello, tuve que establecer a fondo los detalles de este problema, así como los parámetros que demanda su solución correcta y apegada a los estándares de calidad que establece la Administración de Proyectos de Software y la propia Ingeniería de desarrollo de software en su más amplio alcance. De igual manera, la profundización de los conocimientos que adquirí para entender un problema real y saber cómo atacarlo para resolverlo por mí mismo, me permitió analizar y abstraer distintos conceptos a lo largo de mi preparación en esta tesis, para poder discernir con qué herramientas puedo facilitar este desarrollo y cuáles otras no podrían participar correctamente en estas tareas.

Una vez arrancando con la propuesta de solución de este proyecto, pude comenzar a detectar inconsistencias en la documentación y en las buenas prácticas con las que me desarrollé como estudiante, con respecto a todas las actividades y prácticas que ya se realizan más formalmente en la industria y en el sector del desarrollo de software. Esto es debido a que cada empresa tiene manuales de procedimientos propios, así como usos y costumbres adaptados a su forma de trabajo, también en algunas ocasiones, hay técnicas propias que están pensadas en facilitar la producción de elementos y/o sistematizar determinadas tareas orientadas a su normatividad interna. Esto refleja que en una empresa tan grande como lo es un banco, los procedimientos y/o la obtención de respuestas e información, no siempre están al alcance de quién la requiere. Esto debido al gran tamaño que representa un banco, tanto desde el punto de vista del personal como del control y almacenamiento de información; destacando que la importancia de mantener la confidencialidad de datos sensibles limita el acceso a bases de datos de alta privacidad.

Particularmente, en el caso de soluciones para el desarrollo de software, es común ver que las empresas tengan perspectivas muy diferentes con respecto a las definiciones y/o técnicas que establece la Ingeniería de Software desde su definición. Esto se aprecia en la realización personalizada de metodologías, cambios en las responsabilidades de los involucrados en el proyecto, y/o la variedad de desarrolladores y en general, hasta la búsqueda por la reducción de presupuesto, ya sea en nóminas, tiempos, adquisición de herramientas, equipos, etcétera.

Cabe señalar que para la propuesta de solución que mencioné, la elección de la estructura “SDLC”, obedeció a que la he trabajado anteriormente, y más concretamente en el cargo de integrador, es por ello que, además de establecer “SDLC” como una vía factible para la creación del software requerido, introduje el rol de integrador al proyecto, sabiendo de antemano, que no necesariamente se define su participación en este proceso de desarrollo.

De igual manera, todos los procesos de la ingeniería de software pueden ser enriquecidos continuamente por profesionales de otras áreas que ayuden a comprender mejor intereses y riesgos del diseño de un programa informático. Por ejemplo, y para este caso particular, el comprender aspectos legales interactuando con especialistas en materia de derecho y áreas jurídicas del banco podría suponer un mayor conocimiento de los motivos que influyen fuera del banco, dado que la orden gubernamental que da origen a la creación de “DecoTurf” viene de una agencia independiente y externa a esta institución bancaria.

10.1. Asignaturas involucradas

Dado el plan de estudios que culminé (UNAM, Facultad de Ingeniería, carrera Ingeniería en Computación, plan de estudios 2010)⁶ esta propuesta de solución está basada en el contenido del temario de las siguientes asignaturas:

- Computación para ingenieros
- Ingeniería de Software
- Sistemas de comunicaciones
- Lenguajes de programación
- Computación gráfica
- Administración de proyectos de software
- Bases de datos
- Administración de redes
- Verificación y validación de software
- Arquitectura cliente/servidor

De igual forma, cabe señalar que gran parte del temario fue complementado y profundizado con información más precisa y orientada a la aplicación en la industria, así como áreas específicas y diversa documentación de Ciencias de la Ingeniería e Ingeniería Aplicada.

11. Resultados obtenidos

Es muy importante mencionar que cada fase del proyecto arrojó como resultados diversos elementos plenamente orientados a los estándares generales que el banco establece en cada inicio de proyecto, así como los que se trazaron particularmente para este. Además, si bien,

dichos estándares están basados en herramientas, prácticas y marcos de trabajo universales, siempre tienen adaptaciones que buscan optimizar los resultados pretendidos hacia parámetros propios del banco.

