

Leguizamón, Gonzalo Uriel

Superclub canje-post compra

2020

Instituto: Ingeniería y Agronomía
Carrera: Ingeniería en Informática



Esta obra está bajo una Licencia Creative Commons Argentina.
Atribución - No Comercial - Compartir Igual 4.0
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Documento descargado de RID - UNAJ Repositorio Institucional Digital de la Universidad Nacional Arturo Jauretche

Cita recomendada:

Leguizamón, G. U. (2020) Superclub canje-post compra [Informe de la práctica Profesional Supervisada] Universidad Nacional Arturo Jauretche

Disponible en RID - UNAJ Repositorio Institucional Digital UNAJ <https://biblioteca.unaj.edu.ar/rid-unaj-repositorio-institucional-digital-unaj>

Universidad Nacional Arturo
Jauretche

Instituto de Ingeniería y agronomía

Ingeniería en Informática



TRABAJO FINAL DE LA PRACTICA
PROFESIONAL SUPERVISADA

*Superclub Canje-Postcompra
micrositio web*

Estudiante:

Gonzalo Uriel Leguizamón

Tutores:

Lic. Paula Bein

Ing. Brian Galarza

Mg. Ing. Diego Encinas

Lic. Miguel Ángel Almada

Buenos Aires, 2020

PRÁCTICA PROFESIONAL SUPERVISADA (PPS)
Superclub Canje Postcompra
Informe Final

DATOS DEL ESTUDIANTE

Apellido y Nombres: Leguizamón Gonzalo Uriel

DNI: 38369024

Nº de Legajo: 8304

Correo electrónico: gonzalourie32@gmail.com

Cantidad de materias aprobadas al comienzo de la PPS: 45

PPS enmarcada en artículo (4 ó 7) de la Resolución (CS) 103/16: Artículo 7 inciso: b

DOCENTE SUPERVISOR

Apellido y Nombres: Ing. Brian Galarza, Mg. Ing. Diego Encinas

Correo electrónico: BrianGalarza@hotmail.com, dencinas@unaj.edu.ar

DOCENTE TUTOR DEL TALLER DE APOYO A LA PRODUCCIÓN DE TEXTOS ACADÉMICOS DE LA UNAJ

Apellido y Nombres: Lic. Paula Bein

Correo electrónico: paula.bein@gmail.com

DATOS DE LA ORGANIZACIÓN DONDE SE REALIZA LA PPS

Nombre o Razón Social: Santander Tecnología S.A

Dirección: Uspallata 2953, C1437 CAB

Teléfono: 011 5030-4100

Sector: Software Factory/ Bancario

TUTOR DE LA ORGANIZACIONAL

Apellido y Nombres: Lic. Miguel Ángel Almada

Correo electrónico: malmada@santandertecnologia.com.ar

FIRMA DEL COORDINADOR DE LA CARRERA



CANJE POSTCOMPRA

Banco Santander Rio S.A

1. Introducción	4
2. Desarrollo	9
2.1 Tecnologías utilizadas	9
2.2 Metodología de desarrollo	18
2.3 Ejecución del Cronograma de tareas	20
2.3.1 Relevamiento	20
2.3.2 Desarrollo	23
• Logica PL/SQL movimientos a premiar	23
• Logica PL/SQL asignación de premios	34
• Logica PL/SQL topes de reintegro y caducidad	38
• Logica java para filtros finales de estado	43
• Logica batch automatismo del proceso	49
2.3.3 Test y Validación	55
• Pruebas unitarias de funcionalidad	56
• Prueba de regresión	62
2.3.4 Implementación	68
3. Conclusión	73
3.1 Resultado, posibles mejoras y ampliaciones	73
3.2 Reflexión sobre las prácticas profesionales como espacio de formación	74
4. Anexo	76
4.1 Glosario	76
4.2 Bibliografía	77

1. INTRODUCCIÓN:

Esta práctica profesional supervisada (PPS) tiene como propósito presentar un plan de trabajo para el desarrollo del sistema web llamado "Canje Postcompra" implementado en el banco Santander Río. A grandes rasgos, el mismo se apoya en una serie de filtros lógicos sobre información recopilada de varias bases de datos, para ofrecer comodines de descuento como premiación a los consumos de los clientes con tarjetas de débito y crédito en los rubros más masivos de transaccionalidad, como: peluquerías, restaurantes, indumentaria, combustibles y supermercados.

Luego de realizar el pertinente estudio, donde se midió el mercado que abarca el contexto de operación en un país en constante fluctuación; especialmente en la competencia directa como son los programas de beneficios de bancos HSBC, BBVA, Galicia, entre otros; quienes ofrecen sistemas de niveles en los que se va subiendo según consumos y se premia con cuotas sin interés o reintegros (pero a mes vencido). Se observó que hoy día no se brinda algo similar que actúe en la inmediatez de los consumos del día a día de los clientes, como se pretende hacerlo acá. Por lo tanto, se tiene la oportunidad, a través de este nuevo evolutivo, de llevar a los clientes del banco un plus de utilidad que genere un impacto en sus operaciones cotidianas.



ILUSTRACIÓN 1 - FACTORES DE UN ESTUDIO DE MERCADO

FUENTE: [HTTP://MICRO-CREATUEXITO.BLOGSPOT.COM/2016/05/ANALISIS-DEL-MERCADO.HTML](http://micro-creatuexito.blogspot.com/2016/05/analisis-del-mercado.html)

Dentro de un banco, aparecen varias áreas o sectores involucrados (que a su vez pueden presentar otras secciones) dentro del programa de Superclub, así como una entidad externa. Para mayor claridad, se describe a continuación con la ayuda del siguiente organigrama.

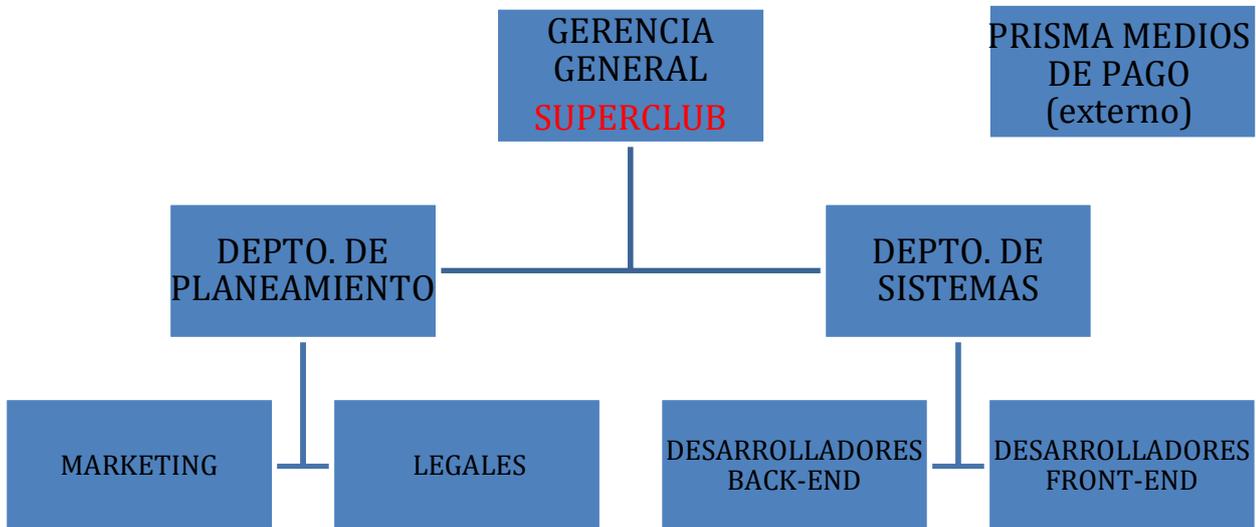


ILUSTRACIÓN 2 - ORGANIGRAMA SUPERCLUB

FUENTE: AUTORÍA PROPIA BASADO EN LO RELEVADO EN EL TRABAJO.



ILUSTRACIÓN 3 - SÍMBOLO SUPERCLUB

FUENTE: [HTTPS://WWW.PLATAFORMA10.COM.AR/PROMOCION/SUPERCLUB](https://www.plataforma10.com.ar/promocion/superclub)

Como se observa en el gráfico anterior, el sector en donde ocurre el desarrollo y funciona como “dueño” del evolutivo o sistema de Canje - Postcompra es Superclub. Este sector es básicamente el programa de fidelización de los socios adheridos al banco, donde se resuelven las necesidades de los clientes y trata de convertirlas en beneficios para que estos se sientan importantes, motivados a tener una actividad diaria en el banco y, por consiguiente, seguir consumiendo los productos que se le ofrece.

Las dos principales funcionalidades de éste programa son:

- ◆ **Acreditar acumulaciones de puntos a través de consumos con las tarjetas de crédito, en donde la relación equivale a 1 pto → \$3.**
- ◆ **Realizar canjes de los puntos acumulados por una variedad sin fin de premios de catálogo, los mismos van desde productos físicos como electrodomésticos, artículos para el hogar y demás, hasta productos virtuales como vouchers para tiendas, millas en despegar, comodines de descuento, etc.**

En la estructura de Superclub existe un mandato gerencial general, y de ahí se desprenden dos departamentos: el de Sistemas (los encargados del desarrollo) y el departamento de Planeamiento (los encargados de idear los nuevos evolutivos o beneficios).

Este último, a su vez, se subdivide en dos departamentos:

* El de Marketing, quien es el encargado de elaborar y presentar la propuesta de este nuevo beneficio de Canje - Postcompra para clientes; además de analizar su influencia en la relación cliente/banco e impacto monetario en la base de clientes actuales.

* El de Legales, quien define los términos y condiciones de la nueva funcionalidad para que quede enmarcada dentro de la ley, por esta razón es el que aprueba los contenidos (ya sea imágenes o textos del sitio web del nuevo beneficio).

* Por último, el de Sistemas, encargado del desarrollo, que está subdividido en dos sectores muy marcados: el backend/lógica (donde se realizan las actividades técnicas detalladas más adelante) y el frontend/gráfica (equipo de desarrolladores y maquetadores HTML).

Ambos departamentos, Sistemas y Planeamiento, tienen una interacción y retroalimentación constante ya que Marketing propone las ideas y Sistemas las acepta o sugiere cambios cuando algo no es muy viable dentro del entorno técnico a desarrollar, convirtiendo así a Marketing como “el cliente interno de sistemas”.

A su vez Legales también interactúa con Marketing señalándole las pautas publicitarias que pueden incluir o no en sus templates, imágenes y demás, convirtiéndolo así en “su” cliente interno.

El flujo de dichas interacciones puede verse graficado de la siguiente manera:





ILUSTRACIÓN 4 - SÍMBOLO PRISMA

FUENTE: [HTTPS://GRAMHO.COM/EXPLORE-HASHTAG/PRISMAMEDIOSDEPAGO](https://gramho.com/explore-hashtag/prismamediosdepago)

Otro de los actores en este desarrollo es Prisma Medios de Pago, una empresa dedicada a procesar las transacciones de distintas marcas (Visa, Amex y Mastercard) de casi todos los bancos. El papel que juega en este desarrollo se basa en varios tópicos:

- Prisma define e identifica los rubros contables que van asociados a los consumos que se quieren premiar
- Informa los movimientos con tarjetas débito y crédito que se sucedieron en el día a partir de los cuales se quiere trabajar
- El banco informa qué comodines de ahorro van a aplicar como premio a qué cliente y se informa por cuántos días tienen vigencia para realizar el canje del beneficio
- Devuelve diariamente un archivo que informa cuáles de los comodines ofrecidos se canjearon realmente y cuales se descartaron

Gracias a la retroalimentación entre Superclub y Prisma, es que se pueden canjear los premios y se procesan dentro de las fechas de vigencia que sistemas le indica.

Si bien los actores anteriores no definen exclusivamente la tecnología, ésta debe seleccionarse dentro del paquete de licencias que tiene el banco ya comprado, debido a que - como toda entidad bancaria- es auditada por el BCRA (Banco Central República Argentina) y no se pueden utilizar herramientas sin licencia. Para la construcción del sitio se van a emplear las siguientes tecnologías: PL/SQL, Batch Shell script y Java; además la metodología de programación denominada Waterfall o “cascada”. Las mismas se eligen por cuestiones meramente técnicas que serán fundamentadas en detalle más adelante.

Finalmente, para poder llevar a cabo lo mencionado, el proyecto estará dividido en diferentes etapas secuenciales: Relevamiento, Desarrollo, Test y Validación, Implementación. Cada una de ellas se describen con precisión en el desarrollo del presente informe.

2. DESARROLLO:

A continuación, se detallan los puntos relevantes del proyecto:

- Tecnologías utilizadas.
- Metodología de desarrollo.
- Ejecución del Cronograma de tareas.

2.1 Tecnologías utilizadas:

De todas las tecnologías que el banco ofrece, se detallarán las seleccionadas para realizar el proyecto así como se explicarán las razones de dicha elección



ILUSTRACIÓN 5 – MARCA PL/SQL

FUENTE: [HTTPS://CURSOS-FREECODE.BLOGSPOT.COM/2019/08/APRENDE-ORACLE-PLSQL-DESDE-CERO.HTML](https://cursos-freecode.blogspot.com/2019/08/aprende-oracle-plsql-desde-cero.html)

Oracle PL/SQL: Es el lenguaje de programación que viene embutido dentro de la base de datos y que desarrolla toda la lógica del proyecto que aquí nos compete. Éste permite soportar todas las consultas a la base Oracle ya que usa las mismas instrucciones que el SQL (lenguaje de consulta), agregándole especificaciones nuevas:

- Bucles
- Estructuras de control de flujos
- Manejo de variables
- Manejo de constantes
- Control de excepciones
- Estructuras modulares

Tiene una sintaxis estructurada en bloques de código y se almacenan dentro de la base de datos como objetos (Packages, Procedures, Functions, Triggers, etc.).

Estos objetos o elementos se ejecutan en el servidor Oracle de la base, por consiguiente no consume recursos del lado del cliente.

La arquitectura de este servidor en el que se encuentra alojada la base de datos y que soporta el lenguaje PL/SQL, es el siguiente.

ORACLE® DATABASE 12^c

ILUSTRACIÓN 6 – ARQUITECTURA BASE DE DATOS

FUENTE: [HTTPS://REVISTABYTE.ES/ANALISIS-BYTE-TI/ORACLE-DATABASE-12C/](https://revistabyte.es/ANALISIS-BYTE-TI/ORACLE-DATABASE-12C/)

La misma tiene como característica importante añadida la disponibilidad en la nube, que su predecesora 11c no tenía. Otras de sus potencialidades son: la arquitectura multiusuario para clientes (que mejora la calidad y el rendimiento de las aplicaciones disponibles ahorrando tiempo en lo que se supone la disponibilidad de la base), y las bases de datos enchufables (que sirven para crear capacidad de aprovisionamiento y portabilidad).

La elección de dicha arquitectura y lenguaje de base de datos descriptos se da por lo siguiente:

- Se va a trabajar con transacciones de clientes, lo cual hace que el volumen de datos a tratar sea inmenso, ya que si se multiplican los tres millones aprox. de clientes por las transacciones de cada uno, terminaría quedando un número desmedido de datos. (Actualmente las únicas dos tecnologías disponibles en el banco para procesar en tiempos cortos y aceptables tanto volumen de información son COBOL-lenguaje de programación- y PL/SQL) Se elige PL/SQL gracias a su infraestructura disponible en la nube con múltiples cores (núcleos) trabajando en paralelo en forma de hilos.

- La naturalidad en sí del objeto predominante con el que se trabaja, “transacciones”, como es el flujo de dinero en movimiento de un lado a otro, requiere trabajar con lenguajes transaccionales como lo es PL/SQL. Esto hace que las sentencias dentro de este lenguaje se procesen como un todo, los datos se validan o se cancelan como una única operación, lo que permite que no quede algo ejecutado por la mitad o a medio terminar.
- Las estructuras modulares o bloques de código que ofrece esta tecnología radica en que la sintaxis ya se encuentra interpretada por la instancia de la base de datos, por lo tanto reduce los tiempos de respuesta desde el lado servidor a las consultas que hace el lado cliente, como muestra el siguiente gráfico.

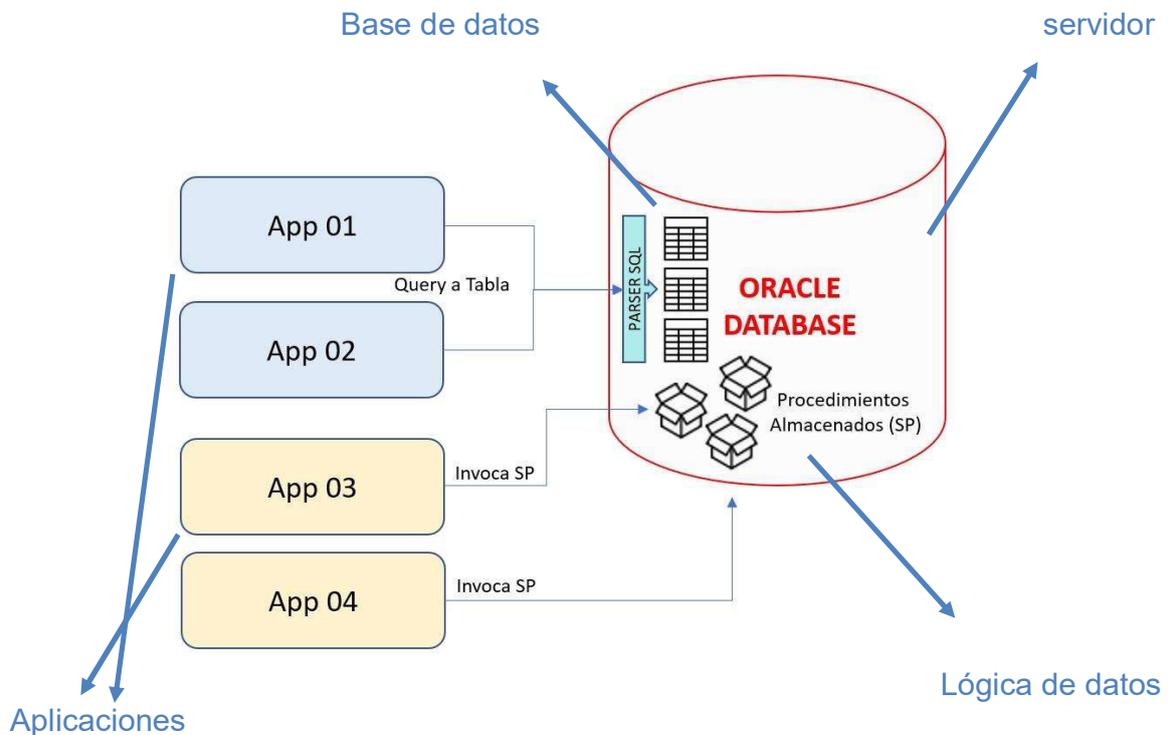


ILUSTRACIÓN 7 – ARQUITECTURA DE PL/SQL

FUENTE: [HTTPS://REVISTABYTE.ES/ANALISIS-BYTE-TI/ORACLE-DATABASE-12C/](https://revistabyte.es/ANALISIS-BYTE-TI/ORACLE-DATABASE-12C/)

En esta arquitectura se observa la base y la lógica de datos dentro de un mismo servidor, de manera que las aplicaciones, que los consultan, sólo tienen que viajar a un servidor unificado; cuando por lo general en las arquitecturas tradicionales, con cualquier otro lenguaje de programación como puede ser Java, .Net, etc. el servidor que posee el lenguaje está en otro distinto al de la base de datos y esto retarda un poco los tiempos de respuesta hacia el usuario final.

