



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Proyecto de automatización  
de reembolsos de tickets  
aéreos**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de

**Ingeniero Industrial**

**P R E S E N T A**

Christopher Kossal Wilson

**ASESOR DE INFORME**

Ing. Omar Alejandro Llanes Briceño



Ciudad Universitaria, Cd. Mx., 2023

# Agradecimientos

Es difícil empezar esta sección porque ha habido tantas personas que me han ayudado y apoyado en este proceso de educación y desarrollo, sería imposible mencionarlos a todos, pero tengan por seguro que los recuerdo a todos con mucho cariño y agradecimiento, sin ustedes esto no sería posible y solo puedo desear devolver un cachito de lo que me han dado a lo largo de este viaje. Dicho esto, quiero poner especial énfasis en algunas personas:

A mis padres, por darme todas las oportunidades, cariño y apoyo durante todos estos años. Nunca me ha faltado nada y nunca me faltará gracias a ustedes. Gracias por inculcarme los valores de la familia, el trabajo y la solidaridad.

A Omar, por todo lo que me has enseñado, ayudado y toda la amistad que me has dado sin pedir nada a cambio. Espero poder crecer en alguien que pueda dar de vuelta al mundo tanto como tú lo has dado. Gracias por toda la paciencia y apoyo para completar este último paso.

A Bruno, por tu insistencia férrea en titularme por experiencia profesional, esta va para ti bro.

A Alex, por darme la oportunidad de crecer junto con tu proyecto, siempre apostando en mí. Espero poder formar a la siguiente generación de profesionales tan bien como tú me ayudaste a formarme.

# Índice

<i>Agradecimientos</i> .....	2
<i>Introducción y Objetivos</i> .....	5
<b>Capítulo I: Antecedentes y contexto de la empresa</b> .....	7
<b>Descripción de la empresa:</b> .....	7
<b>Descripción del portafolio de productos:</b> .....	8
<b>Automated Refund Platform:</b> .....	9
1. Recibo de solicitud .....	9
2. Cotización de solicitud .....	10
3. Validación de cotización y solicitud de reembolso .....	14
4. Confirmación de reembolso y transferencia de fondos .....	15
<b>Fuente: Elaboración propia</b> .....	17
<b>Descripción de los clientes:</b> .....	17
<b>Descripción de la estructura organizacional:</b> .....	19
<b>Capítulo II: Descripción de proyecto</b> .....	22
<b>Descripción general</b> .....	22
<b>Etapa I: Alinear alcance y etapas</b> .....	24
Metodologías de desarrollo de software .....	25
Diseño de Metodología .....	27
<b>Etapa II: Validar y conseguir accesos</b> .....	33
<b>Etapa III: Diseñar métricas de éxito y pruebas manuales</b> .....	36
<b>Etapa IV: Pruebas automatizadas y despliegue en producción</b> .....	40
<b>Capítulo III: Análisis de resultados</b> .....	42
<b>Resultados financieros</b> .....	42
<b>Resultados de calidad</b> .....	43
<b>Resultados de la metodología</b> .....	45
<b>Conclusión</b> .....	48
<b>Bibliografía</b> .....	49

Imagen 1. Ejemplo de cómo se visualiza una reserva en la terminal crítica de Amadeus. ....	10
Imagen 2. Ejemplo de cómo se visualiza un ticket en Amadeus .....	11
Imagen 3. Ejemplo de regla tarifaria sección 16 desplegado en Sabre .....	13
Imagen 4. Impacto neto de la integración con Despegar .....	43
Diagrama 1. Flujo ARP de alto nivel.....	17
Diagrama 2. Organigrama durante proyecto de Despegar .....	19
Tabla 1. Etapas de integración ARP con Despegar.....	24

# Introducción y Objetivos

La industria de viajes y turismo siguen siendo una de las mayores industrias del mundo y un gran motor de crecimiento, aportando 9.5 billones<sup>1</sup> de USD al PIB global en 2023. Dominado por grandes jugadores como las aerolíneas y grandes agencias de viajes (Expedia, Booking), todos dependen mayoritariamente de 3 GDS<sup>2</sup> (Global Distribution System). Principales responsables de la digitalización y automatización<sup>3</sup> de la compra y administración de reservas aéreas a través su sistema CRS, este sistema fue completamente innovador en los años 70, sin embargo, ha enfrentado retos de obsolescencia<sup>4</sup> desde la entrada del Internet en los años 90.