Ejemplos claros de esto, son las relativas facilidades que se le brindaron al proyecto para aceptar cambios, dado que una condición que se practicó a lo largo del ciclo de vida de desarrollo del software consistió en analizar los cambios con relación a los riesgos, pero siempre preponderando culminar el proyecto como se solicitó inicialmente (basado en requerimientos funcionales completamente cubiertos), para que posteriormente se brinden las facilidades mencionadas para realizar un análisis que determine si dichos cambios pueden ser añadidos en fases de actualizaciones y/o mantenimientos posteriores al “release” de este proyecto, o si su tamaño demanda la planeación como proyecto completo alterno a este.

De igual manera, es notable que el equipo de desarrollo realice observaciones al respecto de las técnicas comunes que se desenvuelven en un entorno de alta eficiencia, ya que no siempre empatan con las buenas prácticas que establece un aprendizaje “purista”, y que en su momento tiene el alcance de definir conceptos importantes en torno a dicho desarrollo. Un ejemplo claro de esta técnica que no empata con buenas prácticas es la siguiente:

“El desarrollo de vistas para una base de datos en un proyecto tan grande y a través de un manejador que almacena datos remotamente, puede sufrir modificaciones en beneficio de la eficiencia del sistema. Dado que el número de consultas a la base de datos es alto, además entrega numerosos “arrows” (tuplas) e implica múltiples conexiones a la base de datos de origen, se recomienda que las vistas sean desarrolladas como tablas completas, pero con alojamiento local”

-Comentarios de un DBA especializado en data warehouse bancario, con 18 años de experiencia en la industria.

Lo anterior representa conceptos que buscan jerarquizar los intereses específicos de un sistema involucrando ciertas prácticas que pueden ser consideradas incorrectas. Y es precisamente una base de datos el sistema que más fácilmente se puede prestar para alcanzar estos objetivos, ya que cuando se prioriza el rendimiento de la base de datos, habrá ocasiones en que se haga presente la redundancia y/o duplicidad en dicha base. Lo anterior obedeciendo a criterios de optimización, desempeño y reducción de la cantidad de consultas al manejador alojado de la nube.

Esta relación descrita en la matriz permite ordenar y coordinar las pruebas aplicadas al software una vez finalizado. Se puede identificar que se cubren satisfactoriamente la totalidad de los requerimientos funcionales en los que se basó este proyecto, siempre

respetando el orden y el algoritmo diseñado. Tanto en las pruebas de integración de sistemas de software (llamadas "SIT") como en las pruebas de aceptación del usuario (llamadas "UAT"), se puede constatar que cada uno de los eventos esperados se cumplieron, y que los cambios realizados a los componentes nuevos no conflictúan con ningún otro componente existente, así como su correcta interacción con dispositivos que el cliente manipula (tarjeta de débito y teléfono celular). Dichos resultados son documentados digitalmente, se realizan reuniones de demostración con las áreas solicitantes de este proyecto y se realizaron demostraciones "SIT" con una efectividad del 100% en los eventos mostrados. El comité responsable de validar estas pruebas tiene la obligación de firmar documentos de respaldo a manera de visto bueno por el cumplimiento total y formal de este proyecto.

A continuación, se presenta una matriz de trazabilidad que verifica el cumplimiento de los requerimientos manifestados anteriormente en el desarrollo del proyecto:

Estado/Componente	Cajero automático	Base de datos	Alta/baja de tarjetas	Servicio de mensajería	Microservicio de tiempo/plazos
Activación de tarjeta y alta de número telefónico	✓	✓	✓		
Validación de datos		✓			
Envío de notificación				✓	
Espera de 720 horas					✓
Cumplimiento de 2160 horas					✓
Bloqueo de tarjeta			✓		

Cuando el proyecto concluya completamente, los responsables de cada área y las personas que participan tendrán una última reunión de cierre para dar por finalizado el proyecto y se entregará el reporte correspondiente incluyendo firmas autógrafas y digitales en la documentación, así como el aval de stakeholders que solicitaron este desarrollo y establecieron los requerimientos al principio. Las actividades subsecuentes a esta reunión de cierre culminarán (y ya con aval del comité solicitante) se procederá con las tareas para el "release" del proyecto, en donde se instalará formalmente cada componente y pieza en el sistema global del banco, en horarios inhábiles, con poca afluencia de transacciones y siempre monitoreando a cada instante los eventos y reportes que el equipo de instalación comenten, así como también, se supervisará el archivo log de ventos de sistema. También se

debe tomar en cuenta que, si este proyecto es desarrollado a la par de otros, y los diversos componentes que se requieren desarrollar para la implementación tienen alguna relación y/o no fueron terminados en su totalidad, esto puede retrasar el release para emparejar con ellos.