La arquitectura tradicional puede verse a continuación

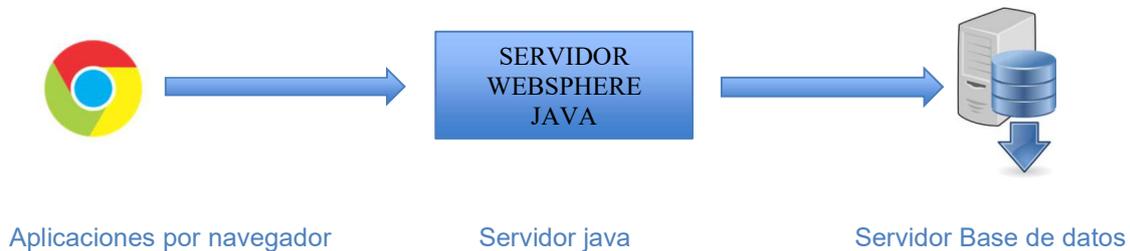


ILUSTRACIÓN 8 – MODELO ARQUITECTURA TRADICIONAL

FUENTE: AUTORÍA PROPIA BASADA EN CONOCIMIENTOS ADQUIRIDOS

Batch Shell:

Otra de las tecnologías empleadas en el proyecto es el Batch Shell de Unix. Es un intérprete de comandos que consiste en una interfaz de usuario tradicional de los sistemas operativos basados en Unix, el cual ejecuta órdenes secuenciales. Las extensiones de estos archivos pueden ser: ash, csh, zsh, tcsh, etc.; pero para este proyecto únicamente se aplicarán las extensiones .sh y .ksh, que son aquellas que se encuentran dentro de la jerarquía de archivos en Unix en “/bin/sh”.

Esta tecnología se utiliza para automatizar el proceso de Canje Postcompra, es decir, para que todos los pasos lógicos escritos en el PL/SQL corran de manera automática a cierta hora programada y en ciertas condiciones dadas.

Estos archivos Batch .sh van a llamar a los .sql donde se encuentra la lógica de negocio y los van a ejecutar con distintos parámetros según sea necesario

La herramienta en donde corren estos archivos Batches es el **Control-M Workload Automation**

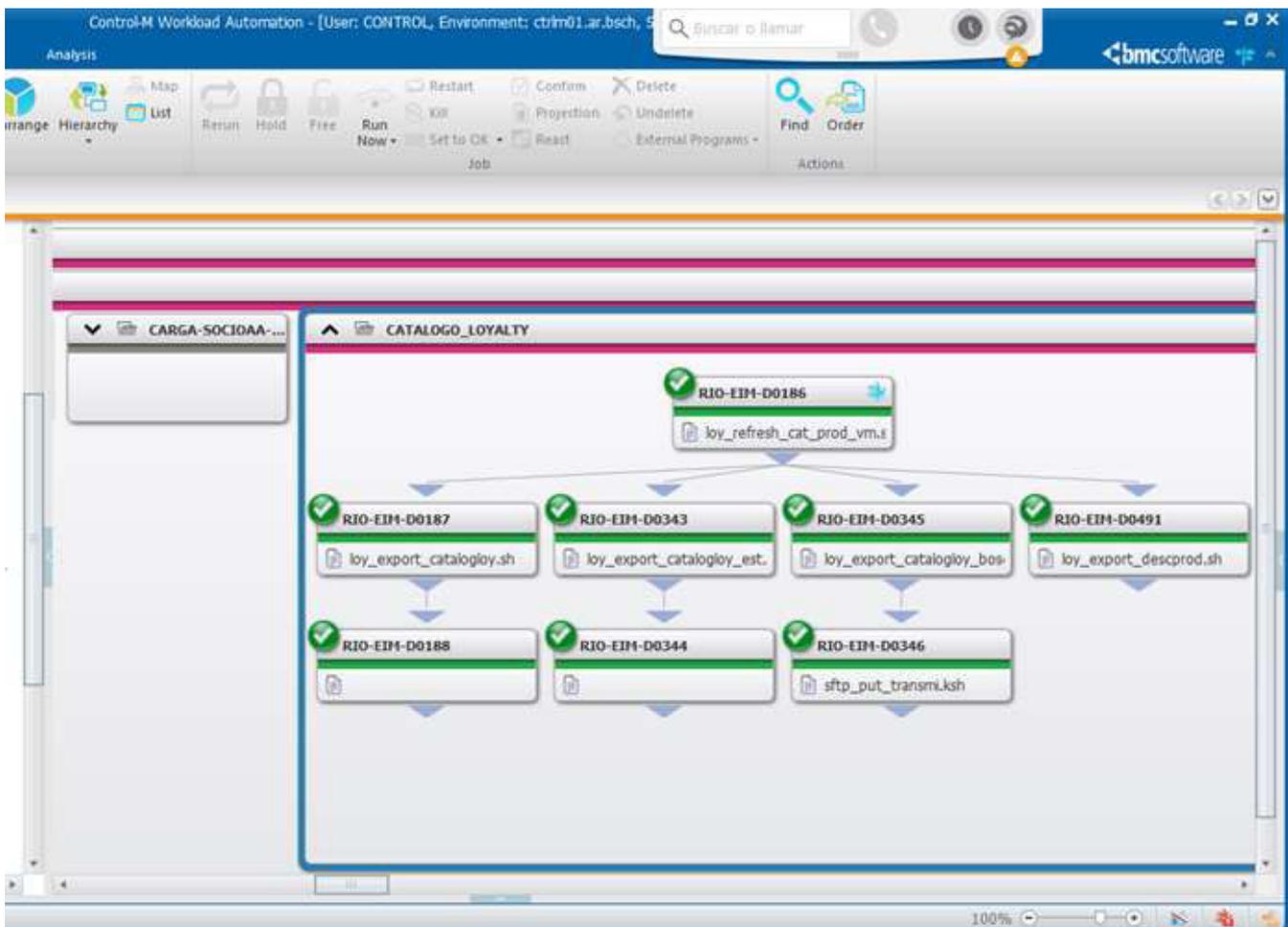


ILUSTRACIÓN 9 – CONTROL-M WORKLOAD AUTOMATION

FUENTE: CAPTURA DE PANTALLA DE LA HERRAMIENTA USADA EN EL TRABAJO

Este software es un programador/scheduler y automatizador de flujo de trabajo, proporcionando un único punto de control donde monitorear toda la carga de tareas.

Como se observa en la imagen (que es un ejemplo de la cadena “bajada de catálogos de premios de Superclub”), cada una de esas cajas son los Jobs/tareas .sh o .ksh que ejecutan los determinados .sql y los cuales tienen antecesores y predecesores con condiciones, por lo que hasta que no corran ciertos Jobs, no van a poder hacerlo los que les siguen. Cabe aclarar que dicha herramienta proporciona varios medidores para monitoreo de las tareas, como lo son estadísticas de tiempos de ejecución, logs de las ejecuciones, colores que indiquen al estado de ejecución (verde para correcto y rojo para incorrecto), entre muchas otras.

Java:



ILUSTRACIÓN 10 - LOGO JAVA

FUENTE: [HTTPS://SITES.GOOGLE.COM/SITE/JAVAYOS/PRINCIPAL/PROGRAMACION/JAVADESDE0](https://sites.google.com/site/javayos/principal/programacion/javadesde0)

Java es un lenguaje de programación compilado que ejecuta desde el lado del servidor. El mismo contiene una máquina virtual (JVM) encargada de la ejecución de las aplicaciones, es decir, un código fuente compilado y un conjunto de bibliotecas estándar que ofrecen una funcionalidad común, pero que también pueden ampliarse descargando bibliotecas nuevas, este se denomina Java Runtime Environment (JRE).

Lo que hace que java sea un lenguaje multiplataforma, permitiendo que una misma aplicación pueda ser ejecutada en una gran variedad de sistemas con arquitecturas distintas es esta máquina virtual (JVM) instalada.

La infraestructura de servidor donde están montados los programas java para el proyecto es **WebSphere Application Server (WAS)**



ILUSTRACIÓN 11 – LOGO SERVIDOR WEBSHERE

FUENTE: [HTTPS://BLOG.MOON.CAT/INSTALAR-WEBSHERE-APPLICATION-SERVER-9-0-EN-OPENSUSE-LEAP](https://blog.moon.cat/instalar-websphere-application-server-9-0-en-opensuse-leap)

Este servidor de aplicaciones de software de la familia de IBM usa estándares abiertos y gratuitos como J2EE, XML, entre otros. Es la base de la infraestructura que soporta entornos nuevos que son SOA (orientados a servicios) y aquellos un poco más anticuados que no lo son. Su principal ventaja es su característica de ser multiplataforma ya que funciona en distintos sistemas operativos como AIX, UNIX, LINUX, SUN SOLARIS, WINDOWS, etc.

Se optó por esta tecnología debido a que la arquitectura antes mencionada ya está implementada en el área de Superclub en otros programas o módulos y funciona de manera robusta, es decir, con una tasa prácticamente despreciable de fallos. En esta ocasión, se usa java para el proyecto para hacer las validaciones finales en tiempo real de ciertos puntos como:

- Criterios con los que tiene que cumplir un cliente para ser premiado, que pueden ser puntos, inactividad y demás.
- Problemas con los comodines de ahorro a canjear como stock insuficiente
- Reportar algún tipo de inactividad del servidor, ya sea que esté caído o inaccesible.
- Reportar demoras excesivas en la respuesta del servidor por sobrecarga.

2.2 Metodología de desarrollo:

La metodología que se escogió para el proyecto es el modelo de desarrollo en cascada, el cual es un proceso de tipo secuencial en donde el software se concibe como un conjunto de etapas que se ejecutan una tras otra y en donde obligatoriamente, no puede comenzar una sin haber culminado la anterior. Esto se denota lógicamente en el gráfico que se presenta a continuación.

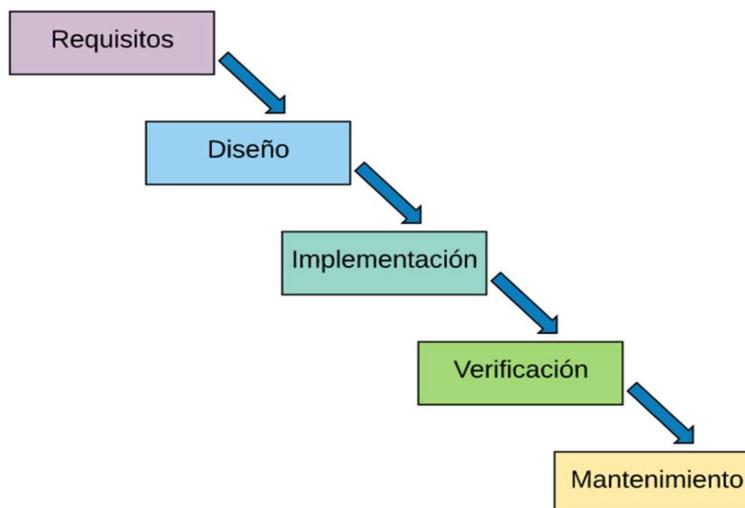


ILUSTRACIÓN 12 – ESQUEMA METODOLOGIA EN CASCADA

FUENTE: [HTTPS://OPENCLASSROOMS.COM/EN/COURSES/4309151-GESTIONA-TU-PROYECTO-DE-DESARROLLO/4538221-EN-QUE-CONSISTE-EL-MODELO-EN-CASCADA](https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada)

Funciona de la siguiente manera: del lado del usuario o cliente se solicita el desarrollo, en este caso, el área de Marketing del Banco Santander, redacta la lista de requisitos, criterios y características que debe cumplir el software para ser funcional y tener éxito en su alcance. Luego el mismo ya queda en manos de los programadores, implementadores y demás, que irán pasando por todas las etapas que muestra el gráfico en cuestión y sin necesidad de que del lado del cliente tenga que volver a participar.

Se escogió esta metodología porque el desarrollo en cuestión no es algo que se pueda iterar e ir mostrando resultados parciales (es muy común cuando se utiliza alguna otra) ya que las pantallas finales son sólo dos y allí debe mostrarse la lógica ya terminada.

Como toda metodología de trabajo presenta ventajas y desventajas.

Ventajas:

- Se sientan las bases del desarrollo en el paso 1 dejando todo documentado y registrado, por lo tanto, si se une un desarrollador en el medio del progreso del proyecto, tiene todo encaminado y exployado para comprenderlo.
- Acorta tiempos de definición de objetivos y requerimientos, ya que se hacen una sola vez.
- Favorece la medición del progreso del proyecto, ya que el alcance completo se conoce de manera anticipada.

Desventajas:

- No hay interacción de la parte cliente/usuario mientras se está desarrollando, así no se puede plantear un cambio de alcance o requisitos sino hasta que se termine todo el proyecto.

- No pueden paralelizarse etapas, por lo tanto si una queda estancada por alguna complicación técnica, retrasará el comienzo de la siguiente y además, los tiempos de entrega del proyecto
- No es recomendable para proyectos muy largos y que requieran de entregas intermedias.
- No hay oportunidad de detectar fallos mientras se está desarrollando y poder corregirlos en el momento. Entonces, luego de que se termine el desarrollo, habrá que repararlos todos juntos.
- Muchas veces la parte usuaria o cliente no sabe bien los requisitos de entrada ya que estos varían por decisiones de negocio. Así que cambiarlos luego de iniciado el desarrollo, implicaría volver a realizar fases que ya concluyeron, y así, aumentar el tiempo y costos

2.3 Ejecución del cronograma de tareas:

El cronograma de tareas fue dividido en diferentes etapas:

1. Relevamiento de requisitos
2. Desarrollo
3. Test y validación
4. Implementación.

A continuación, se describirán detalladamente cada una de las mismas.

2.3.1 Relevamiento de requisitos:

Dentro del glosario de Terminología Estándar de Ingeniería de Software (IEEE: Standard Glossary of Software Engineering Terminology) se define “requisito”

como: condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Se deben respetar estas tres características comunes:

- Ser conciso, es decir, claro y fácil de comprender.
- Ser consistente, no debe entrar en confusión con cualquier otro requisito.
- Ser completo, debe proporcionar la información necesaria para su comprensión y evaluación.

A su vez, como los requisitos son las necesidades que se deben desarrollar en cualquier proyecto de software, es indispensable poder medirlos o comprobarlos. Por ello es clave que se documenten para, luego de finalizados, poder verificar si se cumplieron correctamente o no, sin ambigüedades ni refutaciones.

Como la metodología de trabajo que se usa en este proyecto es la de *cascada*, se llevaron a cabo una serie de reuniones para determinar las necesidades por parte de los interesados en el desarrollo: el cliente interno Marketing y Legales.

Luego de realizada la lista pertinente, Sistemas procede a realizar el diseño/maquetado de pasos lógicos que debería llevar a cabo para poder abarcar a todo lo pedido. Ahí es en donde se decanta si técnicamente hay algún requisito muy complejo de implementar y se realiza una retroalimentación hacia la parte interesada para consensuar cambiarlo.

Finalizados los encuentros, si ya no hay ninguna objeción o retroalimentación de Sistemas y todas las partes están de acuerdo en una definición común de los requerimientos, se procede a dar por terminada esta etapa. Luego se pasa al desarrollo, sabiendo que como marca la metodología de trabajo elegida, no se puede volver para redefinir algún alcance de requisito hasta que no se haya terminado y presentado el sistema final.

En dichas reuniones Marketing comunica a Sistemas las diferentes necesidades y características primordiales que debe tener el nuevo Sistema:

- Identificar universo de consumos del día con tarjetas y en determinados rubros
- Quedarse con los máximos consumos por persona y rubro
- Premiar con comodines de descuento en dichos rubros
- Excluir de la premiación a los consumos que ya hayan canjeado un comodín en el mes.
- Ofrecer siempre el mayor comodín posible como alcancen los puntos del cliente
- Calcular el reintegro en efectivo que daría el comodín
- Calcular el tope de reintegro
- Ofrecer un token único de premio con caducidad en 24hs si no se canjea
- Avisar del premio con un mail diario al cliente
- Automatizar la lógica de proceso para que corra todas las mañanas
- Ofrecer una interfaz gráfica de premiación dinámica y simplificada
- Centralizar los datos de consumo y premiación de manera confiable e integra en la base de datos

Realizado el análisis y diseño de pasos lógicos, se opta por elegir trabajar con las bases de datos Oracle RIO53 y RIO52, en las cuales se alberga toda la información necesaria para poder cumplir con lo pedido, siendo viable casi todos los requerimientos del cliente interno. Finalmente, como retroalimentación, se sugiere un cambio de alcance en el requisito “excluir de la premiación a los consumos que ya hayan canjeado un comodín en el mes”, debido a que los canjes efectuados del mes en curso los informa Prisma a mes vencido y no se posee dicha información hasta dentro de 30 días. Por lo cual se decide limitarlo a “excluir de la premiación a los consumos que ya hayan canjeado un comodín en el día”. La parte interesada acepta el cambio y ya se da por finalizada la etapa de relevamiento.

2.3.2 Desarrollo:

- **Lógica PL/SQL movimientos a premiar:**

Primeramente se extraen los movimientos del día en tarjetas de crédito Visa, Amex y las de débito para el universo de empleados acotados y para los rubros antes mencionados. Una vez recolectados, se guardan todos en la interfaz 'ofsa40.mov_presentados'.