Actualmente el sistema CRS sigue siendo el núcleo de la tecnología de los GDSs, lo que presenta muchos retos a los usuarios (aerolíneas y agencias de viajes) ya que dificulta y encarece la automatización de procesos postventa como el cambio y reembolso de tickets aéreos. Deal Engine<sup>5</sup> nace en 2017 como una empresa que busca automatizar procesos de la industria de viajes, particularmente la administración de reembolsos a través de las conexiones con GDS.

Este reporte busca describir el proyecto de integración de la herramienta de automatización de reembolsos en Despegar.com, la mayor agencia de viajes online en Latinoamérica<sup>6</sup>, el cual lideré mientras servía como director de operaciones en Deal Engine, con el objetivo de reducir los procesos operativos en Despegar.

---

<sup>1</sup> *Travel and Tourism: Contribution to Global GDP 2023* | Statista. (2023, 7 noviembre). Statista.

<https://www.statista.com/statistics/233223/travel-and-tourism--total-economic-contribution-worldwide/>

<sup>2</sup> Fernández-Paniagua, A. M. del C. (2017). *¿Qué son los Softwares GDS y para qué sirven?* Thinking for Innovation. <https://www.iebschool.com/blog/software-gds-marketing-digital/>

<sup>3</sup> Editor. (2019, 12 abril). History of flight booking: CRSs, GDS distribution, travel agencies, and online reservations. *AltexSoft*. <https://www.altexsoft.com/blog/history-of-flight-booking-crss-gds-distribution-travel-agencies-and-online-reservations/>

<sup>4</sup> Raymond, L., & Bergeron, F. (1997). Global distribution systems: A field study of their use and advantages in travel agencies. *Journal of Global Information Management (JGIM)*, 5(4), 23-32.

<sup>5</sup> Jara, A. (n.d.). *Digital transformation for the travel industry*. Deal Engine. <https://deal-engine.com/>

<sup>6</sup> Statista. (2022, 20 septiembre). *Latin America: Market share of main online travel agencies 2019*. <https://www.statista.com/statistics/1235726/latin-america-leading-online-travel-platforms/>

Primero describiré los antecedentes y el contexto de Deal Engine dentro de la industria de viajes, después expondré los detalles del producto de automatización de reembolsos, seguido por la descripción del proyecto y finalizaré con el análisis de resultados.

A lo largo de este informe, describiré detalladamente las responsabilidades y logros clave que he alcanzado en mi rol, destacando cómo mi experiencia se relaciona directamente con los atributos de egreso, perfil de egreso, perfil profesional y competencias del programa de Ingeniería Industrial que he cursado. Además, demostraré cómo mi trabajo ha contribuido al crecimiento y la eficiencia de Deal Engine, así como mi propio crecimiento profesional.

# Capítulo I: Antecedentes y contexto de la empresa

En este capítulo, proporcionaré una visión integral del entorno laboral en el que se llevó a cabo la experiencia profesional. Comenzaré por presentar una descripción general de la empresa en la que tuve el privilegio de trabajar, destacando su giro empresarial, su misión y visión corporativa. A continuación, profundizaré en la descripción de los objetivos y actividades del departamento o área específica en la que me desempeñé como egresado.

## Descripción de la empresa:

La empresa Deal Engine pertenece al sector de Tecnología, específicamente desarrolla Software desplegado en la nube (también conocido como SaaS<sup>7</sup> o Software as a Service) para agencias de viajes y aerolíneas dentro de la industria de viajes. Sus productos buscan automatizar procesos manuales de posventa que de otra manera hubieran sido ejecutados por un equipo de agentes de viaje o consultores externos. Sus principales clientes son Despegar, CVC, Latam Airlines y Gol Airlines.