La documentación es altamente importante por diversos motivos, como son, la certificación de aquello solicitado contra lo que es entregado, auditorías, resguardo de información confidencial, seguimiento de entregables, reporte de incidencias, gestión de calidad, actualizaciones, mantenimiento, escalabilidad y en general cualquier otro objetivo que persigan los estatutos del departamento de Tecnologías de Información del banco en eventos posteriores a la entrega de un software. Es aquí donde se pone de manifiesto que es imperativo poner en orden cada uno de los documentos, registros e históricos de actividades y piezas modificadas, así como también los registros de posibles depuraciones. De igual manera, las prácticas bancarias deben dejar en claro los procedimientos de respaldo correspondientes a la información como también a los sistemas como tal.

12. Conclusiones

En lo que corresponde a la experiencia de aplicar los conocimientos de la carrera e investigar detalladamente todos los elementos necesarios para el planteamiento de un problema, su análisis y la resolución más adecuada y eficiente del mismo, fui capaz de darme cuenta de que las herramientas adquiridas a lo largo de mi formación académica como ingeniero son esenciales para el ejercicio de una profesión tan cambiante y que se mantiene en constante evolución como lo es la Ingeniería en Computación. De igual manera, considero que mi puesta en marcha de una solución formal y estructurada en búsqueda de solucionar un problema real en un campo laboral establecido y en una empresa tan compleja en el plano tecnológico, me fue muy nutritiva, completa y altamente formadora, dado que pude conocer algunos de los diversos conceptos particulares y orientados totalmente a un ambiente bancario y más específicamente a un área de Tecnologías de Información, así como los altísimos parámetros de seguridad informática que representan gran parte de las inversiones en recursos que maneja un banco para proteger todos sus activos. También pude ver que la distribución de las actividades para los participantes del desarrollo del software va más allá de lo que establece cada metodología, porque depende de las habilidades individuales, y que cada procedimiento establecido siempre es adaptado a las diversas necesidades y lineamientos que impone un banco para cubrir sus propios intereses, tanto dentro, como fuera de los alcances del proyecto, inclusive, cuando también se involucran empresas de tecnología que ofrecen ser aliados en el desarrollo tecnológico y se sientan las bases para beneficios mutuos.

Otros aspectos que acentúan la trascendencia de la carrera de ingeniería en computación con la solución de necesidades a nivel industrial son las diversas capacidades de generar sistemas

informáticos que pueden ser tan complejos o simples como se presente en cada requerimiento(s) en particular, o en su defecto, manifestando la posibilidad de englobar necesidades particulares de uno o más proyectos en uno solo, con el objetivo de promover la eficiencia de sistemas, controlar de manera óptima ambientes únicos, o incluso, reducir presupuestos . Es decir, si tomamos en cuenta que un problema como el que utilicé para este caso de estudio, las grandes ventajas son, que se está apoyando ampliamente en múltiples sistemas informáticos más grandes y numerosos aplicativos que ya brindan servicios en específico, y tienen funcionamientos que facilitan la creación de otros sistemas, porque desde un punto de vista global, podría considerarse que “DecoTurf” es una escalabilidad o actualización de una parte determinada de un complejo sistema conformado por otros múltiples sistemas que interactúan entre sí para poder cumplir múltiples objetivos particulares y generales simultáneamente. Entonces, partiendo de esa base, se presume que el futuro que tiene un sistema global como éste, tanto a mediano como a largo plazo, sería continuar su expansión sin un límite máximo de alcances u objetivos, tanto para generar muchos más sistemas nuevos, como para modificar los que ya existen. Sin descartar la posibilidad de que, conforme avancen las necesidades de la institución bancaria, así como las mejoras en sistema, hardware, conectividades e incluso la participación activa de los clientes, también es posible que algún software pueda ser deshabilitado o eliminado parcial o totalmente de su operación.