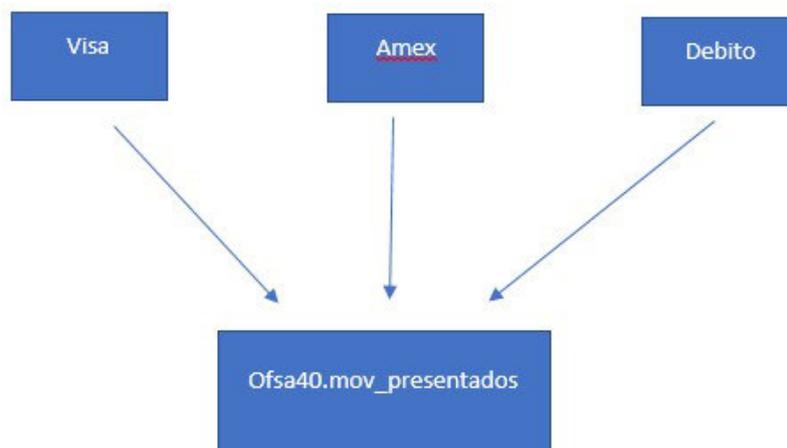


ILUSTRACIÓN 13 – ESQUEMA REFERENCIAL DE EXTRACCIÓN DE MOVIMIENTOS/CONSUMOS

FUENTE: EXPERIENCIA PROPIA DE LO REALIZADO EN EL TRABAJO

```

mov_presentados_paso1.sql
1 declare
2 limit_in number:=1024;
3 vFecha VARCHAR2(10):='&1';
4
5
6     CURSOR movimientos_cur
7     IS
8     SELECT distinct a.FPRES fecha,
9         a.TARJETA,
10        a.importe_1 importe,
11        a.rubro,
12        a.denest Local,
13        a.nup_titu Nup,
14        a.codop,
15        a.usuario,
16        'AMEX' as v_mdp
17 FROM ofsa40.TMOVPRES_amex a
18 WHERE
19     a.RUBRO IN (5172 --combustible
20                ,5813 -- desayuno
21                ,4225 -- indumentaria
22                ,7230 -- peluquerías
23                ,5812 -- restaurante
24                ,5411 --supermercado
25        )
26     AND a.NUP_TITU in ('10559540'
27                       , '00567048'
28                       , '80259640'
29                       , '02515316'
30                       , '80113458'
31                       , '05998215'
32                       , '80203949'
33                       , '05999528'
34                       , '10094123'
35                       , '05818034')
36     AND LPAD(CODOP, 6,'0') LIKE '0005%'
37     AND a.fpres = TO_DATE (vFecha, 'dd/mm/yyyy') --dia del consumo
38 UNION
39 SELECT a.FPRES fecha,
40        a.TARJETA,
41        importe_1 importe,
42        rubro,

```

ILUSTRACIÓN 14 – FRAGMENTO SCRIPT PLSQL EXTRACCIÓN DE MOVIMIENTOS

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

En la resolución del script PL/SQL, se crea un cursor* llamado movimientos_cur, quien contendrá todos los datos arrojados por la consulta y actuara a su vez como arreglo de datos.

La consulta no es más ni menos que un select* a las tres tablas base ofsa40.TMOVPRES_AMEX, ofsa40.TMOVPRES_VISA y ofsa40.TRANSACCIONES_ATM (débito), en donde se traen los campos que

aportan mayor información del consumo como el número de tarjeta, importe, rubro del consumo, nup del cliente (número único e inequívoco que identifica a un cliente dentro de la entidad bancaria), etc. Como se puede ver en la imagen está recortada ya que el script es muy largo, estos datos se restringen dentro de la cláusula where* por el universo limitado de empleados, los rubros específicos y la fecha del día que se pasa como parámetro vFecha.

Estos select* van por separado y se agrupan con la sentencia UNION*.

```
102     TYPE movimientos_aat IS TABLE OF movimientos_cur%ROWTYPE
103     INDEX BY PLS_INTEGER;
104
105     l_movimientos movimientos_aat;
106 BEGIN
107
108
109     OFSA40.PKG_TABLE_TOOLS.TRUNC_TABLE('ofsa40.mov_presentados');
110
111     commit;
112
113     OPEN movimientos_cur;
114     LOOP
115         FETCH movimientos_cur
116         BULK COLLECT INTO l_movimientos LIMIT limit_in;
117
118         FOR indx IN 1 .. l_movimientos.COUNT
119         LOOP
120             begin
121                 insert into ofsa40.mov_presentados values l_movimientos(indx);
122             exception
123             when others then
124                 null;
125             end;
126         END LOOP;
127         commit;
128         EXIT WHEN l_movimientos.COUNT < limit_in;
129
130     END LOOP;
131
132     CLOSE movimientos_cur;
133 END ;
134
135 /
```

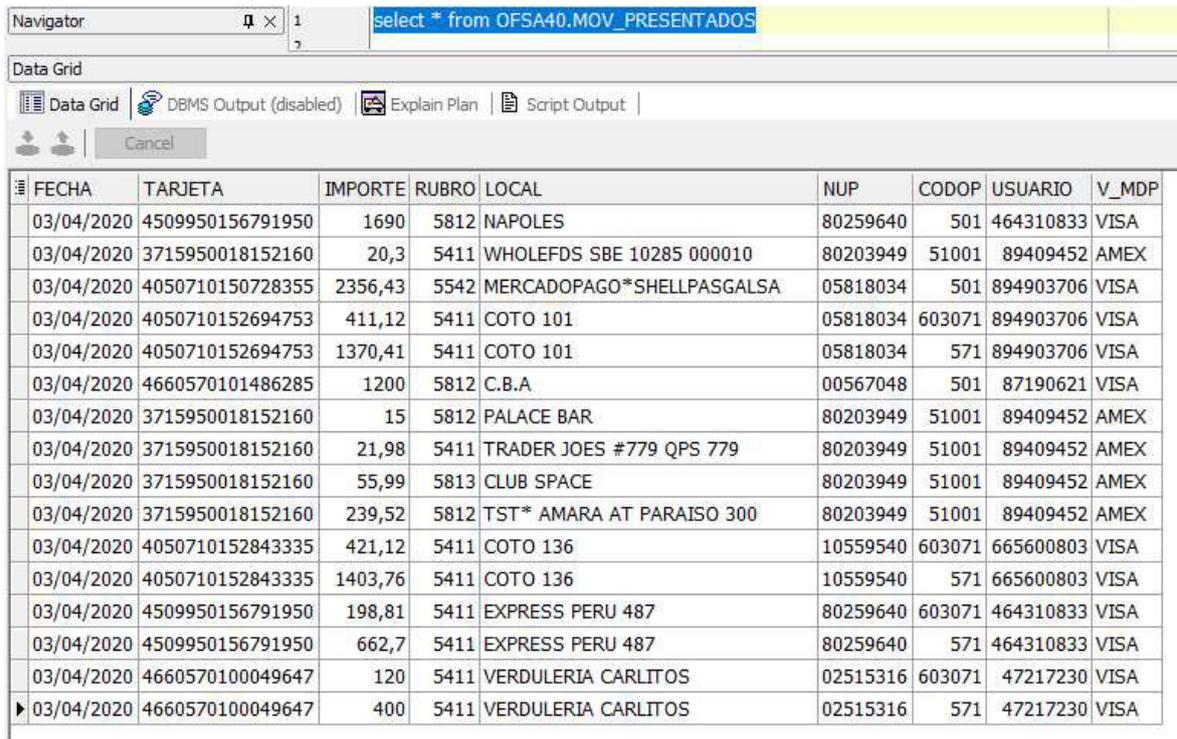
ILUSTRACIÓN 15 – FRAGMENTO SCRIPT PLSQL CARGA DE INTERFAZ

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Luego de terminada la consulta, como se aprecia en la imagen anterior, se limpia la tabla o interfaz donde se va a colocar la información nueva mediante la sentencia truncate*. Es allí donde se procede a abrir el cursor* que contiene los datos, movimientos_cur, para que a través de un bucle volcar los mismos dentro

de la interfaz 'ofsa40.MOV_PRESENTADOS', hasta que el cursor se quede sin datos.

Al concluir la carga, la interfaz quedara de la siguiente manera.



The screenshot shows a database interface with a SQL query window and a data grid. The query is 'select * from OFSA40.MOV_PRESENTADOS'. The data grid contains the following data:

FECHA	TARJETA	IMPORTE	RUBRO	LOCAL	NUP	CODOP	USUARIO	V_MDP
03/04/2020	4509950156791950	1690	5812	NAPOLES	80259640	501	464310833	VISA
03/04/2020	3715950018152160	20,3	5411	WHOLEFDS SBE 10285 000010	80203949	51001	89409452	AMEX
03/04/2020	4050710150728355	2356,43	5542	MERCADOPAGO*SHELLPASGALSA	05818034	501	894903706	VISA
03/04/2020	4050710152694753	411,12	5411	COTO 101	05818034	603071	894903706	VISA
03/04/2020	4050710152694753	1370,41	5411	COTO 101	05818034	571	894903706	VISA
03/04/2020	4660570101486285	1200	5812	C.B.A	00567048	501	87190621	VISA
03/04/2020	3715950018152160	15	5812	PALACE BAR	80203949	51001	89409452	AMEX
03/04/2020	3715950018152160	21,98	5411	TRADER JOES #779 QPS 779	80203949	51001	89409452	AMEX
03/04/2020	3715950018152160	55,99	5813	CLUB SPACE	80203949	51001	89409452	AMEX
03/04/2020	3715950018152160	239,52	5812	TST* AMARA AT PARAISO 300	80203949	51001	89409452	AMEX
03/04/2020	4050710152843335	421,12	5411	COTO 136	10559540	603071	665600803	VISA
03/04/2020	4050710152843335	1403,76	5411	COTO 136	10559540	571	665600803	VISA
03/04/2020	4509950156791950	198,81	5411	EXPRESS PERU 487	80259640	603071	464310833	VISA
03/04/2020	4509950156791950	662,7	5411	EXPRESS PERU 487	80259640	571	464310833	VISA
03/04/2020	4660570100049647	120	5411	VERDULERIA CARLITOS	02515316	603071	47217230	VISA
03/04/2020	4660570100049647	400	5411	VERDULERIA CARLITOS	02515316	571	47217230	VISA

ILUSTRACIÓN 16 – INTERFAZ MOVIMIENTOS PRESENTADOS CARGADA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como siguiente paso, se parte de la interfaz de transacciones anterior ya cargada, ofsa40.MOV_PRESENTADOS y se filtra para que sólo queden los máximos consumos del día agrupados por nup y por rubro, es decir, que si una persona tenía en el día dos consumos distintos en el mismo rubro, se descarta el de menor monto.

Por ej: rubro Combustibles. Si hay dos movimientos por cargar nafta en el día, la lógica se queda con el del monto máximo.

Esto hará que el escenario o universo de consumos que se tenía se reduzca aún más ya que no quedarán consumos con rubros duplicados por persona y se cargarán en una nueva interfaz ofsa40.MOV_PRESENTADOS_MAX.



ILUSTRACIÓN 17 – ESQUEMA REFERENCIAL DE CONSUMOS MÁXIMOS

FUENTE: EXPERIENCIA PROPIA DE LO REALIZADO EN EL TRABAJO

```

mov_presentados_max_paso2.sql x
1 declare
2 limit_in number:=1024;
3
4
5     CURSOR max_cur
6     IS
7     select a.importe, a.nup, a.rubro, b.local, b.fecha, b.tarjeta, b.codop, b.usuario, B.V_MDP from
8     (select max(importe) as importe , nup, rubro from OFSA40.MOV_PRESENTADOS
9     group by nup, rubro) a,
10    OFSA40.MOV_PRESENTADOS b
11    where a.nup = b.nup
12    and a.rubro = b.rubro
13    and a.importe = b.importe;
14
15    TYPE max_aat IS TABLE OF max_cur%ROWTYPE
16    INDEX BY PLS_INTEGER;
17
18    l_max max_aat;
19 BEGIN
20
21
22    OFSA40.PKG_TABLE_TOOLS.TRUNC_TABLE('ofsa40.mov_presentados_max');
23
24    commit;
25
26    OPEN max_cur;
27    LOOP
28        FETCH max_cur
29        BULK COLLECT INTO l_max LIMIT limit_in;
30
31        FOR indx IN 1 .. l_max.COUNT
32        LOOP
33            begin
34                insert into ofsa40.mov_presentados_max(IMPORTE,NUP,RUBRO,LOCAL,FECHA,TARJETA,CODOP,USUARIO,V_MDP)
35                values (l_max(indx).IMPORTE,
36                    l_max(indx).NUP,
37                    l_max(indx).RUBRO,
38                    l_max(indx).LOCAL,
39                    l_max(indx).FECHA,
40                    l_max(indx).TARJETA,
41                    l_max(indx).CODOP,
42                    l_max(indx).USUARIO,
43                    l_max(indx).V_MDP);
44            exception
45            when others then

```

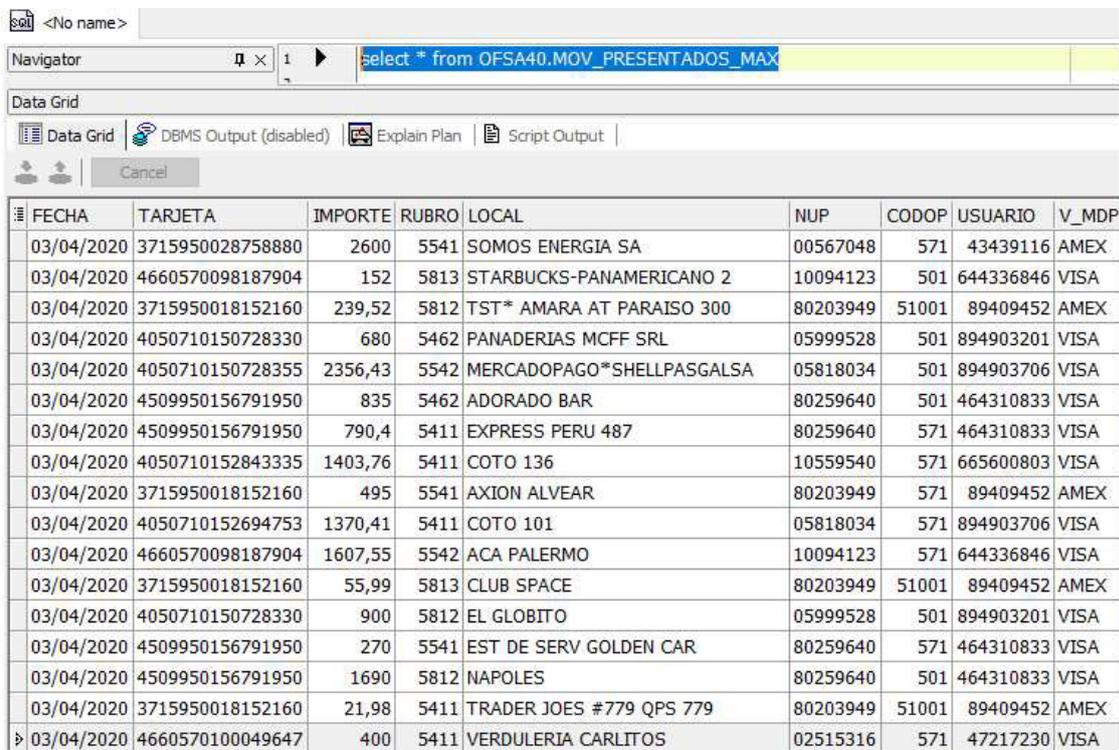
ILUSTRACIÓN 18 – FRAGMENTO SCRIPT PLSQL CONSUMOS MÁXIMOS X RUBRO Y X PERSONA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

En la resolución del script se crea el cursor max_cur, quien almacenará los datos de la consulta y se lo utilizará como arreglo de datos. Como se percibe en la imagen, a través del uso de la función MAX()* se selecciona el consumo de máximo importe. Este se agrupa por nup y rubro con la sentencia GROUP BY*. Luego se realiza una subconsulta con la misma tabla para ir a buscar todos los otros datos que faltan.

Una vez que el cursor está completo con la información, se limpia la interfaz nueva donde se va a depositar a través de la sentencia truncate*. Allí se procede a abrir el cursor* movimientos_cur, para insertar los datos a través de un bucle, dentro de la interfaz 'ofsa40.MOV_PRESENTADOS_MAX', hasta que el cursor se quede sin datos.

Al concluir la carga, la interfaz quedara de la siguiente manera.



The screenshot shows a database interface with a SQL query window containing the command: `select * from OFSA40.MOV_PRESENTADOS_MAX`. Below the query window is a 'Data Grid' displaying a list of transactions. The grid has columns for FECHA, TARJETA, IMPORTE, RUBRO, LOCAL, NUP, CODOP, USUARIO, and V_MDP.

FECHA	TARJETA	IMPORTE	RUBRO	LOCAL	NUP	CODOP	USUARIO	V_MDP
03/04/2020	3715950028758880	2600	5541	SOMOS ENERGIA SA	00567048	571	43439116	AMEX
03/04/2020	4660570098187904	152	5813	STARBUCKS-PANAMERICANO 2	10094123	501	644336846	VISA
03/04/2020	3715950018152160	239,52	5812	TST* AMARA AT PARAISO 300	80203949	51001	89409452	AMEX
03/04/2020	4050710150728330	680	5462	PANADERIAS MCFF SRL	05999528	501	894903201	VISA
03/04/2020	4050710150728355	2356,43	5542	MERCADOPAGO*SHELLPASGALSA	05818034	501	894903706	VISA
03/04/2020	4509950156791950	835	5462	ADORADO BAR	80259640	501	464310833	VISA
03/04/2020	4509950156791950	790,4	5411	EXPRESS PERU 487	80259640	571	464310833	VISA
03/04/2020	4050710152843335	1403,76	5411	COTO 136	10559540	571	665600803	VISA
03/04/2020	3715950018152160	495	5541	AXION ALVEAR	80203949	571	89409452	AMEX
03/04/2020	4050710152694753	1370,41	5411	COTO 101	05818034	571	894903706	VISA
03/04/2020	4660570098187904	1607,55	5542	ACA PALERMO	10094123	571	644336846	VISA
03/04/2020	3715950018152160	55,99	5813	CLUB SPACE	80203949	51001	89409452	AMEX
03/04/2020	4050710150728330	900	5812	EL GLOBITO	05999528	501	894903201	VISA
03/04/2020	4509950156791950	270	5541	EST DE SERV GOLDEN CAR	80259640	571	464310833	VISA
03/04/2020	4509950156791950	1690	5812	NAPOLES	80259640	501	464310833	VISA
03/04/2020	3715950018152160	21,98	5411	TRADER JOES #779 QPS 779	80203949	51001	89409452	AMEX
03/04/2020	4660570100049647	400	5411	VERDULERIA CARLITOS	02515316	571	47217230	VISA

ILUSTRACIÓN 19 – INTERFAZ MOVIMIENTOS PRESENTADOS MÁXIMOS CARGADA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como se comprueba en la imagen, el último consumo del comercio 'Verdulería Carlitos' sólo aparece una vez con el monto máximo de \$400, descartando el importe menor, mientras que en la primer interfaz aparecía dos veces con ambos montos.