La empresa se denomina a sí mismo como un startup<sup>8</sup>, es decir, nace con la intención de crear un producto que no tiene un modelo de negocios desarrollado y no está validado con el mercado, con el objetivo de experimentar y encontrar un nicho de mercado. Dado el alto riesgo que representa, todo el capital es financiado por capital de riesgo a través de rondas de inversión. Esta empresa inició operaciones a finales de 2017 y se financió a través de una ronda de inversión inicial de \$ 500,000 USD, el objetivo inicial era capitalizar la volatilidad en los precios de los tickets aéreos dentro de agencias de viajes corporativas. La idea era ofrecer un software que detectara en tiempo real ofertas con precios más bajos de reservas ya

---

<sup>7</sup> *What is Software as a Service (SaaS)?* (n.d.). Salesforce.com. <https://www.salesforce.com/mx/saas/>

<sup>8</sup> Grant, Mi. (2021, September 15). *What You Should Know About Startups*. Investopedia. <https://www.investopedia.com/terms/s/startup.asp>

realizadas por una agencia de viajes. Si lo encontraba, se realizaría el cambio automáticamente y se repartiría el ahorro entre el pasajero, la agencia y Deal Engine.

A finales de 2022 se consiguió levantar una ronda de \$ 5,000,000 USD<sup>9</sup> con el objetivo de crecer el equipo, financiar el desarrollo de otros productos de automatización y expandir a nivel global. Actualmente, la empresa tiene 60 empleados de tiempo completo, presencia en 17 países y 3 productos principales.

La misión de la empresa es liderar procesos de transformación digital a la industria de viajes, usando tecnologías de comunicación robóticas e inteligencia artificial propietarias para eliminar procesos manuales, buscando que los viajeros usen su tiempo en actividades de valor. La visión de la empresa es volverse la empresa de tecnología en transacciones de venta y posventa más grande de la industria de viajes.

## Descripción del portafolio de productos:

A continuación, describiré brevemente sus 3 productos principales:

1. **EPS:** Expiry Protection Service o Servicio de Protección contra la Expiración es un servicio para agencias de viajes que escanea su portafolio de reservas vendidas para encontrar aquellas donde los pasajeros no tomaron sus vuelos, con el objetivo de solicitar el reembolso de estas reservas a la aerolínea 1 día antes de que expiraran. Este servicio funciona porque, aunque el pasajero no haya tomado su vuelo, la aerolínea debe devolver los impuestos de uso aeroportuario que no se hayan usado, y la agencia de viajes tiene generalmente 1 año para solicitar este reembolso. Este servicio genera un ingreso adicional a la agencia de viajes, por lo que se cobra un porcentaje de comisión por el monto de reembolso total que se genera mensualmente.
2. **ARP:** Automated Refund Platform o Plataforma Automatizada de Reembolsos es un servicio que ayuda a agencias de viajes y aerolíneas a cotizar y solicitar

---

<sup>9</sup>Dawes, J. (2022, November 18). *Deal Engine Raises \$5.3 Million to Help Airlines Save Post-Booking*. Skift. <https://skift.com/2022/11/18/deal-engine-raises-5-3-million-to-help-airlines-save-post-booking-travel-startup-funding-this-week/>



automáticamente los reembolsos de tickets aéreos. Este servicio es muy importante para ellos, ya que reduce la carga operativa y disminuye el tiempo de atención al cliente, además de ofrecer un sistema automatizado que se puede integrar en la página web o aplicación celular del cliente. Se cobra una tarifa de transacción por reembolso cotizado y solicitado.

3. **ACP:** Automated Change Platform o Plataforma Automatizada de Cambios es un servicio que ayuda a agencias de viajes y aerolíneas a cotizar y solicitar automáticamente los cambios de tickets aéreos. Este servicio tiene las mismas ventajas que ARP, con la diferencia de que hay 10 veces más cambios que reembolsos, por lo que es más atractivo para los clientes, pero por lo mismo, es común que el cliente ya haya desarrollado una solución interna o haya más competencia externa.