Concluyo concretamente afirmando que la propuesta de solución con el proyecto diseñado cumple a plenitud y puede ser exitoso en la ejecución formal y completa de cada uno de los requerimientos funcionales, expuestos para este caso de estudio, tanto en virtud de la normatividad legal a la que está sujeta esta entidad bancaria como también en los lineamientos que contempla el propio manual de procedimientos que rige al desarrollo o actualización de software dentro del banco. No cabe duda de que un análisis correcto y profesional del problema, así como sus necesidades, son indispensables para invocar las herramientas adecuadas que permiten atacar un problema de una forma simplificada y eficiente. Ello, siempre estacando que la profundidad de las acciones solicitadas y la complejidad que se presenta para poder alcanzar las metas en cada requerimiento funcional son demandantes y requieren de establecer índices de calidad en sus componentes, debido a la delicadeza de interactuar con ambientes y aplicativos que van a brindar servicios tanto a “DecoTurf”, como a otros sistemas que lo demanden, así como el cuidado de la información sensible. Por lo que debo hacer mención de que las condiciones operativas que se plantean dependen mucho de la eficiencia en cada fase de desarrollo y exigen alta coordinación en cada una de las líneas de trabajo y por parte de los profesionales involucrados. Afortunadamente, las semanas dedicadas a ciclos de iteración, son horas que brindan grandes facilidades para generar mejoras particulares en la calidad del software entregado como resultado de dicha iteración a través de los resultados analizados de las pruebas, lo que favorece la superación nivel funcional del software, así como su calidad, lo que tiende a garantizar que se pueda finalizar todo el ciclo del proceso de desarrollo satisfactoriamente y con los resultados correctos.

13. Apéndices

Apéndice #1

A lo largo de gran parte del contenido de esta tesis, utilicé terminología propia de la ingeniería de software en donde pude dejar entrever ciertas similitudes importantes y usos casi indistintos entre los conceptos “metodología de desarrollo de software” y el proceso conocido como “SDLC”. A manera de definición concreta de los conceptos mencionados, cito lo siguiente:

Metodología de desarrollo de software:

La propuesta del VTT Technical Research Centre of Finland con respecto a modelos de procesos es la siguiente:

“La idea básica es establecer un proceso de desarrollo de software que debe progresar desde una fase a otra, cada una produciendo salidas que se utilizan como entradas en la siguiente fase. El modelo define fases, actividades y productos que deben completarse en estricto orden.”⁷

De igual manera, la definición de una enciclopedia en línea explica lo siguiente:

“Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Esta definición se presenta dentro del contexto de ingeniería de software.”⁸

SDLC: por sus siglas en inglés Systems* Development Life Cycle, y su significado en español, Ciclo de Vida de Desarrollo de Sistemas:

La definición del profesor omaní Geoffrey Elliot en 2004 fue:

“El ciclo de vida de desarrollo de sistemas puede considerarse el marco metodológico formalizado más antiguo para la construcción de sistemas de información. La idea principal del SDLC ha sido "perseguir el desarrollo de sistemas de información de una manera muy deliberada, estructurada y metódica, requiriendo cada etapa del ciclo de vida - desde el inicio de la idea hasta la entrega del sistema final - para llevarse a cabo de forma rígida y secuencial”⁹

También, una enciclopedia en línea enuncia lo siguiente:

“Es un proceso para planificar, crear, probar e implementar un sistema de información. El concepto del ciclo de vida del desarrollo de sistemas se aplica a una variedad de configuraciones que pueden ser de hardware, de software, o de ambos.”¹⁰

*Ocasionalmente, la letra S puede ser definida como “Systems” o como “Software”, según el autor y/o contexto del desarrollo.

Basado en las definiciones, se puede apreciar que las diferencias conceptuales entre “metodología” y “SDLC” son sutiles, por ello, es común que los responsables de emplear SDLC como marco de trabajo para desarrollar un sistema, lo nombran y trabajan como un concepto paralelo al término “metodología de desarrollo de software”, o únicamente “metodología”. Situación que es más acentuada cuando se considera que los ambientes laborales en donde se desarrolla software pueden tener procedimientos y prácticas que generan que operativamente sean términos casi homólogos.

Dicho lo anterior, y para efectos de precisar conceptos e ideas en el presente documento, estoy considerando que los planteamientos y aplicaciones de fases, etapas y actividades que plantea SDLC corresponden igualmente a lo que estipula una metodología plena en su marco de trabajo (independientemente de su tipo y/o clasificación). Reconociendo que esto pueda no ser técnicamente correcto.