El siguiente paso es capturar en una nueva tabla los comodines de descuento canjeados del día por homebanking, ya que se necesitan para evitar y restringir que a las personas se les ofrezca como premio el comodín de algún rubro ya canjeado el mismo día.

```

canjes_loy_comd_paso3.sql x
2 limit_in number:=1024;
3 vFecha VARCHAR2(10):='&1';
4
5
6 CURSOR canjes_cur
7 IS
8 select t.nup , t.nom_y_ape, t.cod_articulo,c.cod_rubro,C.DESC_RUBRO, t.id_canje, t.fecha_mov from ofsa40.loyal_TRANSAX t, ofsa40.loyal_catalogo_rubro c
9 where t.nup in ('10559540'
10 , '00567048'
11 , '80259640'
12 , '02515316'
13 , '80113458'
14 , '05998215'
15 , '80203949'
16 , '05999528'
17 , '10094123'
18 , '05818034')
19 and t.cod_articulo = c.cod_articulo
20 and fecha_mov = to_date(vFecha,'dd/mm/yyyy');
21
22 TYPE canjes_aat IS TABLE OF canjes_cur%ROWTYPE
23 INDEX BY PLS_INTEGER;
24
25 l_canjes canjes_aat;
26 BEGIN
27
28 OFSA40.PKG_TABLE_TOOLS.TRUNC_TABLE('ofsa40.canjes_comodin');
29
30 commit;
31
32 OPEN canjes_cur;
33 LOOP
34 FETCH canjes_cur
35 BULK COLLECT INTO l_canjes LIMIT limit_in;
36
37 FOR indx IN 1 .. l_canjes.COUNT
38 LOOP
39 begin
40 insert into ofsa40.canjes_comodin(NUP,NOM_Y_APE,COD_ARTICULO,COD_RUBRO,DESC_RUBRO,ID_CANJE,FECHA_MOV)
41 values (l_canjes(indx).NUP,
42 l_canjes(indx).NOM_Y_APE,
43 l_canjes(indx).COD_ARTICULO,
44 l_canjes(indx).COD_RUBRO,
45

```

ILUSTRACIÓN 20 – FRAGMENTO SCRIPT PLSQL CANJES DIARIOS EN RUBROS

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

El script es bastante similar a los demás, donde se utiliza el cursor canjes_cur para ir buscar los datos de los canjes por homebanking, los cuales se almacenan en la tabla base ofsa40.LOYAL_TRANSAX. Luego se hace una intersección con otra tabla ofsa40.LOYAL_CATALOGO_RUBRO para limitar los canjes solamente a los rubros a los que queremos premiar, ya que la primer tabla contiene los de todos los rubros totales y existentes en el banco. Esto va aplicado exclusivamente para el universo de personas para el que se lanzaba primeramente la aplicación dentro de la cláusula where* y para la fecha del día referenciada como vFecha.

Después, como en todas las cargas, se limpia la nueva interfaz mediante la sentencia truncate* y se introduce el cursor en un bucle para insertar los datos en 'ofsa40.CANJES_COMODIN', se muestra a continuación.



The screenshot shows a database interface with a SQL query editor and a Data Grid. The query is `select * from OFSA40.CANJES_COMODIN`. The Data Grid displays the following data:

NUP	NOM_Y_APE	COD_ARTICULO	COD_RUBRO	DESC_RUBRO	ID_CANJE	FECHA_MOV
00567048	JUAN ANDRES CLEMENTE	CRT011	5411	SUPERMERCADOS	1-9157011602	03/04/2020
05999528	NICOLAS SEBASTIAN FRINO	CRT005	5462	DESAYUNOS	1-0133011882	03/04/2020
10094123	JESICA CECILIA CONTINO	CRT006	5813	DESAYUNOS	1-1143011490	03/04/2020
10094123	JESICA CECILIA CONTINO	CRT015	5137	INDUMENTARIA	1-9153911976	03/04/2020

ILUSTRACIÓN 21 – INTERFAZ CANJES COMODÍN CARGADA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Finalmente queda restar o descartar de los movimientos máximos a premiar almacenados en 'ofsa40.MOV_PRESENTADOS_MAX' a las personas que ya hayan canjeado en esos rubros en el día almacenadas en 'ofsa40.CANJES_COMODIN'.

De manera que se vuelve a achicar el universo de transacciones a premiar por última vez, dejando ya definitivamente cargada la interfaz 'ofsa40.MOV_A_PREMIAR'

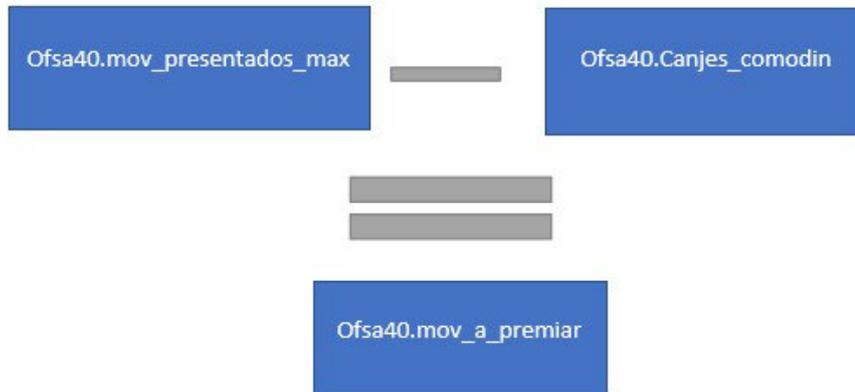


ILUSTRACIÓN 22 – ESQUEMA REFERENCIAL DE MOVIMIENTOS A PREMIAR

FUENTE: EXPERIENCIA PROPIA DE LO REALIZADO EN EL TRABAJO

```

mov_a_premiar_paso4.sql x
1 declare
2 limit_in number:=1024;
3
4
5
6   CURSOR apremiar_cur
7   IS
8     select t.nup, t.rubro ,t.codop, t.fecha, t.importe, t.local, t.tarjeta, t.usuario, t.v_mdp, S.SALDO_PUNTOS from
9     (select a.nup as nup, a.rubro as rubro ,B.CODOP as codop ,B.FECHA as fecha ,B.IMPORTE as importe ,B.LOCAL as local ,B.TARJETA as tarjeta
10    (select nup, rubro from OFSA40.MOV_PRESENTADOS_MAX
11    minus
12    select nup, cod_rubro from OFSA40.CANJES_COMODIN) a,
13    OFSA40.MOV_PRESENTADOS_MAX b
14    where a.nup = b.nup
15    and a.rubro = B.RUBRO) t , ofsa40.LOYAL_SALDOS s
16    where T.NUP = S.NUP
17    and S.PGM_SCH = 'SUPERCLUB';
18
19   TYPE apremiar_aat IS TABLE OF apremiar_cur%ROWTYPE
20   INDEX BY PLS_INTEGER;
21
22   l_apremiar apremiar_aat;
23 BEGIN
24
25   OFSA40.PKG_TABLE_TOOLS.TRUNC_TABLE('OFS40.MOV_A_PREMIAR');
26
27   commit;
  
```

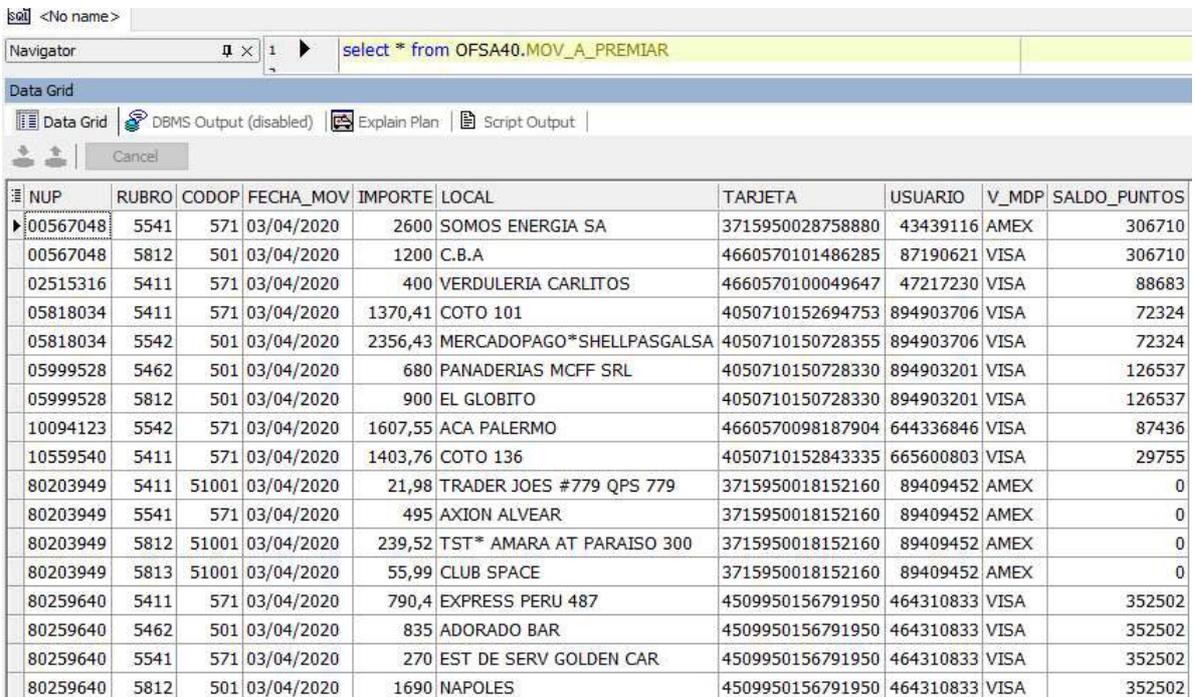
ILUSTRACIÓN 23 – FRAGMENTO SCRIPT PLSQL RESTA DE MOV CON CANJES

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Para la resta, se invoca una función que provee el motor PL/SQL llamada minus*. Esta realiza una diferencia entre dos universos, posicionando en la parte de arriba al universo minuendo y en la parte de abajo al universo sustraendo, obteniendo como resultado la resta.

Como se observa en la imagen, se realiza una subconsulta para ir a buscar el saldo de puntos del cliente que va a ser premiado, alojado en la tabla ofsa40.LOYAL_SALDOS. Este servirá más adelante para poder calcular -en base a la cantidad de puntos que se posea- cuál es el “premio comodín” de descuento a canjear que se le va a ofrecer, ya que estos tienen un costo en puntos.

La carga de la interfaz se realiza como en todos los pasos, abriendo el cursor donde se alojan los datos de la consulta ‘apremiar_cur’ y vuelcan mediante un bucle loop* a la interfaz, previo vaciado de la misma para tener certeza que no queden datos viejos acumulados. A continuación se ilustra como quedo cargada ofsa40.MOV_A_PREMIAR.



NUP	RUBRO	CODOP	FECHA_MOV	IMPORTE	LOCAL	TARJETA	USUARIO	V_MDP	SALDO_PUNTOS
00567048	5541	571	03/04/2020	2600	SOMOS ENERGIA SA	3715950028758880	43439116	AMEX	306710
00567048	5812	501	03/04/2020	1200	C.B.A	4660570101486285	87190621	VISA	306710
02515316	5411	571	03/04/2020	400	VERDULERIA CARLITOS	4660570100049647	47217230	VISA	88683
05818034	5411	571	03/04/2020	1370,41	COTO 101	4050710152694753	894903706	VISA	72324
05818034	5542	501	03/04/2020	2356,43	MERCADOPAGO*SHELLPASGALSA	4050710150728355	894903706	VISA	72324
05999528	5462	501	03/04/2020	680	PANADERIAS MCFF SRL	4050710150728330	894903201	VISA	126537
05999528	5812	501	03/04/2020	900	EL GLOBITO	4050710150728330	894903201	VISA	126537
10094123	5542	571	03/04/2020	1607,55	ACA PALERMO	4660570098187904	644336846	VISA	87436
10559540	5411	571	03/04/2020	1403,76	COTO 136	4050710152843335	665600803	VISA	29755
80203949	5411	51001	03/04/2020	21,98	TRADER JOES #779 QPS 779	3715950018152160	89409452	AMEX	0
80203949	5541	571	03/04/2020	495	AXION ALVEAR	3715950018152160	89409452	AMEX	0
80203949	5812	51001	03/04/2020	239,52	TST* AMARA AT PARAISO 300	3715950018152160	89409452	AMEX	0
80203949	5813	51001	03/04/2020	55,99	CLUB SPACE	3715950018152160	89409452	AMEX	0
80259640	5411	571	03/04/2020	790,4	EXPRESS PERU 487	4509950156791950	464310833	VISA	352502
80259640	5462	501	03/04/2020	835	ADORADO BAR	4509950156791950	464310833	VISA	352502
80259640	5541	571	03/04/2020	270	EST DE SERV GOLDEN CAR	4509950156791950	464310833	VISA	352502
80259640	5812	501	03/04/2020	1690	NAPOLES	4509950156791950	464310833	VISA	352502

ILUSTRACIÓN 24 – INTERFAZ MOVIMIENTOS A PREMIAR CARGADA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como se puede observar, uno de los movimientos que se excluyó de la premiación fue “Starbucks panamericano 2” (se puede ver en la ilustración 19) del rubro 5813 (Desayunos), ya que la persona que lo realizó tenía canjes en ese rubro para ese mismo día (esto se advierte en la ilustración 21).

- **Lógica PL/SQL asignación de premios:**

Primeramente se parte desde ‘ofsa40.MOV_A_PREMIAR’, la interfaz final que había quedado desde el paso anterior, que contiene las transacciones dignas de otorgamiento de comodín de descuento.

Ahora se tratará la asignación de qué comodín por rubro se le ofrece a cada movimiento, ya que cada uno hay varios como por ejemplo: combustibles posee tres de 10%, 20% y 30% de descuento, siendo el mayor el más caro en puntaje para el canje.

En este paso se va a eliminar de ‘ofsa40.MOV_A_PREMIAR’ los consumos en los cuales los puntos del cliente no alcancen ni para el comodín más barato del rubro donde se realizó el consumo. Se estaría disminuyendo aún más el universo, dejando cargada una nueva interfaz llamada ‘ofsa40.MOV_PREMIADOS’.

```
declare
limit_in number:=1024;

CURSOR premiados_cur
IS
select P.nup, p.rubro, p.codop , p.fecha_mov, p.importe,p.local, p.tarjeta, p.usuario,p.v_mdp,p.saldo_puntos, T.DESC_RUBRO from
(select M.*, C.precio_minimo from ofsa40.mov_a_premiar M, (select min (precio_base) as precio_minimo, cod_rubro from OFSA40.LOYAL_COMODINES_POSTC group by cod_rubro) C
where M.saldo_puntos >= C.precio_minimo
and M.RUBRO = C.cod_rubro) P, OFSA40.LOYAL_COMODINES_POSTC T
where P.RUBRO = T.cod_rubro
and P.min_pre = T.precio_base;

TYPE premiados_aat IS TABLE OF premiados_cur%ROWTYPE
INDEX BY PLS_INTEGER;

l_premiados premiados_aat;
```

ILUSTRACIÓN 25 – FRAGMENTO SCRIPT PLSQL ASIGNACIÓN COMODÍN

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

En el script se vuelve a usar un cursor 'premiados_cur' para guardar la información, se hace intersección con la tabla 'ofsa40.LOYAL_COMODINES_POSTC' donde se encuentran los datos de los comodines por rubro con su costo en puntos y se selecciona el precio mínimo agrupado por rubro con una función provista por el motor Oracle llamada min()*.

Luego, mediante una subconsulta, se apunta a la tabla donde están los registros candidatos a premiar 'ofsa40.MOV_A_PREMIAR' y se traen todos los datos especificando en el where* que el saldo del cliente sea mayor o igual al precio mínimo del comodín para ese rubro. Acto seguido, mediante otra subconsulta, se traen los otros datos que faltan para llenar la nueva interfaz. El volcado se hace como en los pasos anteriores, abriendo el cursor donde está la información cargada en memoria 'premiados_cur' y mediante un bucle o loop* se realiza el volcado, dejando así la interfaz 'ofsa40.MOV_PREMIADOS' cargada.



The screenshot shows a database interface with a SQL query: `select * from OFSA40.MOV_PREMIADOS`. Below the query is a 'Data Grid' containing a table with the following columns: NUP, RUBRO, CODOP, FECHA_MOV, IMPORTE, LOCAL, TARJETA, USUARIO, V_MDP, SALDO_PUNTOS, and DESC_RUBRO. The table contains 18 rows of transaction data.

NUP	RUBRO	CODOP	FECHA_MOV	IMPORTE	LOCAL	TARJETA	USUARIO	V_MDP	SALDO_PUNTOS	DESC_RUBRO
02515316	5411	571	03/04/2020	400	VERDULERIA CARLITOS	4660570100049647	47217230	VISA	88683	SUPERMERCADO
05818034	5411	571	03/04/2020	1370,41	COTO 101	4050710152694753	894903706	VISA	72324	SUPERMERCADO
10559540	5411	571	03/04/2020	1403,76	COTO 136	4050710152843335	665600803	VISA	29755	SUPERMERCADO
80259640	5411	571	03/04/2020	790,4	EXPRESS PERU 487	4509950156791950	464310833	VISA	352502	SUPERMERCADO
00567048	5812	501	03/04/2020	1200	C.B.A	4660570101486285	87190621	VISA	306710	RESTAURANTE
05999528	5812	501	03/04/2020	900	EL GLOBITO	4050710150728330	894903201	VISA	126537	RESTAURANTE
80259640	5812	501	03/04/2020	1690	NAPOLES	4509950156791950	464310833	VISA	352502	RESTAURANTE
00567048	5541	571	03/04/2020	2600	SOMOS ENERGIA SA	3715950028758880	43439116	AMEX	306710	DESAYUNO
80259640	5541	571	03/04/2020	270	EST DE SERV GOLDEN CAR	4509950156791950	464310833	VISA	352502	DESAYUNO
05999528	5462	501	03/04/2020	680	PANADERIAS MCFF SRL	4050710150728330	894903201	VISA	126537	DESAYUNO
80259640	5462	501	03/04/2020	835	ADORADO BAR	4509950156791950	464310833	VISA	352502	DESAYUNO
05818034	5542	501	03/04/2020	2356,43	MERCADOPAGO*SHELLPASGALSA	4050710150728355	894903706	VISA	72324	COMBUSTIBLE
10094123	5542	571	03/04/2020	1607,55	ACA PALERMO	4660570098187904	644336846	VISA	87436	COMBUSTIBLE

ILUSTRACIÓN 26 – INTERFAZ MOVIMIENTOS PREMIADOS CARGADA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como se observa en la captura anterior, la interfaz ya no posee los consumos con 0 puntos (que si estaban en la ilustración 24), ya que estos no califican ni para canjear el comodín más barato, por lo cual fueron descartados.