## Automated Refund Platform:

Este reporte describe un proyecto específico de ARP, por lo que vale la pena ahondar. ARP es un producto SaaS que busca automatizar el cálculo, solicitud y seguimiento de un reembolso aéreo. Para entender cómo funciona el producto, describiré como es el proceso de solicitud de un reembolso para una reserva aérea en 4 pasos desde la perspectiva de una agencia de viajes (el proceso es muy similar para una aerolínea):

1. Recibo de solicitud
2. Cotización de solicitud
3. Validación de cotización y solicitud de reembolso
4. Confirmación de reembolso y transferencia de fondos

### *1. Recibo de solicitud*

Un pasajero contacta a la agencia informando que quiere reembolsar una reserva. Este contacto ocurre por email, teléfono, formulario página web o en aplicación móvil y suele ser respondido por un equipo de atención de cliente de la agencia. Para identificar la reserva, el

pasajero comparte el código PNR<sup>10</sup> (Passenger Number Record) o el apellido, la agencia de viajes usa una terminal críptica de GDS, un programa desarrollado internamente o externo para interactuar con el sistema GDS (posteriormente referido como interfaz GDS) y encontrar la reserva. También tiene la habilidad de buscar el código PNR a través del apellido<sup>11</sup>. La reserva contiene todos los datos del itinerario y también puede contener múltiples pasajeros, estos pueden ser los miembros de una familia, por lo que es importante que el pasajero que contactó la agencia especifique que pasajeros desea reembolsar. Una vez la agencia haya identificado la reserva y los pasajeros a reembolsar, le informará al pasajero que se pondrán en contacto cuando hayan cotizado el monto de reembolso. Dependiendo del tamaño de la agencia, atención al cliente cotizará el reembolso o solicitará la cotización a un equipo especializado en reembolsos.

```

--- RLR ---
RP/MNLIA0980/MNLIA0980          FM/SU 12OCT21/0736Z  PPZ608
MNLIA0980/0423EO/12OCT21
1.AMADEUS/PNR MR
2  6X 968 Y 10DEC 5 HKGLHR HK1 0100 0620 10DEC E 6X/PPZ608
3  6X3916 Y 10DEC 5 LHRJFK HK1 1000 1200 10DEC E 6X/PPZ608
4  6X3629 Y 20DEC 1 JFKLHR HK1 1100 2150 20DEC E 6X/PPZ608
5  6X5915 Y 20DEC 1 LHRHKG HK1 2350 1945 21DEC E 6X/PPZ608
6 AP MNL 8899003 - ABC CORP - A
7 AP MNL 123459899 - H
8 APE FAXMAIL@YYY.COM
9 TK OK12OCT/MNLIA0980
10 SSR NSST 6X HK1 LHRJFK/04C,P1/S3 SEE RTSTR
11 SSR NSST 6X HK1 JFKLHR/18D,P1/S4 SEE RTSTR
12 SSR NSST 6X HK1 LHRHKG/29C,P1/S5 SEE RTSTR
13 SSR SFML 6X HNL/S2
14 SSR SFML 6X HNL/S3
15 SSR SFML 6X HNL/S4
16 SSR SFML 6X HNL/S5
17 OPW-13OCT:1600/1C7/6X REQUIRES TICKET ON OR BEFORE
    14OCT:1600/S2-5
18 OPC-14OCT:1600/1C8/6X CANCELLATION DUE TO NO TICKET/S2-5
19 RC MNLIA0980-W/FF56EIF9W3NQ2JWQNECJD09HJGR/69HHXMOLCDEFOEKXW

```

Imagen 1. Ejemplo de cómo se visualiza una reserva en la terminal críptica de Amadeus.

Fuente: Amadeus Service Hub<sup>12</sup>

## 2. Cotización de solicitud

El equipo encargado de la cotización de reembolsos usa la interfaz GDS para encontrar los datos de la reserva, si no lo encuentra puede ser debido a que el GDS haya purgado la reserva,

<sup>10</sup> *Connect to Amadeus Travel APIs | Amadeus for Developers.* (s. f.). Amadeus IT Group SA.

<https://developers.amadeus.com/blog/what-is-pnr-booking-reference>

<sup>11</sup> *How to retrieve/display a PNR (Cryptic) - Amadeus Service Hub.* (s. f.).