Apéndice #2

Basado en la cita que menciono en la sección de “Resultados obtenidos” relacionada a comentarios de un DBA al respecto del manejo de una base de datos en la industria, y en correspondencia a la propuesta de la necesidad de base de datos que propongo en la sección de “Manejo y almacenamiento de información”, quiero hacer los siguientes comentarios:

Comprendo la importancia que ameritan todos los aspectos relacionados con una base de datos en un proyecto, desde la designación del DBA, la tecnología que se empleará, el manejador de base de datos y la propuesta de diseño que ésta requiere; sin embargo, en mi (todavía) breve experiencia en la industria, tuve contacto directo con expertos en el tema, pude conocer las prácticas que se llevan a cabo y las soluciones que se aplican a cada necesidad, y derivado de todo ello, pude identificar que, desafortunadamente no se cumplen las buenas prácticas que aprendí a lo largo de mi preparación en la Facultad de Ingeniería para una buena administración de una base de datos.

Conozco a plenitud que es altamente importante proponer una base de datos casi como un proyecto completo, paralelo al proyecto principal, porque sé los alcances y el impacto que

esto demanda. Pero para efectos de este proyecto, en donde el almacenamiento de información es relativamente limitado, y donde establezco que no es necesario generar un sistema que cuente con altos privilegios de interacción con la base de datos del banco, creo que es muy importante reconocer que estoy consciente del bajo perfil en el que dejo a la sección de “manejo y almacenamiento de información”.

Dicho lo anterior, comprendo plenamente que es posible proponer una base de datos funcional a través de diferentes modelados, mismos que podrían tener más impacto e interacción con la base de datos general del banco, (incluso, dedicando un modelo expreso para este proyecto, sin interactuar con la base ya existente) pero también estoy preponderando que la información que dicha base de datos ya contiene, demanda niveles de seguridad y eficiencia superiores a los que Decoturf podría aspirar, es por ello que, para cuestiones de eficiencia y simplicidad, terminé por proponer algo más básico.

14. Glosario

ADOBE FLEX: Este término hace referencia a un conjunto de tecnologías de soporte para desarrollo y despliegue de aplicaciones ejecutables en internet que particularmente se denominan “aplicaciones enriquecidas”. Estas tecnologías están basadas en la plataforma propietaria FLASH, anteriormente conocida como MACROMEDIA, y que hoy se llama ADOBE.

Api: Denominada en español como interfaz de programación de aplicaciones y en inglés como application programming interface. Se refiere a una subrutina o conjunto de subrutinas que brindan acceso a ciertas bibliotecas de software para obtener información y/o cumplir funciones específicas. Una API permite un vínculo de comunicación entre uno o más componentes de software.

Arrow: O “tupla”, en español. Es el contenido de datos del resultado de una consulta concreta en un manejador de bases de datos. Corresponde a la información que en conjunto contienen un dato por cada columna de la consulta invocada, aunque la consulta también puede arrojar un dato vacío (blank).

Backend: Se refiere a toda aquella sección de sistema cuyo alcance está ubicado en alguna capa que permite el acceso a datos y que puede ser de nivel de servicio, administración o control del sistema. La operación backend suele estar oculta del usuario, y su ejecución, generalmente, depende de otras capas o enlaces que se pueden manipular desde frontend.

CORBA: Common Object Request Broker Architecture. Es un estándar propuesto por el consorcio Object Management Group. Tiene como finalidad facilitar la comunicación entre múltiples componentes de software que fueron desarrollados en diferentes lenguajes de programación y que pueden ser ejecutados en diferentes computadoras.

Data warehouse: Es un almacén de datos. Comúnmente este término se refiere específicamente a datos informáticos. Ralph Kimball, un computólogo y arquitecto de sistemas estadounidense famoso en los años 80s y 90s, definió a un data warehouse como: "Es un almacén de datos que extrae, limpia, conforma y entrega una fuente de datos dimensional para la consulta y el análisis".¹¹

DBA: Es el acrónimo de Data Base Administrator. Es el administrador de una base de datos

Frontend: Es aquella sección de un sistema que está pensada en ser colocada para tener contacto directo con el usuario. Este segmento del sistema suele estar representado por capas de interfaces gráficas que contienen gráficos y/o animaciones que puede ver el usuario que interactúa con el sistema.

Gsm: del inglés Global System for Mobile communications. Es el sistema global para las comunicaciones móviles de segunda generación (2G). Es un sistema estandarizado para las comunicaciones móviles inalámbricas propuesta y desarrollado por el European Telecommunications Standards Institute.¹² Al 26 de agosto de 2017, se estima que GSM es el estándar en telecomunicaciones móviles más extendido en el mundo, con un 82% de los terminales mundiales en uso, y un aproximado de 3000 millones de usuarios en 159 países distintos.