Consecuente a lo explicado, ahora que se tiene en una interfaz todos los consumos que mínimamente pueden ser premiados con el comodín más barato, solo resta ofrecerle al movimiento el más caro posible. Esto quiere decir que a un cliente se lo puede premiar con el más barato, el intermedio o el más caro según le alcance su saldo de puntos.

```

1 declare
2   limit_in number:=1024;
3
4
5   CURSOR premiadosmax_cur
6   IS
7     select y.*, to_number( w.PENJMDOC) as nrodoc, w.PETIPDOC as tipodoc, PEPRIAPE as apellido, PENOMPER as nombre
8     from (select z.*, x.cod_articulo, x.prod_row_id
9     from (select h.*, f.codop, f.fecha_mov, f.importe, f.local, f.tarjeta, f.usuario,f.v_mdp,f.saldo_puntos, f.desc_rubro from
10    (select max (t.precio_base) as pre_max, t.nup, t.rubro from ( select p.*, C.PROD_ROW_ID, C.COD_ARTICULO, C.PRECIO_BASE from
11    ofsa40.mov_premiados p inner join OFSA40.LOYAL_COMODINES_POSTC c on p.rubro = c.cod_rubro
12    where P.SALDO_PUNTOS > C.PRECIO_BASE) t
13    group by t.nup, t.rubro) h, ofsa40.mov_premiados f
14    where h.nup = f.nup
15    and h.rubro = f.rubro) z, OFSA40.LOYAL_COMODINES_POSTC x
16    where z.pre_max = x.precio_base
17    and z.rubro = x.cod_rubro) y , OFSA40.VW_PEDT001 w
18    where y.nup = w.penumper;
19
20   TYPE premiadosmax_aat IS TABLE OF premiadosmax_cur%ROWTYPE
21   INDEX BY PLS_INTEGER;
22
23   l_premiadosmax premiadosmax_aat;

```

ILUSTRACIÓN 27 – FRAGMENTO SCRIPT PLSQL ASIGNACIÓN MÁXIMO COMODÍN

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Este script tiene una trama bastante compleja con cuatro subconsultas, una dentro de otra, donde simplícidamente lo que se hace es ir a la interfaz que posee los datos de los comodines de descuento 'ofsa40.LOYAL_COMODINES_POSTC' y se cruza con 'ofsa40.MOV_PREMIADOS' que contiene los movimientos con las personas a premiar. Como cada movimiento pertenece a un rubro diferente, se traen los posibles premios (comodín) de aquel donde el saldo de puntos del cliente

sea mayor al precio de los comodines. Una vez que se tienen los mismos, se opta por el máximo a través de la función Max(*) provista por el motor de base de datos Oracle.

Para ejemplificar: en Indumentaria se tienen comodines de 10%, 20% y 40%, y cuestan para canjearlos 1000, 2500 y 5000 puntos, respectivamente. Si el saldo del cliente es 3000 puntos, según la lógica explicada cuando se eligen los posibles premios, únicamente se traen los que cuestan menos de los puntos que tiene la persona (en este caso únicamente se traen 10% y 20%. Una vez allí, con estas dos opciones, se prioriza la elección del mayor, el comodín de 20 % queda con un costo de 2500 puntos.

Finalmente se guardan todos los premios (comodines) asignados a los consumos en una nueva interfaz 'ofsa40.MOV_PREMIADOS_MAX', vista a continuación.

Navigator

Data Grid

NOMBRE	APELLIDO	NRODOC	NUP	RUBRO	FECHA_MOV	LOCAL	IMPORTE	V_MDP	SALDO_P...	DESC_RUBRO	PROD_ROW_ID	COD_ARTICULO	PRECIO_BASE
ESTEBAN	LOPEZ GRANADO	34653307	05818034	5542	03/04/2020	MERCADOPAGO*SHELLPASGALSA	2356,43	VISA	72324	COMBUSTIBLE	1-29GFOS	PSC009	22000
NICOLAS SEBASTIAN	FRJINO	34520120	05999528	5812	03/04/2020	EL GLOBITO	900	VISA	126537	RESTAURANTE	1-24RBBZ	PSC002	25000
ENEAS NICOLAS	DELGADO	38709664	10559540	5411	03/04/2020	COTO 136	1403,76	VISA	29755	SUPERMERCADO	1-29WAL3	PSC011	22000
JESICA CECILIA	CONTINO	31915517	10094123	5542	03/04/2020	ACA PALERMO	1607,55	VISA	87436	COMBUSTIBLE	1-29GFOS	PSC009	22000
JUAN ANDRES	CLEMENTE	26770941	00567048	5541	03/04/2020	SOMOS ENERGIA SA	2600	AMEX	306710	DESAYUNO	1-29GFJC	PSC006	4000
VALERIA LAURA	VENEZIA	23100937	02515316	5411	03/04/2020	VERDULERIA CARLITOS	400	VISA	88683	SUPERMERCADO	1-29WAM7	PSC012	40000
PEDRO IGNACIO	SANTI	34027566	80259640	5411	03/04/2020	EXPRESS PERU 487	790,4	VISA	352502	SUPERMERCADO	1-29WAM7	PSC012	40000
PEDRO IGNACIO	SANTI	34027566	80259640	5462	03/04/2020	ADORADO BAR	835	VISA	352502	DESAYUNO	1-29GFJC	PSC006	4000
JUAN ANDRES	CLEMENTE	26770941	00567048	5812	03/04/2020	C.B.A	1200	VISA	306710	RESTAURANTE	1-24RBBZ	PSC002	25000
NICOLAS SEBASTIAN	FRJINO	34520120	05999528	5462	03/04/2020	PANADERIAS MCFF SRL	680	VISA	126537	DESAYUNO	1-29GFJC	PSC006	4000
PEDRO IGNACIO	SANTI	34027566	80259640	5812	03/04/2020	NAPOLES	1690	VISA	352502	RESTAURANTE	1-24RBBZ	PSC002	25000
ESTEBAN	LOPEZ GRANADO	34653307	05818034	5411	03/04/2020	COTO 101	1370,41	VISA	72324	SUPERMERCADO	1-29WAM7	PSC012	40000
PEDRO IGNACIO	SANTI	34027566	80259640	5541	03/04/2020	EST DE SERV GOLDEN CAR	270	VISA	352502	DESAYUNO	1-29GFJC	PSC006	4000

ILUSTRACIÓN 28 – INTERFAZ MOVIMIENTOS PREMIADOS MÁXIMOS CARGADA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como bien se explicó anteriormente, aquí se tienen los datos del consumo que se premió y los últimos tres campos hacen referencia al premio máximo que se asignó. Se puede observar que hay varios premios de comodines adjudicados a consumos de supermercado, en algunas ocasiones se otorgó el artículo código: PSC011, precio_base:22.000 puntos y en otras se otorgó artículo código: PSC012, precio_base: 40.000 puntos, esto es debido al saldo_puntos del cliente para lo que alcanzaba.

- **Lógica PL/SQL topes de reintegro y caducidad:**

La próxima acción a realizar es calcular los topes de reintegro que van a dar cada comodín de descuento ofrecido a los consumos que se tenían en la última interfaz mostrada (Ilustración 28), es decir cuánta plata Es lo que se le va a acreditar a la persona en cada premio.

```
for p in postc loop
  v_tope:=0;
  v_reintegro:=0;
  v_porc_desc:=0;
  --select nup into v_nup from
  -- selecciono el tope
  select cod_atl, tope_dto, porc_desc_usu into v_prod_part_num, v_tope, v_porc_desc
  from add01.ARTICULODATOSADICIONALES
  where cod_atl = p.PROD_PART_NUM;

  v_reintegro := (p.CONSUMO*(v_porc_desc/100));
  -- valido que el descuento no sea mayor al tope
  if v_tope < v_reintegro then
    v_reintegro := v_tope;
  end if;
```

ILUSTRACIÓN 29 – SCRIPT PLSQL CALCULO REINTEGRO EFECTIVO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

La operatoria del script es bastante simple, se apunta a la interfaz 'add01.ARTICULOSADICIONALES' es allí donde se encuentra el tope y porcentaje de descuento para cada comodín de ahorro, se traen ambos datos guardándolos en las variables v_tope y v_poc_desc. Luego el cálculo que hay que realizar para el reintegro en efectivo es $(\text{monto} * (\text{v_porc_desc}/100))$, siendo monto el valor de dinero de cada consumo que se quiere premiar, los cuales como se vio anteriormente estaban cargados en la interfaz final 'ofsa40.MOV_PREMIADOS_MAX'. Una vez calculado el reintegro en efectivo se guarda en la variable v_reintegro y allí se pregunta si el reintegro es mayor al tope guardado antes en v_tope, de ser afirmativo se actualiza v_reintegro con el valor de v_tope, ya que es lo máximo con que se puede premiar al cliente. Con estos valores calculados se procede a adjuntalos en la interfaz.

ID	CONS_NOMBRE_RUBRO	CONS_LOCAL	CONS_MONTO	APELLIDO	NOMBRE	PUNTOS_CLIENTE	PROD_PART_NUM	PROD_PUNTOS	REINT_EFECTIVO	PROD_TOPE	PORCENTAJE_DESC
395	RESTAURANTE	C.B.A	1200	CLEMENTE	JUAN ANDRES	306710	PSC002	25000	200	200	30
396	RESTAURANTE	EL GLOBITO	900	FRINO	NICOLAS SEBASTI...	126537	PSC002	25000	200	200	30
397	COMBUSTIBLE	ACA PALERMO	1607,55	CONTINO	JESICA CECILIA	87436	PSC009	22000	230	230	30
398	SUPERMERCADO	EXPRESS PERU 487	790,4	SANTI	PEDRO IGNACIO	352502	PSC012	40000	316,16	1300	40
399	DESAYUNO	EXPRESS PERU 487	385	SANTI	PEDRO IGNACIO	352502	PSC006	4000	60	60	30
400	COMBUSTIBLE	MERCADOPAGO*SHEL...	2356,43	LOPEZ GRANA...	ESTEBAN	72324	PSC009	22000	230	230	30
401	COMBUSTIBLE	ACA PALERMO	1607,55	CONTINO	JESICA CECILIA	87436	PSC009	22000	230	230	30
402	DESAYUNO	ADORADO BAR	835	SANTI	PEDRO IGNACIO	352502	PSC006	4000	60	60	30
403	DESAYUNO	SOMOS ENERGIA SA	2600	CLEMENTE	JUAN ANDRES	306710	PSC006	4000	60	60	30
405	SUPERMERCADO	COTO 101	1370,41	LOPEZ GRANA...	ESTEBAN	72324	PSC012	1000	548,16	1300	40
406	RESTAURANTE	NAPOLIS	1690	SANTI	PEDRO IGNACIO	352502	PSC002	25000	200	200	30
408	DESAYUNO	PANADERIAS MCFE SRL	680	FRINO	NICOLAS SEBASTI...	126537	PSC006	4000	60	60	30
409	COMBUSTIBLE	ESTACION DE SERV A...	1000	BENEDET	VERONICA MARIA	233899	PSC009	22000	230	230	30

ILUSTRACIÓN 30 – REINTEGROS EFECTIVO Y TOPE CARGADOS

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como se muestra en la imagen, ya está todo el circuito lógico completo para la premiación. Para ejemplificar se toma el primer caso:

La persona **CLEMENTE JUAN ANDRES** consumió en el local **C.B.A** del rubro **RESTAURANTE** por un monto de **\$1200**. Por dicho consumo se le ofrece para canjear un comodín de ahorro que tiene código **PSC002** y vale **25000** puntos, el

cual el cliente puede canjear correctamente ya que su puntaje es de **306710**. Este comodín de ahorro tiene un tope de **\$200** y un porcentaje de descuento de **30%**, por lo tanto, realizado el cálculo descrito anteriormente, el reintegro en efectivo que le quedaría por canjear este premio al cliente sería:

Monto * (v_porc_desc/100) = 1200*(30/100) = **\$360**, pero como este importe es mayor al tope, el reintegro termina quedando en **\$200**.

Lo que sigue es darle caducidad al premio, ya que cuando el cliente reciba la notificación de que fue beneficiado con la posibilidad de canje de un comodín ahorro por sus consumos mediante mail, no podrá hacerlo cuando quiera, sino que tendrá un plazo de 24hs luego de la notificación para efectuarlo.

Este plazo está marcado por un token que se genera segundos antes del envío de notificación.

CONS_TARJE	APELLIDO	NOMBRE	NRODOC	TOKEN	FECHA_GENERACION	ESTADO
4660570101486285	CLEMENTE	JUAN ANDRES	26770941	9EDC25047E950060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4050710150728330	FRINO	NICOLAS SEBASTI...	34520120	9EDC25047E920060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4660570098187904	CONTINO	JESICA CECILIA	31915517	9EDC25047E970060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4509950156791950	SANTI	PEDRO IGNACIO	34027566	9EDC25047E900060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4509950156791950	SANTI	PEDRO IGNACIO	34027566	9EDC25047E900060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4050710150728355	LOPEZ GRANA...	ESTEBAN	34653307	9EDC25047E930060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4660570098187904	CONTINO	JESICA CECILIA	31915517	9EDC25047E970060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4509950156791950	SANTI	PEDRO IGNACIO	34027566	9EDC25047E900060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
3715950028758880	CLEMENTE	JUAN ANDRES	26770941	9EDC25047E950060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4660570100049647	VENEZIA	VALERIA LAURA	23100937	9EDC25047E960060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA
4050710152694753	LOPEZ GRANA...	ESTEBAN	34653307	9EDC25047E930060E053B4A6297C8819	18/02/2020 12:15:39,000000 PM	ENVI-MYA

ILUSTRACIÓN 31 – TOKEN Y FECHA DE TOKEN

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

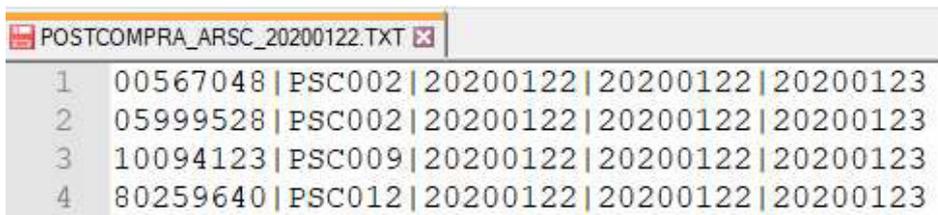
El token -como se puede apreciar en la ilustración- es un ID de treinta y dos caracteres, que se genera a través de un SYS_GUID()* ,una función del motor Oracle de la base de datos y es único (por persona y no por consumo). Si la misma persona tiene varios consumos en rubros diferentes y se les ofrece un

premio por cada uno, la caducidad de todos ellos se va a regir por este token . La fecha de generación del mismo está dada por SYSDATE()* otra función del motor Oracle. Si un cliente intenta canjear el premio pasadas las 24hs de la fecha_generacion, saldrá como expirado.

Una vez cargada la interfaz final con los consumos, con su correspondiente premio, los topes y reintegros y la caducidad de los mismos, se exporta la información y se realizan dos reportes:

- el primero para PRISMA. Se envía a la entidad el detalle de las personas que tienen posibilidad de premio y podrían llegar a canjear dentro de las 24hs. El formato del archivo es el siguiente

id_cliente | cód_articulo | fecha_consumo | inicio_premio | caducidad_premio



	id_cliente	cód_articulo	fecha_consumo	inicio_premio	caducidad_premio
1	00567048	PSC002	20200122	20200122	20200123
2	05999528	PSC002	20200122	20200122	20200123
3	10094123	PSC009	20200122	20200122	20200123
4	80259640	PSC012	20200122	20200122	20200123

ILUSTRACIÓN 32 – REPORTE PARA PRISMA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Con este reporte de potenciales canjes (que dependen de si el cliente quiere aprovechar el premio y canjearlo o no) pasadas las 24 hs Prisma revisa quiénes efectivamente canjearon el comodín de ahorro y procesa la acreditación del reintegro en efectivo en la tarjeta del cliente, de este modo se verá un importe positivo (+) en el próximo resumen de la misma.

- El segundo template diseñado a través de herramientas de maquetación como html y css, se envía vía mail a cada cliente la notificación de que posee comodín/es de ahorro para canjear como beneficio -gracias a sus consumos- y se le adjunta el link para poder ingresar a la web a efectuarlo.



Canje
Post Compra
SuperClub

Hola ESTEBAN,

De acuerdo con los consumos que realizaste la última semana, **si canjeas algunos de estos comodines podés ahorrar:**

Fecha	13/04/2020
Comercio	COTO 101
Importe	\$ 12764.32
Comodín	SUPERMERCADO
Puntos	40000
Ahorro	\$ 1300

Podés realizar el canje hasta las 23:59 hs del día de hoy.
[desde aquí.](#)

ILUSTRACIÓN 33 – MAIL NOTIFICACIÓN
CLIENTE

FUENTE: RELEVAMIENTO DE LO
REALIZADO EN EL TRABAJO

- **Lógica Java filtros finales de estado:**

Luego de que el cliente abra el mail descripto anteriormente, puede presionar el link que contiene el mismo (Ilustración 33), y será redirigido a la página web del proyecto. Ya dentro del sitio, se validarán todas las casuísticas de error posibles en tiempo real a la hora de canjear un comodín. Las mismas son descriptas a continuación

1. Token Vencido:

Como se explicó antes, el comodín de ahorro ofrecido tiene un tiempo de caducidad que está determinado por el token generado inmediatamente antes de enviar el mail o notificación. De esta manera, si una persona intenta ingresar vía link al sitio luego de transcurrido ese tiempo, esta situación será capturada por el código de excepciones de error java y devolverá la siguiente pantalla, indicándole al cliente que ya no posee canjes disponibles.



ILUSTRACIÓN 34 – PANTALLA TOKEN VENCIDO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

2. Producto sin stock:

Otro de los posibles escenarios que pudiera ocurrir al momento del canje del comodín, es que no se posea stock ya que estos son, en realidad, productos dentro del catálogo de Superclub y como tales, se posee un stock finito. Por ello, si esto llegara a suceder, el sitio arrojará lo siguiente.



← → ↻ 🏠 ⓘ No es seguro

 SuperClub poscompra

¡Canjeá puntos y ahorrá en tus consumos de la última semana!

Puntos SuperClub: 285.986 | Puntos a canjear: 0 | Ahorro total: \$0 | [Canjear](#)

Tus comodines disponibles

Icono	Categoría	Compras	Ahorro
	Supermercado	Compraste en EXPRESS PERU por \$790,40	Canjeá 40.000 puntos y ahorrá \$316,16
	Desayuno	Compraste en ADORADO BAR por \$835,00	Canjeá 4.000 puntos y ahorrá \$60,00
	Restaurante	Compraste en NAPOLÉS por \$1.690,00	Canjeá 25.000 puntos y ahorrá \$200,00

ILUSTRACIÓN 35 – INTENTO CANJE SIN STOCK

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO



ILUSTRACIÓN 36 – PANTALLA PRODUCTO SIN STOCK

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

3. Puntos insuficientes:

Otra variante de estado de error se da cuando se quiere canjear un determinado comodín de ahorro que tiene un costo en puntos mayor a los puntos que realmente se poseen. Esto sucede cuando un cliente tiene movimientos en distintos rubros a premiar y se le ofrecen varios comodines, uno por cada uno de ellos. Si bien de entrada se le ofrecieron todos (porque los puntos le alcanzaban para cada uno de ellos individualmente) puede pasar que luego de canjear una parcialidad de ellos, los puntos sean insuficientes para los restantes. Cuando esto pasa, el sitio muestra el siguiente error.

 SuperClub poscompra

¡Ahorrá en tus consumos de la última semana!