<https://servicehub.amadeus.com/c/portal/view-solution/453392470/como-recuperar-ver-un-pnr-criptico-#>

<sup>12</sup> *Amadeus Customer Profiles: How to create a profile from a PNR (Cryptic) - Amadeus Service Hub.* (s. f.).

<https://servicehub.amadeus.com/c/portal/view-solution/838421/perfiles-de-clientes-de-amadeus-como-crear-un-perfil-a-partir-de-un-pnr-criptico->

lo cual suele hacer 10 días después del último vuelo, en ese caso deberán usar otro método para encontrar la reserva histórica<sup>13</sup>. Una vez encontrado deben revisar el historial de la reserva para revisar si la aerolínea ocasionó alguna cancelación, retraso o adelanto que pudiera darle derecho al pasajero a un reembolso completo. Luego buscan los códigos de ticket que corresponden a los pasajeros a reembolsar. Este código es único por pasajero y sirve para buscar la información del itinerario, tarifas pagadas, reglamento e impuestos pagados. En la mayoría de los casos la información de estos tickets es equivalente, pero puede cambiar cuando alguno de los pasajeros es un infante o un niño, ya que suelen pagar tarifas inferiores.

```

TKT-2205913021086      RCI-          1A LOC-2ZEULH
OD-BKKBKK SI-      FCMI-0  POI-BKK  DOI-28JUN12  IOI-35303354
  1.WAHLBERG/MARK MR      ADT      ST
1 OBKK LH 0783 D 01OCT 2355 OK DRFTH      0      2PC
2 XFRA LH 0172 D 02OCT 0735 OK DRFTH      0      2PC
3 OTXL LH 0197 D 12NOV 1900 OK DRFTH      0      2PC
4 XFRA LH 0772 D 12NOV 2215 OK DRFTH      0      2PC
  BKK
FARE   F THB      125000
TOTALTAX THB      19790
TOTAL   THB      144790
/FC BKK LH X/FRA LH BER M2038.35LH X/FRA LH BKK M2038.35NUC4076.
70END ROE30.662000
FE FL/CNX/CHG RESTRICTED CHECK FARE NOTE
FP CASH
FOR TAX DETAILS USE TWD/TAX

```

Imagen 2. Ejemplo de cómo se visualiza un ticket en Amadeus

Fuente: Amadeus Service Hub<sup>14</sup>

<sup>13</sup> How to display the PNR History (CRYPTIC) - Amadeus Service Hub. (s. f.).

<https://servicehub.amadeus.com/c/portal/view-solution/813131/como-mostrar-el-historial-pnr-criptico->

<sup>14</sup> Price, S. A. (s. f.). Amadeus Central Ticketing Reissue/Exchange. ppt download.

<https://slideplayer.com/slide/16156344/>

Una vez identificado los tickets, se despliega la información de los cupones<sup>15</sup> (equivalente a un vuelo) y se revisa la construcción tarifaria<sup>16</sup>, este es un texto que describe cómo el GDS calculó la tarifa, ahora se muestra un ejemplo básico:

**SFO AM MEX606.50 AM LAX AM TYO AM HKG606.50 USD1363.00END**

- **SFO 7X MEX606.50** hace referencia al primer segmento tarifario, es una intención de vuelo que puede contener uno o más cupones. En este caso, es un segmento que consta de 1 único vuelo de San Francisco (código IATA SFO) hasta Ciudad de México (código IATA MEX) usando la aerolínea Aeroméxico (código ICAO AM) que debe costar 606.50 de tarifa base (mas adelante veremos que moneda usa)
- **AM LAX AM TYO AM HKG606.50** es el segundo segmento tarifario, cada segmento se separa usando el monto de tarifa base. Consta de 2 vuelos, el primero es de Ciudad de México (destinación del anterior segmento tarifario) a Los Ángeles (código IATA LAX) la cual es una escala hacia Hong Kong (código IATA HKG), todo volado a través de Aeroméxico y costando 606.50 de tarifa base
- **USD1363.00END** nos dice que la moneda de las tarifas bases es dólares (USD) y que la tarifa total a pagar es de 1363 USD que equivale a la suma de las tarifas bases. END indica que ya acabó de describir los segmentos tarifarios, aunque podría haber información a la derecha que describe montos como cambios de moneda, impuestos u otros indicadores.