Hacker: Término en inglés, pronunciado en español como "jáquer". Se define como "persona con grandes habilidades en el manejo de computadoras que investiga un sistema informático para avisar de los fallos y desarrollar técnicas de mejora".¹³ Por lo general es sinónimo de "pirata informático", acepción que es mal empleada.

Http: En inglés Hypertext transfer protocol, y en español, protocolo de transferencia de hipertexto. Es un protocolo de comunicaciones que permite transferencias de archivos en formatos dedicados a la World Wide Web (red informática mundial).

Issue, issues: Término en inglés cuya traducción al español es "cuestiones". En el ambiente informático este término está relacionado a la detección de un fallo informático, que compromete a su correspondiente arreglo. Ocasionalmente, también es empleado como sinónimo de "tarea solicitada", "requerimiento" o una "solicitud concreta originada por el equipo de trabajo".

J2EE: Java™ 2 Platform, Enterprise Edition: Es una plataforma de programación para el desarrollo y ejecución de aplicaciones en lenguaje JAVA.

J2SE: Java Platform, Standard Edition: Es una colección de APIs, utilizada para el desarrollo de componentes en el lenguaje de programación JAVA.

JDBC: Java™ 2 Database Connectivity: Es una API para el lenguaje de programación JAVA especializada en operaciones sobre bases de datos en dicho lenguaje.

LIGHTSPARK: Es un software de código abierto, desarrollado por Alessandro Pignotti, bajo la licencia de Lighthspark Developers que permite la reproducción de archivos SWF en sistemas operativos Windows y Linux.

Linux: Es una distribución de software basada en el núcleo de nombre Linux. Aunque hay proyectos de desarrollo de aplicaciones e interfaces para usuarios basadas en el núcleo Linux, su mayor campo de desarrollo se orienta hacia los sistemas operativos (Red Hat, Fedora, Knoppix, etcétera...)

Microsoft: Es una compañía tecnológica multinacional especializada en el desarrollo de software.

Pishing: Es un término netamente informático, que se utiliza para identificar aquellas técnicas que tienen como finalidad el engaño a una víctima para obtener su confianza y de esa manera poder robar información importante para diversos objetivos, como lo son el acceso a sistemas, la suplantación de identidad, compromiso de cuenta, extorsión, o cualquier otro delito informático que se derive de obtener beneficios de estas técnicas de fraude.

Release: liberación o lanzamiento. Este término hace referencia al evento en el que se ha culminado totalmente el software, y está listo para ser instalado e iniciar operaciones.

SMS: en inglés, Short Message Service, en español, servicio de mensajes cortos. Es un servicio de mensajes de texto entre teléfonos móviles limitado a cierta cantidad de caracteres. Inicialmente incluido en el estándar GSM, los SMS han prevalecido hasta las redes más actuales como la 4G.

SQL: Structured Query Language, o en español, Lenguaje estructurado de consulta. Es un lenguaje específico de dominio utilizado particularmente para programar, diseñar y administrar datos almacenados en un manejador de bases de datos relacionales.

SWF: Es un formato de archivos orientado a multimedia que cuenta con posibilidad de generar animación, que específicamente contienen gráficos vectoriales, aunque es compatible con mapas de bits.

Web service: Se refiere a todo aquel servicio que se brinda entre sistemas mediante protocolos y estándares desde un servidor a una red para intercambiar datos entre aplicaciones.

WINDOWS XP: Es una versión de la gama de sistemas operativos de la familia de Microsoft Windows, creado por Microsoft y lanzado en octubre de 2001. Debido a su estabilidad, compatibilidad y características novedosas para su tiempo, el soporte de este sistema operativo llegó a mantenerse hasta abril de 2014.¹⁴ Luego de 20 años de su lanzamiento, y después de casi 8 años de dejar de recibir soporte oficial, se estima que Windows XP es un sistema operativo que representa una cuota activa de alrededor del 0.42% del mercado para el último trimestre del 2021.

XML: en inglés, eXtensible Markup Language, en español, Lenguajes de Mercado Extensible. Es un metalenguaje que permite definir lenguajes de marcas utilizado para almacenar datos en forma legible.