Tus comodines disponibles



Supermercado

Compraste en EXPRESS PERU
por \$790,40

Canjeá 40.000 puntos y ahorrá
\$316,16



Restaurante

Compraste en NAPOLES por
\$1.690,00

Canjeá 25.000 puntos y ahorrá
\$200,00

Puntos SuperClub 38590	Puntos a canjear 40.000	Ahorro total \$316,16
----------------------------------	-----------------------------------	---------------------------------

Canjear

ILUSTRACIÓN 37 – INTENTO CANJE CON PUNTOS INSUFICIENTES

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO



ILUSTRACIÓN 38 – PANTALLA ERROR PUNTOS INSUFICIENTES

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

4. Error inesperado:

Por último, aparece un estado de error que puede darse inesperadamente por diferentes situaciones, como pueden ser: interrupción de la comunicación web entre cliente (navegador) y servidor, saturación del servidor, bajo tiempo de respuesta (timeout) del servidor lógico. Etc.

Siempre que ocurra algún tipo de conflicto de infraestructura como los nombrados, java catchea el estado de error y devolverá la siguiente pantalla.



ILUSTRACIÓN 39 – PANTALLA ERROR INESPERADO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

- **Lógica Batch automatismo del proceso:**

Finalmente queda automatizar toda la cadena lógica de pasos PL/SQL que se explicaron hasta el momento, de manera que el front-end* o la visual del sitio web cuente con datos y comodines para ofrecer.

Esto se logra gracias a los archivos batchs que se describieron en el apartado de tecnologías. Los mismos funcionan de una manera genérica, conectando primero el ambiente del servidor en donde se encuentra el archivo de formato PL/SQL a ejecutar y luego llamándolo con las condiciones que necesite por ej: cuando arrancó el paso inicial de ir a buscar las transacciones de los clientes a las tablas base, hay que pasarle - como condición- la fecha específica con la

cual se quieren recuperar estas transacciones, ya que las tablas en cuestión contienen toda la historia.

A continuación se muestra el batch del primer paso sql.

```

canje_postc_movpres.sh
12 #=====
13 # setvar
14 # Declaracion De Variables de ambiente
15 #=====
16 setvar(){
17     APP_SRV {} = ['DBLXORAFRONT01', @user, @pass]
18     DIR_SP = /aplicaciones/ofsa/sql
19     FILE_LOG =//aplicaciones/ofsa/output
20     export DIR/SP, FILE_LOG
21 }
22
23 #=====
24 # setarch
25 # Seteo De Archivos
26 #=====
27 setarch(){
28     #Archivos Sql
29     FILE_SP1=${DIR_SP}/mov_presentados_paso1.sql
30     export FILE_SP1
31 }
32
33 FECHA_SQL=$1
34
35 #=====
36 # Inicio programa principal
37 #=====
38
39 #Seteo
40 setvar
41 setarch
42 ./aplicaciones/ofsa/ejecutables/ofsa_utiles.sh
43 #setvariables#
44 #setarchivos
45
46 echo -e "===== " >>${FILE_LOG}
47 echo -e " Ejecución de PROCESO POST COMPRA " >>${FILE_LOG}
48 echo -e "===== " >>${FILE_LOG}
49 echo -e " " >>${FILE_LOG}
50 echo -e " " >>${FILE_LOG}
51 #Log Para el Operador
52 #Main
53 testconn
54
55
56 echo -e "*" Ejecución de PROCESO mov_presentados_paso1 "*" >>${FILE_OPER}
57 excsql ${FILE_SP1} ${FECHA_SQL} "mov_presentados_paso1.sql"
58
59 resumen
    
```

ILUSTRACIÓN 40 – SCRIPT BATCH CANJE_POSTC_MOVPRES.SH

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

En la imagen se observa el script batch ‘canje_postc_movpres.sh’, donde inicialmente se setean los parámetros de conexión al servidor y las rutas correspondientes dentro del mismo (APP_SRV, DIR_SP, FILE_LOG). Estos apuntan dónde están los códigos fuentes sql y un directorio de outputs donde

se loguea la ejecución de los archivos lo que permite ver e investigar los errores/excepciones técnicos en el caso de fallos o cancelaciones productivas. Luego se declara el fuente en cuestión que se quiere ejecutar en el parámetro FILE_SP1. Por último se hace la llamada explícita en la línea 'execsql \${FILE_SP1} \${FECHA_SQL}' aquí se da la orden de que se ejecute el primer paso lógico PL/SQL 'mov_presentados_paso1.sql' que se detalló anteriormente en la sección [Lógica PL/SQL movimientos a premiar](#).

De la misma manera en que opera este fuente batch operan todos los demás, que en conjunto forman la cadena que mecaniza el proceso entero de 'Canje Postcompra'. Como bien se explicó anteriormente, la herramienta donde está montada esta cadena de automatización es control-m, que se presenta a continuación.

En primera instancia, se muestra la cadena dentro de la herramienta, la cual por ser muy larga y tener varios pasos, se ve cortada

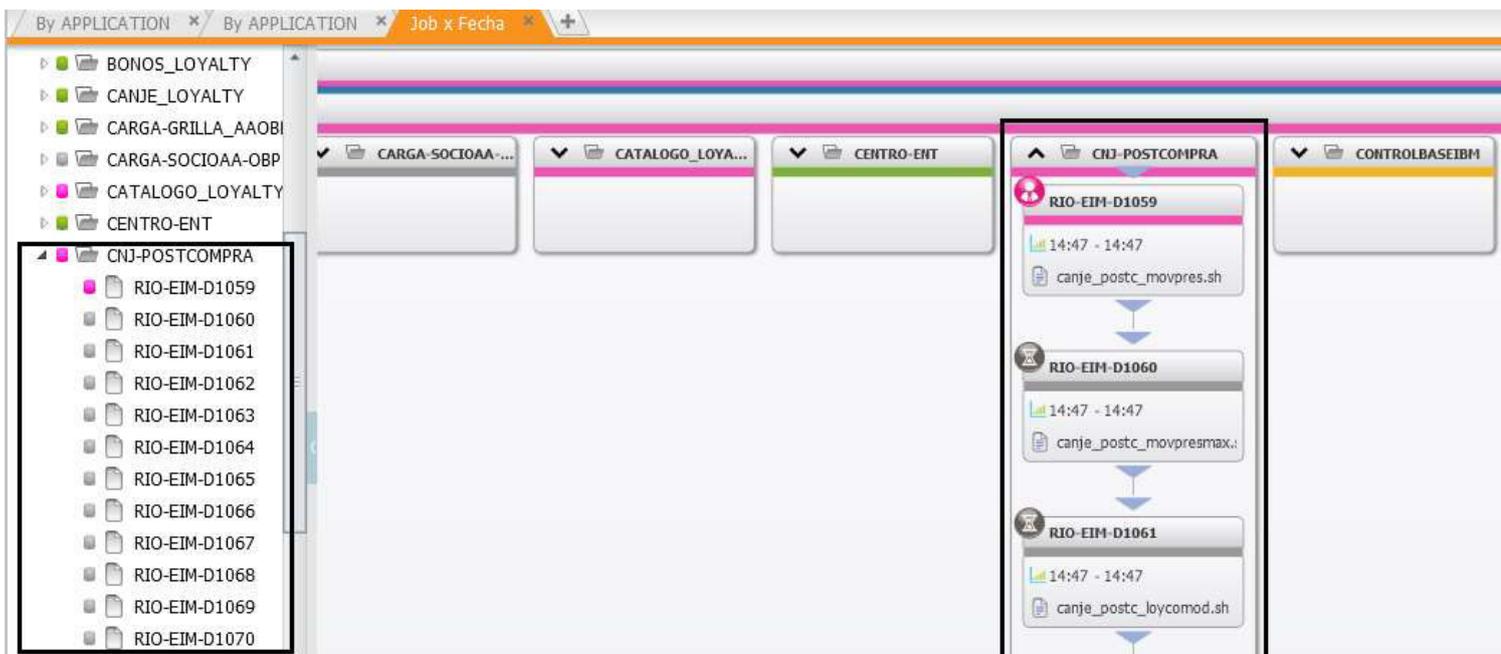


ILUSTRACIÓN 41 – CADENA CANJE-POSTCOMPRA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Luego, la otra imagen donde se ve la cadena completa con sus continuaciones adjuntadas de manera lateral para poder apreciar la cantidad extensa de pasos que esta contiene.

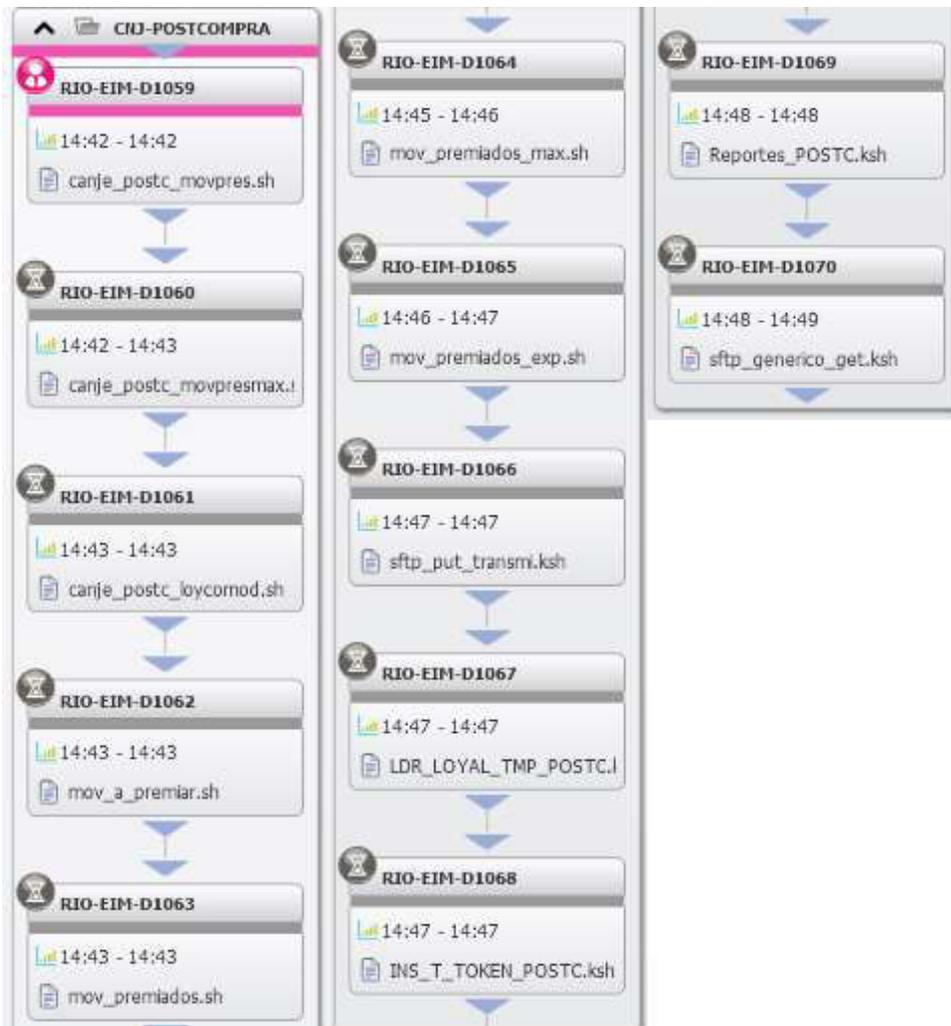


ILUSTRACIÓN 42 – CADENA CANJE-POSTCOMPRA ENTERA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Cada uno de estos cubos/cajas, llamados jobs, ejecutan de manera individual cada paso lógico PL/SQL que se explicó a lo largo de este trabajo dejando toda la información lista para su posterior visualización en el sitio web. Todo inicia en el primer job RIO-EIM-D1059, que está programado para ejecutar todos los días luego de que se cumplan sus condiciones predecesoras, que son las siguientes:

Job: RIO-EIM-D1059

General | Scheduling | Run Information | **Prerequisites** | Actions

In Conditions

Name	Order Date	Specific Date	
RIO-AFDMDR90-TO-RIO-EIM-D1059	Specific Dat	15/06	<input checked="" type="checkbox"/>
RIO-AFDMDR88-TO-RIO-EIM-D1059	Specific Dat	15/06	<input checked="" type="checkbox"/>
HCI_AFDMDR72-EIM1059	Specific Dat	15/06	<input checked="" type="checkbox"/>

Conditions Relationship: AND Between All

Control Resources

Quantitative Resources

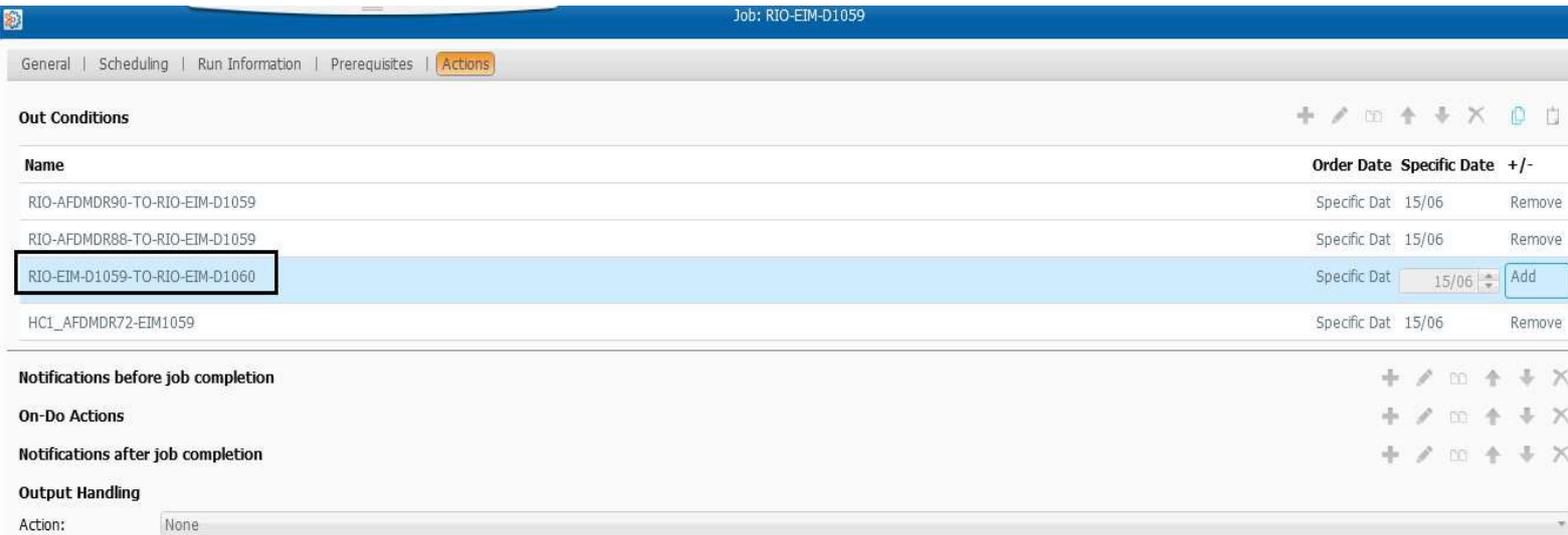
Resource Name	Required Q	Total Am
CPU@dbkorafront01	1	100

ILUSTRACIÓN 43 – PRE CONDICIONES CADENA BATCH

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Estas tres condiciones llenan las tablas base que se necesitan para comenzar la lógica de este sistema, se está hablando de ofsa40.TMOVPREPRES_AMEX (transacciones crédito Amex), ofsa40.TMOVPREPRES_VISA (transacciones crédito Visa) y ofsa40.TRANSACCIONES_ATM (transacciones debito). Una vez que el sistema detecte que esos pasos previos corren correctamente y está la información disponible, arranca el primer job de la cadena RIO-EIM-D1059 el cual ejecuta el fuente batch 'canje_postc_movpres.sh'.

Luego de que termine su ejecución exitosa, da condición sucesora al siguiente job de la cadena RIO-EIM-D1060. Se aprecia en la siguiente ilustración.



Name	Order Date	Specific Date	+/-
RIO-AFDMDR90-TO-RIO-EIM-D1059	Specific Dat	15/06	Remove
RIO-AFDMDR88-TO-RIO-EIM-D1059	Specific Dat	15/06	Remove
RIO-EIM-D1059-TO-RIO-EIM-D1060	Specific Dat	15/06	Add
HCL_AFDMDR72-EIM1059	Specific Dat	15/06	Remove

ILUSTRACIÓN 44 – POS CONDICIONES CADENA BATCH

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

De esta manera, transcurren todos los Jobs sucesivamente en la cadena hasta finalizar el proceso. No comenzará el próximo hasta que el anterior no termine y deje la condición para arrancar. Esta consecución de pasos es mecánica y no requiere que ningún usuario este pendiente de ella, salvo que surja alguna cancelación u error productivo.

2.3.3 Test y Validación:

Esta etapa es también conocida como QA (Quality Assurance) o Control de Calidad. En ella se realizan una serie o batería de pruebas en un ambiente anterior a **producción** al que llamamos **beta**. Esto es a modo de control del flujo

de ejecución, sirve para asegurar que los resultados arrojen un flujo correcto y no se presenten errores tanto en la operatoria como en los resultados; ya que caso contrario de encontrar errores o incongruencias, será necesario corregirlos antes de dar por exitosa esta etapa.

Se abordarán dos tipos de pruebas: las unitarias y de regresión.

- **Pruebas unitarias de funcionalidad:**

Aquí se llevan a cabo ensayos de varias casuísticas de premios de algunos miembros ficticios que están disponibles en el ambiente beta mencionado anteriormente.

Se simula la ejecución de la cadena batch que automatiza toda la lógica. La persona recibe un mail en su casilla con el ofrecimiento de premios por sus consumos junto con el link el que le permite ingresar al sitio de canjes.

Luego de presionarlo, sucede la redirección al sitio web el cual mostramos a continuación.



ILUSTRACIÓN 45 – SITIO WEB PREMIOS PARA CLIENTE FICTICIO 1

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Para esta primer prueba se observa un **cliente ficticio 1**, con un solo consumo, que sucedió en **Supermercado COTO** por un total de **\$1.403.76**. El ofrecimiento que se le da al cliente es canjear un comodín de ahorro con un costo de **22.000** puntos y ahorrar **\$280.75**. En el lateral del botón de canje aparece el resumen de los puntos que el cliente posee en Superclub, y lo que se le va a restar. Se procede a tildar el comodín y luego a presionar el botón de canjear. Allí se deben aceptar los términos y condiciones

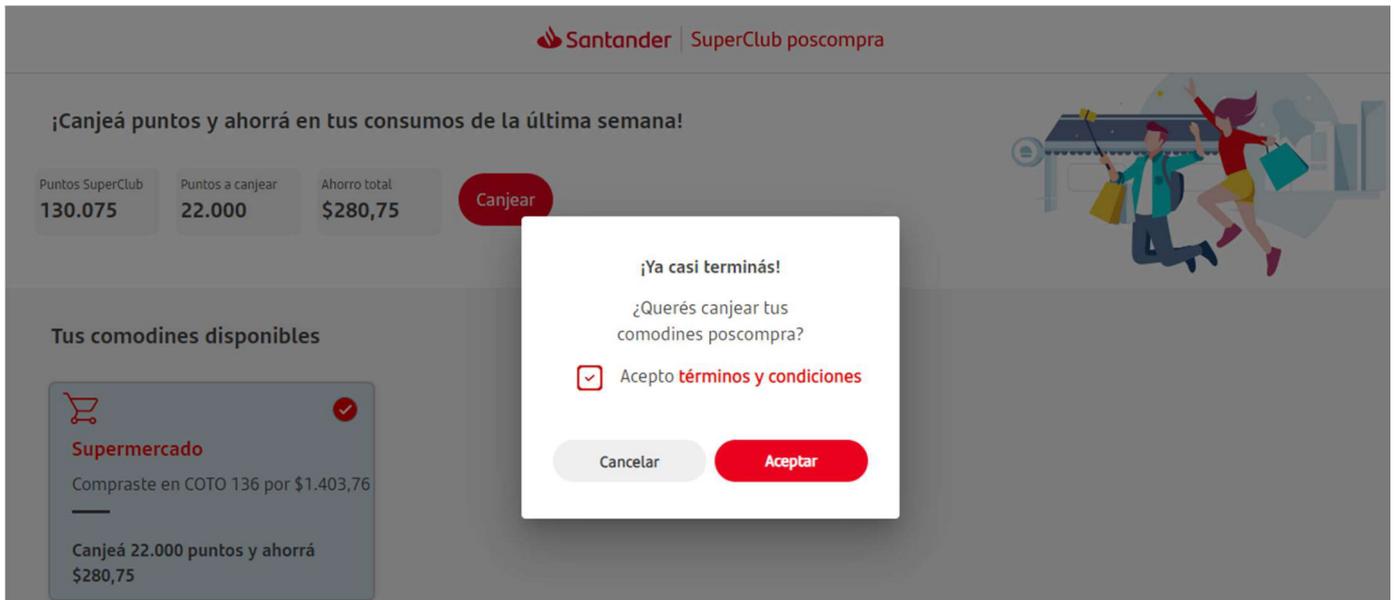


ILUSTRACIÓN 46 – SITIO WEB PREMIOS TÉRMINOS Y CONDICIONES

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Finalmente el sitio nos arroja el resultado del canje



ILUSTRACIÓN 47 – SITIO WEB PREMIOS CANJE CORRECTO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Otra prueba más que se debe hacer dentro de las casuísticas es canjear varios premios de una sola vez, ver cómo se comporta el sitio y si realmente descuenta los puntos anunciados.

Por consiguiente se observa el funcionamiento de la web para otro cliente ficticio 2



ILUSTRACIÓN 48 – SITIO WEB PREMIOS PARA CLIENTE FICTICIO 2

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Se procede a tildar dos comodines para su canje, sumando en la parte superior el total de los puntos a canjear y ahorro total.

Paso siguiente presionamos el botón de canje y aguardamos la respuesta.

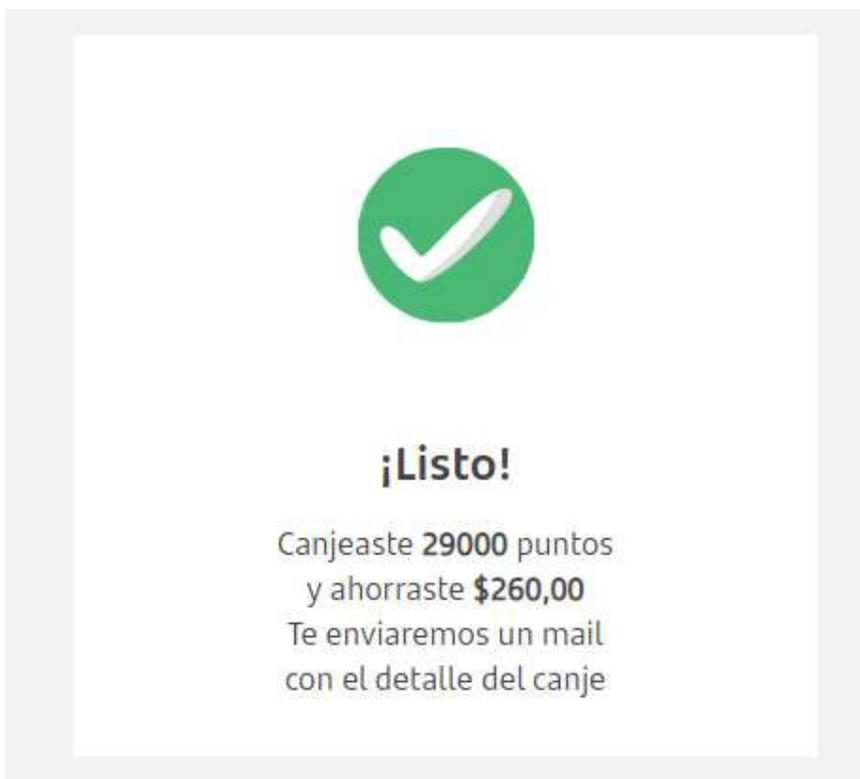


ILUSTRACIÓN 49 – SITIO WEB PREMIOS CANJE CORRECTO CLIENTE FICTICIO 2

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Finalmente, luego del canje, se vuelve a la web del **cliente ficticio 2**



ILUSTRACIÓN 50 – SITIO WEB PREMIOS PARA CLIENTE FICTICIO 2

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Hay que fijarse que, de los tres comodines de descuento que tenía inicialmente luego de canjear dos de ellos, los mismos desaparecen (sólo pueden ser usados una vez) y queda disponible únicamente el que no se tildó.

También se confirma el correcto descuento de puntos.

Queda mostrar para una prueba más integral, la acreditación real de dinero que reintegran los comodines. Esto se detallará más adelante en las pruebas que siguen.

- **Prueba de regresión :**

En este apartado se lleva a cabo una prueba del sistema punta a punta, integradora, de manera de cubrir todos los pasos lógicos conjuntos y asegurarse de que no solo funcionan bien individualmente, sino que lo hacen también en conjunto.

Para que la prueba sea más gráfica, se lleva a cabo sobre el cliente/empleado Leguizamón Gonzalo Uriel.

Primeramente se ejecuta la cadena batch del sistema que automatiza la lógica.

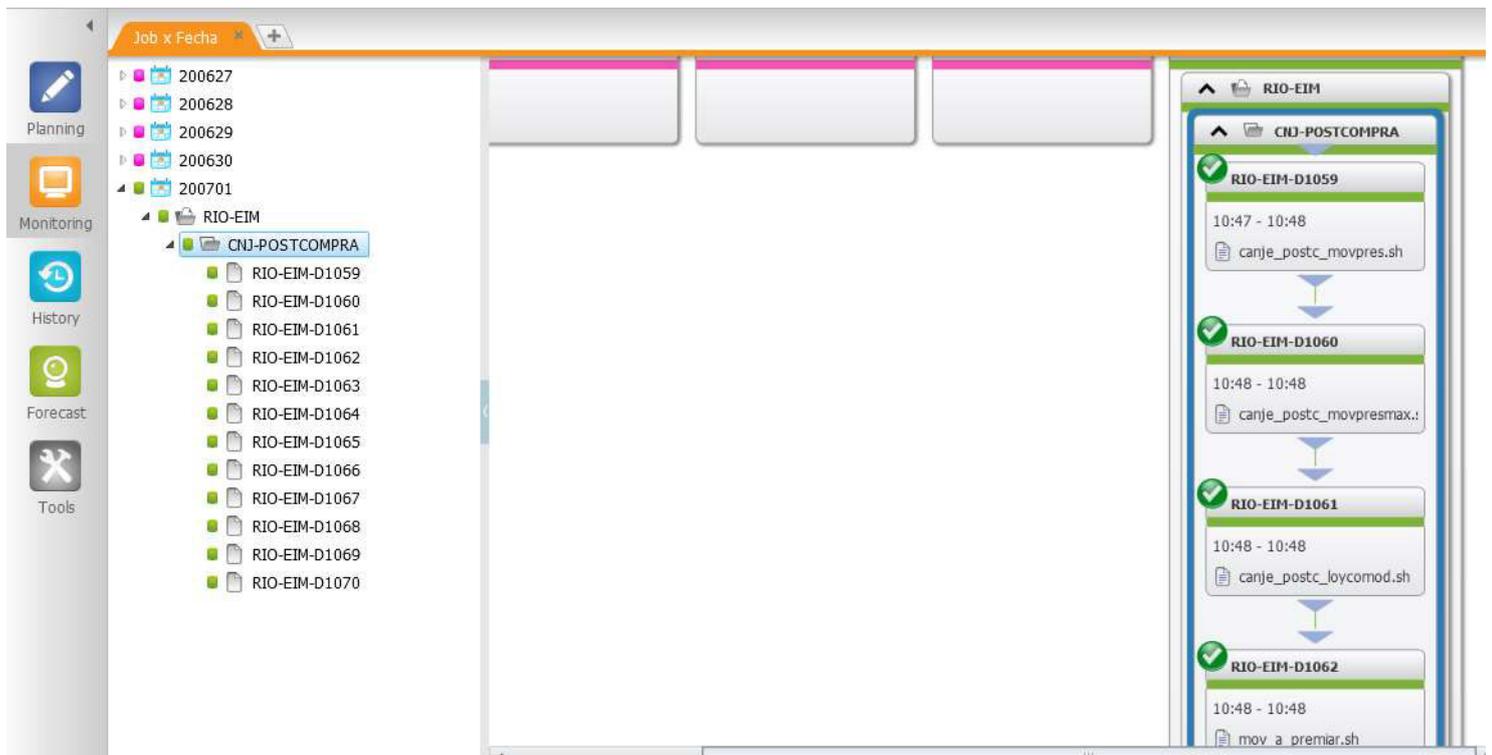


ILUSTRACIÓN 51 – CADENA BATCH POSTCOMPRA

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Luego de que la cadena termina se lanzan los mails con las ofertas de comodines postcompra del día. Se adjunta en la siguiente imagen la recibida por el cliente en cuestión GL.



ILUSTRACIÓN 52 – MAIL POSTCOMPRA LEGUIZAMÓN GONZALO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Una vez recibido el mail, se presiona el link señalado el cual nos redirecciona a la web del sitio de canjes, en donde se puede observar el comodín disponible por un único consumo realizado en el rubro Combustibles.

Se procede a canjearlo.



The screenshot shows a web browser window with the Santander SuperClub poscompra interface. At the top, there is a navigation bar with the Santander logo and the text "SuperClub poscompra". Below this, a promotional banner reads "¡Canjeá puntos y ahorrá en tus consumos de la última semana!". To the right of the banner is an illustration of a man and a woman shopping. Below the banner, there are three data boxes: "Puntos SuperClub" with the value 103.040, "Puntos a canjear" with the value 38.000, and "Ahorro total" with the value \$500,00. To the right of these boxes is a red "Canjear" button, which is highlighted by a blue arrow. Below the banner, there is a section titled "Tus comodines disponibles" containing a card for "Combustible" with a red checkmark icon. The card details a purchase at JUMBO QUILME for \$1.000,00 and states that 38.000 points were redeemed, resulting in a \$500,00 saving.

Puntos SuperClub	Puntos a canjear	Ahorro total
103.040	38.000	\$500,00

Canjear

Tus comodines disponibles

Combustible ✓

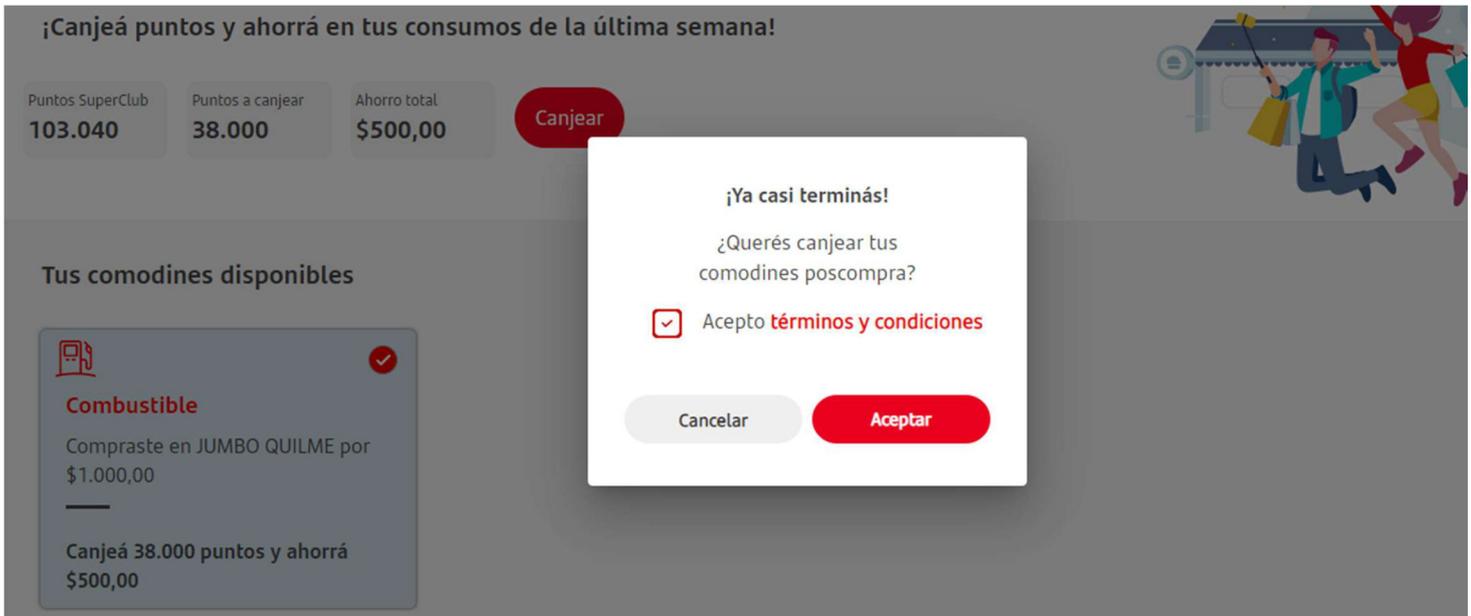
Compraste en JUMBO QUILME por \$1.000,00

Canjeá 38.000 puntos y ahorrá \$500,00

ILUSTRACIÓN 53 – SITIO WEB DE CANJES LEGUIZAMÓN GONZALO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Se aceptan los términos y condiciones para confirmar el canje.



¡Canjeá puntos y ahorraré en tus consumos de la última semana!

Puntos SuperClub	Puntos a canjear	Ahorro total	Canjear
103.040	38.000	\$500,00	

Tus comodines disponibles

Combustible ✓

Compraste en JUMBO QUILME por \$1.000,00

Canjeá 38.000 puntos y ahorrá \$500,00

¡Ya casi terminás!

¿Querés canjear tus comodines poscompra?

Acepto **términos y condiciones**

Cancelar Aceptar

ILUSTRACIÓN 54 – SITIO WEB DE CANJES TÉRMINOS Y CONDICIONES LEGUIZAMÓN GONZALO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Luego de aceptados y efectuado el canje, el sitio informa el resultado de la operación.

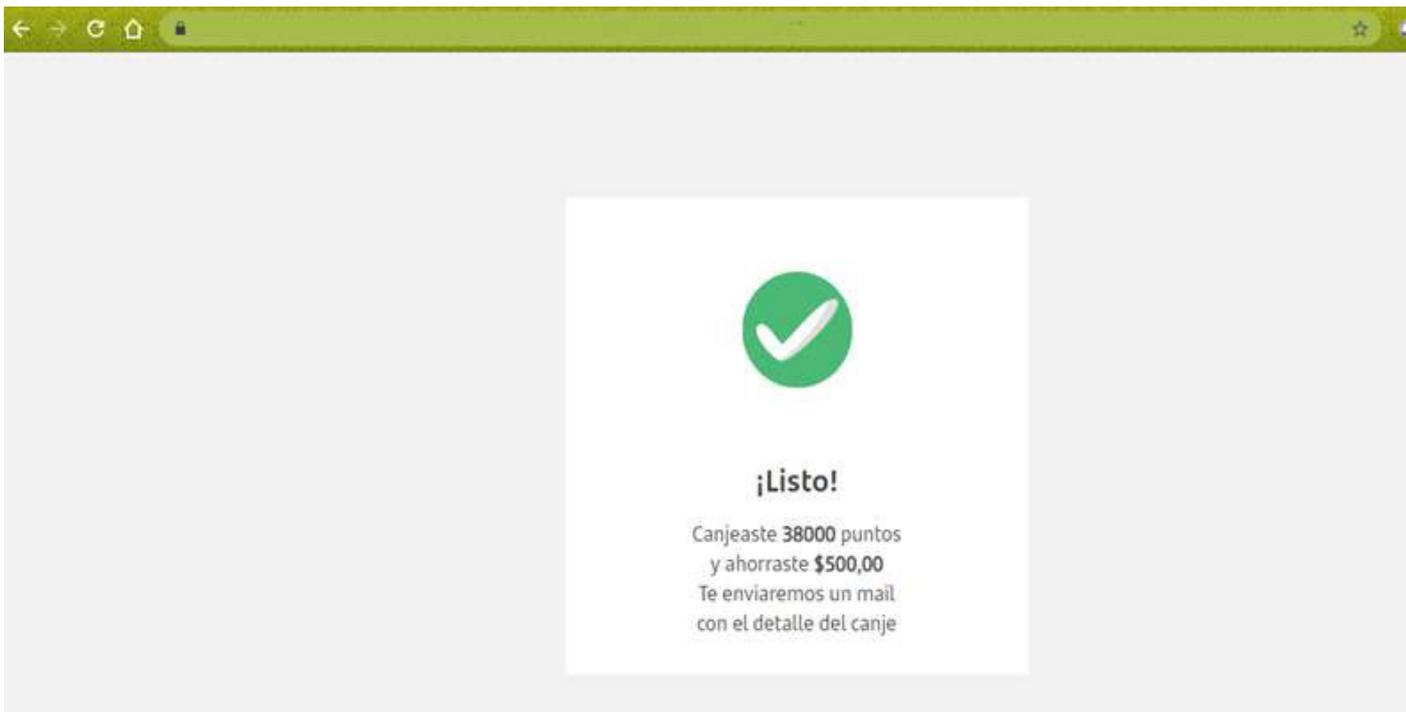


ILUSTRACIÓN 55 – SITIO WEB CANJE CORRECTO LEGUIZAMÓN GONZALO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Finalmente luego de terminada la operatoria, cuando el canje es correcto se envía nuevamente por mail (como bien muestra la imagen) el detalle de lo realizado. Esto es a manera de comprobante para el cliente por si llegase a surgir algún inconveniente a futuro





Canje de puntos SuperClub

GONZALO URIEL LEGUIZAMON,

¡Nos alegra ayudarte a ahorrar en tus compras, incluso después de que las hiciste!