15. Referencias

1. Artículo del Ministerio de Cultura de la Junta de Galicia: “*La historia de la tecnología*”
[https://www.edu.xunta.gal/espazoAbalar/sites/espazoAbalar/files/datos/1464945204/contido/12 la historia de la tecnologia.html](https://www.edu.xunta.gal/espazoAbalar/sites/espazoAbalar/files/datos/1464945204/contido/12%20la%20historia%20de%20la%20tecnologia.html)
2. Artículo del blog de la IEBS Digital School, escrito por la periodista Sonia Mañé Vernia
<https://www.iebschool.com/blog/ventajas-desventajas-tecnologia/>
3. Artículo del Sistema de Medios de la Universidad Nacional de Cuyo, Argentina: “*Historia y evolución de los teléfonos celulares: ¿con cuál empezaste?*”
<https://www.universidad.com.ar/historia-y-evolucion-de-los-telefonos-celulares-con-cual-empezaste>
4. Publicación en línea de la revista “*Muy Interesante*”. Julio de 2021, adaptada por Nacho Grosso
<https://www.muyinteresante.es/tecnologia/articulo/la-historia-del-sms-un-sistema-que-se-resiste-a-desaparecer-691626879691>
5. Normatividades vigentes en julio 2021 de la Comisión Nacional Bancaria y de Valores.
<https://www.cnbv.gob.mx/Paginas/NORMATIVIDAD.aspx>
6. Plan de estudios de la carrera de Ingeniería en Computación, plan 2010 de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, campus Ciudad Universitaria.
https://www.ingenieria.unam.mx/programas_academicos/licenciatura/computacion_plan2010.php

7. Publicación #535 de la revista de VTT *Technical Research Centre of Finland*. Junio de 2004, escrita por Seija Komi-Sirviö, p. 15
<https://www.vttresearch.com/sites/default/files/pdf/publications/2004/P535.pdf>
8. Wikipedia. Artículo: Metodología de desarrollo de software.
https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software
9. Geoffrey Elliott & Josh Strachan (2004) *Global Business Information Technology*. p. 87
10. Wikipedia. Artículo: *Systems Development Life Cycle*.
https://en.wikipedia.org/wiki/Systems_development_life_cycle
11. Kimball, Ralph (2004). *The Data Warehouse ETL Toolkit* (en inglés). Wiley. p. 23.
12. Wikipedia. Artículo: *Sistema global para las comunicaciones móviles*.
https://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_m%C3%B3viles
13. Diccionario de la Real Academia de la Lengua Española.
<https://dle.rae.es/j%C3%A1quer>
14. Wikipedia. Artículo: *Windows XP*.
https://es.wikipedia.org/wiki/Windows_XP

16. Bibliografías

Impresas

- Roger S. Pressman. (2010). *Ingeniería del Software. Un enfoque práctico*. Nueva York: McGraw-Hill.
- Ian Sommerville. (2011). *Ingeniería de Software 9*. México: Pearson Educación.
- Jacobson, I., Booch, G., Rumbaugh, J. (2003). *El Proceso Unificado de Desarrollo de Software*. España: Addison Wesley.
- (múltiples autores y colaboradores). (2013). *Guía de los Fundamentos para la dirección de proyectos (Guía del PMBOK)*. Estados Unidos: Project Management Institute, Inc.
- Craig, Larman. (2010). *UML y patrones. Una introducción al análisis y diseño orientado a objetos y proceso unificado*. España: Pearson Prentice Hall.

Electrónicas

- Fischer, Gerhard. (2001). *The Software Technology of the 21st Century: From Software Reuse to Collaborative Software Design*. 11 de agosto de 2021, de Universidad de Colorado Sitio web: <http://l3d.cs.colorado.edu/~gerhard/papers/isfst2001.pdf>
- Desconocido. (Revalidado en 2008). *Selecting a development approach*. 24 de septiembre de 2021, de Department of health & Human Services - USA Sitio web: <https://web.archive.org/web/20190102182947/https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
- Zepeda Gorostiza, Adán. (2012). *Material para clase de Ingeniería de Software*. 02 de junio de 2021, de Universidad Nacional Autónoma de México Sitio web: <http://profesores.fi-b.unam.mx/adanzg/>
- Belzarena, P., Vallespir, D. (2012). *Métodos de Gestión de riesgo de proyectos de Software*. 17 de junio de 2021, de Universidad de la República Uruguay Sitio web: <https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/2967/1/tesis-jaureche.pdf>