Resumen del canje:

 **Combustible** Canjeaste

Reintegros: si la compra fue realizada con Tarjeta Santander Débito, el reintegro se realizará en la cuenta dentro de los 15 días hábiles posteriores de realizado el canje de puntos; y si se realizó con la Tarjeta Santander Crédito, se efectuará dentro de los dos resúmenes posteriores al canje.

ILUSTRACIÓN 56 – MAIL AVISO CANJE CORRECTO LEGUIZAMÓN GONZALO

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

En la imagen anterior aparece una leyenda importante llamada “*Reintegros*”. La misma describe de qué manera se dará el depósito de los \$500 ahorrados con el comodín canjeado en este caso particular. Esto depende del tipo de tarjeta con el cual se realizó el consumo que fue premiado (\$1000 comprados en Combustible en el establecimiento Jumbo Quilmes). Por lo tanto, si se realizó con débito, dentro de los próximos 15 días hábiles -como plazo máximo- ya se debería poder ver el depósito en la cuenta del cliente. Si se realizó con crédito (Visa o Amex) recién se podrá ver impactado dentro de los próximos dos resúmenes de la tarjeta- como plazo máximo-.

2.3.4 Implementación:

Finalmente luego de la etapa de pruebas, quedan pasar los códigos fuentes del desarrollo a **producción**. Es decir, se pone a disposición de varios usuarios empleados para que puedan acceder y utilizar el sitio web del nuevo beneficio en el día a día. Como ya fue explicado anteriormente, el sistema sólo será de acceso para un universo reducido de personas (Marketing y Legales, aproximadamente 30 personas) de manera de ser probado diariamente y evaluar cuál es el real impacto económico y de utilidad del mismo, para luego liberarlo hacia todo el universo de clientes del banco.

Para la implementación se trabaja con un esquema de infraestructura DevOps*, en donde se utiliza el repositorio GIT para el versionado y subida de los códigos fuentes.

A continuación se describen los pasos necesarios:

1. Partimos de un proyecto en GitLab llamado premiaciones, es allí donde se encuentran los diferentes hilos para la inserción de los códigos fuentes.

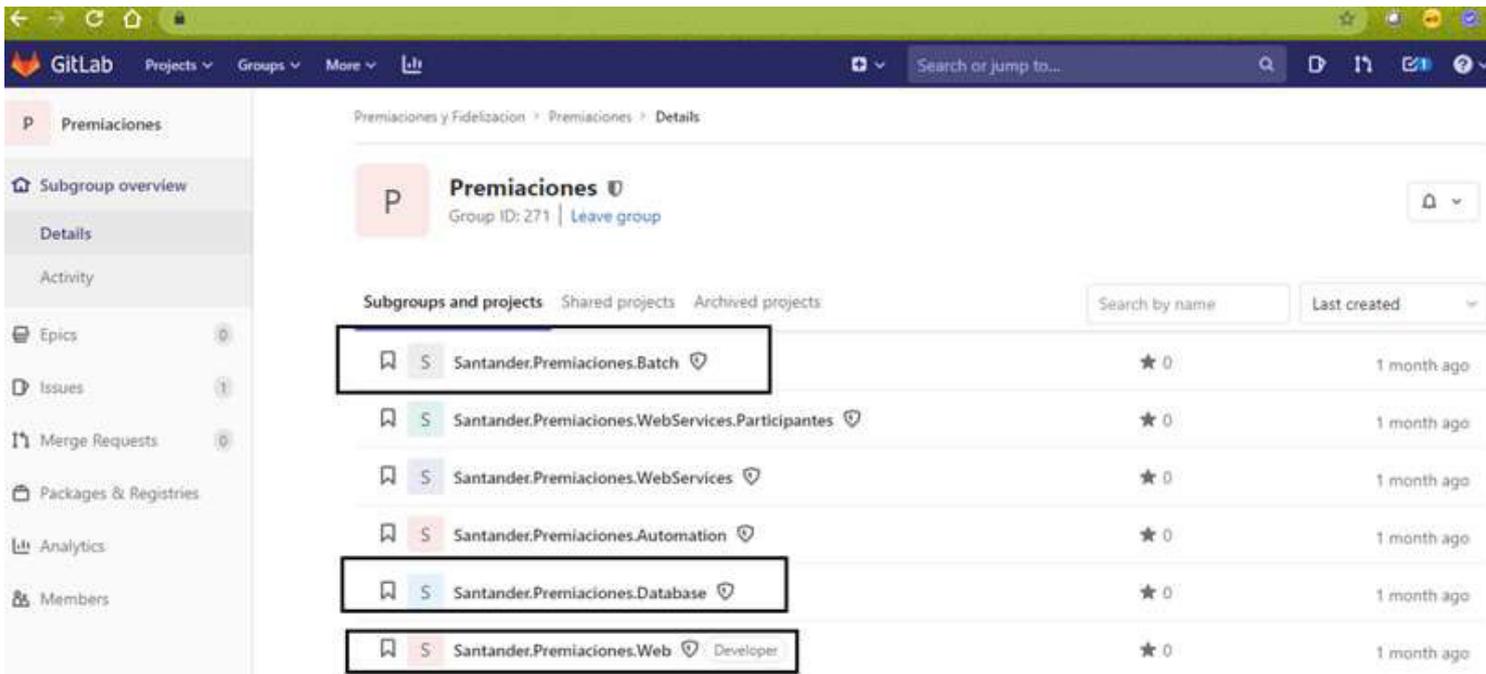


ILUSTRACIÓN 57 – GITLAB PREMIACIONES

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Como se puede observar en la imagen, las diferentes carpetas con recuadros son donde van a insertarse los códigos fuentes Batches, Pl/sql y Java respectivamente.

2. Se realiza un pull del repositorio anterior mencionado para actualizar los archivos que están en la nube a la última versión disponible, quedando así el espacio listo para poder subir los códigos fuentes nuevos de este evolutivo Canje-Postcompra

3. Se genera una rama o branch nuevo con un nombre que hace alusión a lo que se va a subir, por ejemplo: **'feature/postcompra'**. Esto se hace de manera local, en la pc del desarrollador. Seguido a esto se procede a subir, también de manera local, los archivos de desarrollo por separado Pl/sql, batch y java.
4. Una vez ya subidos, se agregan al repositorio con la sentencia "Git Add ."

```
A308781@NBi660392 MINGW64 /c/Gitlab/PREM/BATCH/santander.premiaciones.batch (feature/postcompra)
$ git add .
```

5. El paso siguiente, se confirman con la sentencia " Git Commit -m "

```
A308781@NBi660392 MINGW64 /c/Gitlab/PREM/BATCH/santander.premiaciones.batch (feature/postcompra)
$ git commit -m 'subida de fuentes postcompra'
[feature/postcompra 0878cb5] subida de fuentes postcompra
```

6. Y finalmente se realiza un push con la sentencia "Git push origin feature/postcompra". Esto lo que hace es terminar de impactar los códigos fuentes, que estaban de manera local en la pc del desarrollador, en el repositorio gitlab en la nube.

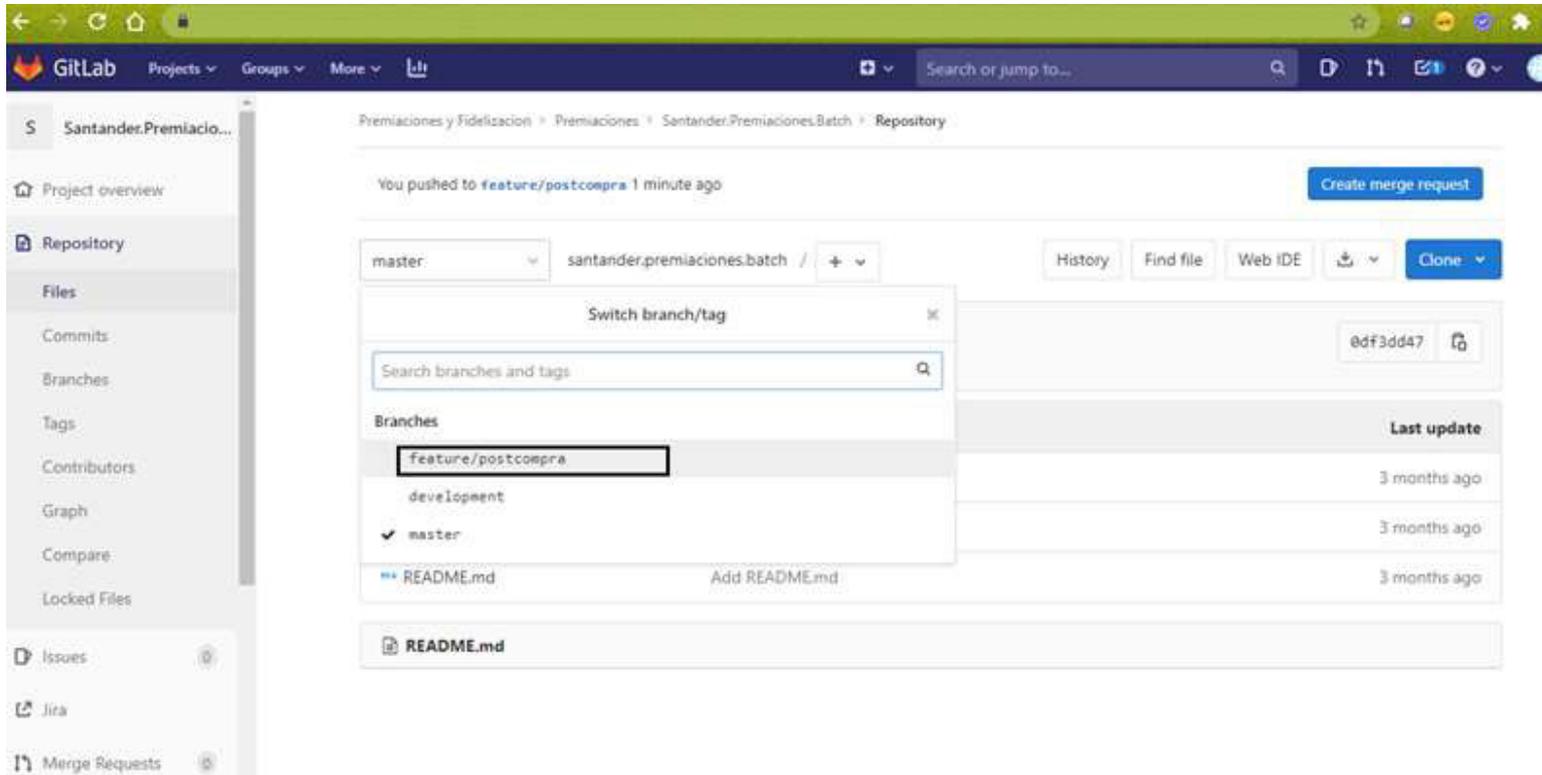


ILUSTRACIÓN 58 – GITLAB PREMIACIONES RAMAS

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Aquí se puede observar cómo la rama **feature/postcompra** que existía sólo localmente, ya está subida e impactada en gitlab en la nube, junto con las otras dos ramas que vienen por defecto, **development** y **master**

7. Lo que resta para culminar la implementación es que la rama nueva creada **feature/postcompra** llegue a la rama **master**. Para ello, primeramente, se pide un merge request* hacia **development** que lo que hace es impactar los códigos fuentes en la misma.

Premiaciones y Fidelización > Premiaciones > Santander.Premiaciones.Batch > Merge Requests > New

New Merge Request

Source branch	Target branch
premiaciones-y-fidelizacion/pre... feature/postcompra	premiaciones-y-fidelizacion/pre... development
 prueba 2 Leguizamon, Gonzalo Uriel authored 18 minutes ago b3334037	 Anthony actualizacion yml Leon Ruiz, Anthony Jose authored 2 months ago c8e9f545

ILUSTRACIÓN 59 – GITLAB MERGE REQUEST DESDE POSTCOMPRA HACIA DEVELOPMENT

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

Una vez que están los códigos fuentes mergeados, se pide un nuevo merge request* pero ya desde la rama **development** hacia **master**. Al finalizar esto, el desarrollo ya se encuentra productivo.

New Merge Request

Source branch	Target branch
premiaciones-y-fidelizacion/pre... development	premiaciones-y-fidelizacion/pre... master
 Anthony actualizacion yml Leon Ruiz, Anthony Jose authored 2 months ago c8e9f545	 Anthony Primer Commit Leon Ruiz, Anthony Jose authored 3 months ago 0df3dd47

[Compare branches and continue](#)

ILUSTRACIÓN 60 – GITLAB MERGE REQUEST DESDE DEVELOPMENT HACIA MASTER

FUENTE: RELEVAMIENTO DE LO REALIZADO EN EL TRABAJO

3. CONCLUSION:

3.1 Resultado, posibles mejoras y ampliaciones:

Luego de varias semanas de uso, las estadísticas arrojaron un saldo positivo. Gracias al diseño y desarrollo del beneficio se vio un marcado aumento de interacción de los clientes de la entidad, como también la participación de clientes inactivos. Esto no sólo generó impacto comercial sino que también supuso una aprobación de parte del público para con la funcionalidad del sistema. Sin embargo, para seguir generando valor a dicha funcionalidad y ofrecer un plus a quienes lo utilizarán, ya se estimaron algunas modificaciones al proyecto:

- Adaptar la lógica para transformar el proceso en semanal y no diario como ocurre hoy día. De manera que el envío de mail con los comodines de descuento no sucederá todos los días sino una vez por semana, por lo tanto, no será tan invasivo de cara al cliente.
- Ampliar la gama de rubros actuales que se premian, incluyendo algunos nuevos de necesidades básicas como Farmacias, etc.
- A futuro, será necesario mejorar el tiempo de respuesta de la aplicación en la transición desde que se canjea hasta que muestra el estado de la operación (correcto o inválido). Para ello, será primordial analizar la forma en que se generan las vistas y encontrar la manera de mejorar esta performance.
- Incorporar al beneficio las transacciones realizadas con plásticos MasterCard que hoy día no están incluidas (solo Visa y Amex). En este caso, además de desarrollar una interfaz nueva del lado de

Superclub, también se debe coordinar con Prisma para que haga lo propio para incluirlas.

- Este sitio interactúa con otros componentes dentro del banco, como lo es Mensajes & Avisos y con agentes externos como Prisma. Hoy día, esa interacción se da a través de reportes o archivos que viajan vía batch. Para mejorar los tiempos de transferencia y comunicación con dichos agentes sería crucial desarrollar una api y disponibilizarla para que ellos la consulten

3.2 Reflexión sobre las prácticas profesionales como espacio de formación:

La Práctica Profesional Supervisada fue una experiencia sumamente útil para poder aplicar gran parte de los conocimientos aprendidos en la carrera universitaria. Tuve la posibilidad de desarrollar la misma en mi puesto de trabajo actual, interrelacionándome con distintas personas y departamentos llevando a cabo el rumbo del proyecto. Aquí uno aprende a lidiar ciertas cuestiones fuera del ámbito técnico, como lo son la negociación con personas, estimaciones y demás, lo cual me permitió desarrollar las habilidades blandas o soft skills que complementan al contenido técnico ya adquirido.

A su vez, también me di cuenta de que muchos supuestos teóricos no se aplican al pie de la letra en la vida real, debido a que, en general, las acciones de las personas con las cuales se trabaja están condicionadas por las fechas de entrega que hay que cumplir y se trata de hacer todo lo más rápido posible.

En cuanto al desarrollo, al comienzo se me presentaron algunas dificultades para esquematizar las distintas etapas o fases en las que se tenía que dividir el contenido de lo realizado, pero con la ayuda de algunos colegas de la facultad, como también de mis tutores asignados -los cuales siempre estuvieron predispuestos para conmigo en todo momento-, logré comprender cómo disponer de la información y cómo abordarla para conseguir los objetivos del proyecto. Durante el transcurso del mismo, también pude incrementar mi conocimiento funcional en el área y profesionalismo al conocer a profundidad los distintos sectores que trabajan en el banco y su modo de operar, como por ej M&A (sector encargado de enviar cualquier notificación a clientes), Marketing, Legales, etc. En conclusión, la PPS aportó un gran valor agregado a mi desarrollo personal y profesional, me permitió hacerme cargo de un desarrollo de punta a punta, es decir, desde el relevamiento hasta la implementación en ambiente productivo.

4. ANEXO

4.1 GLOSARIO

Cursor: Contenedor virtual que almacena conjuntos de datos devueltos por una instrucción de tipo select

Select: Instrucción que recupera filas/información de una base de datos

Where: Palabra que viene a complementar a la instrucción select especificando los criterios que tienen que cumplir los registros para ser recuperados o seleccionados

Union: Sentencia que se utiliza para concatenar/adjuntar los resultados de dos o más sentencias select

Truncate: Comando que se utiliza para eliminar el contenido de datos que posee una tabla de base de datos

Max(): Función que sirve para obtener el mayor valor para un dato cuantificable determinado

Group By: Sentencia que se utiliza para agrupar resultados en base a una columna de una tabla seleccionada

Minus: Sentencia que hace la resta entre dos conjuntos de resultados. Devuelve los registros del primer conjunto de resultado que no estén presentes en el segundo

Loop: Término informático que hace referencia a un proceso en forma de bucle el cual comienza donde termina y viceversa.

Min(): Función que sirve para obtener el menor valor para un dato cuantificable determinado

SYS_GUID(): Función que devuelve un identificador único global de 32 bytes

SYSDATE(): Función que devuelve la fecha y hora actual establecidas para el sistema operativo en el que reside la base de datos

Front-end: Parte gráfica de un sitio web que interactúa con los usuarios

DevOps: Enfoque de software que puede adoptar una organización en donde el programador no solo programa sino que también realiza las tareas específicas para impactar sus cambios en otros ambientes como beta o producción

Merge Request: Acción en la cual se solicita la integración, de los códigos fuentes/ejecutables subidos en una rama, hacia otra rama de mayor nivel

4.2 Bibliografía:

Domínguez, Pablo (s.f). "En qué consiste el modelo en cascada". Disponible en:
<<https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>>

Ortiz Ornelas, Francisco (2013). "Cómo configurar un Web Application Server". Disponible en:
<<https://www.ibm.com/developerworks/ssa/websphere/configurawas/index.html>>

Vericat, Alex (2019). "Qué es Bash Scripting". Disponible en:
<<https://openwebinars.net/blog/que-es-bash-scripting/>>

EcuRed (2016). "Requisitos de Software". Disponible en:
<[https://www.ecured.cu/Requisitos de Software](https://www.ecured.cu/Requisitos_de_Software)>

Alvarez, Isaac (2018). "¿Por qué un test de Regresión?". Disponible en:
<<https://qanewsblog.com/2018/01/22/por-que-un-test-de-regresion/>>

Marsá, Sandra (2015). "Descubriendo GitLab". Disponible en:
<<https://inlab.fib.upc.edu/es/blog/descubriendo-gitlab>>

Banco Santander S.A (2020). "¿Qué es Superclub?". Disponible en:
<<https://www.santander.com.ar/banco/online/personas/superclub>>