La construcción tarifaria es vital ya que las reglas de reembolsabilidad aplican por segmento y no por vuelo, adicionalmente solo se pueden reembolsar las tarifas de segmentos tarifarios que no estén volados por completo, lo cual nos lleva al siguiente paso. Se verifica el estatus de los cupones (vuelos) para ver que cupones tienen un estatus abierto (el pasajero no voló ese vuelo). También se debe buscar la regla tarifaria usando el comando críptico, esta regla

---

<sup>15</sup> *How to display an e-ticket record (Cryptic) - Amadeus Service Hub.* (s. f.).

<https://servicehub.amadeus.com/c/portal/view-solution/939515/como-mostrar-un-registro-de-boleto-electronico-cryptic->

<sup>16</sup> *Fare calculation symbols and explanations - Amadeus Service Hub.* (s. f.).

<https://servicehub.amadeus.com/c/portal/view-solution/774026/fare-calculation-symbols-and-explanations>

tiene múltiples secciones<sup>17</sup>, pero para el reembolso solo se usa la sección 16 (penalizaciones), ahora se comparte un ejemplo:

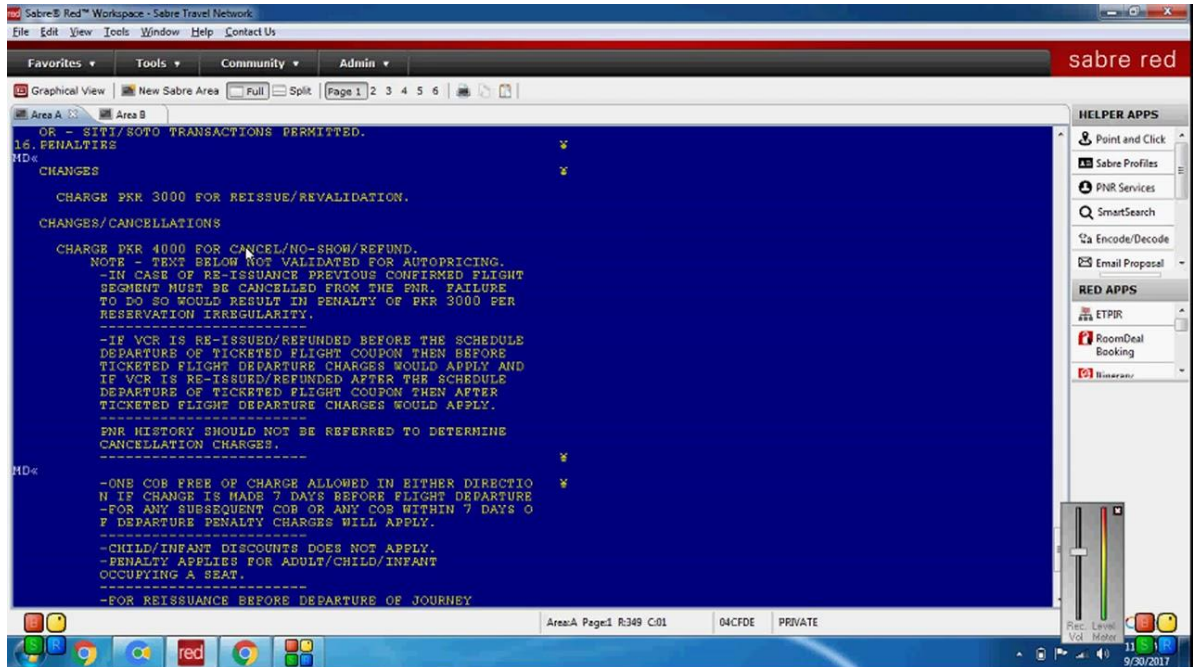


Imagen 3. Ejemplo de regla tarifaria sección 16 desplegado en Sabre

Fuente: Sabre RED (interfaz críptica)<sup>18</sup>

Para cada segmento tarifario habrá una regla de cancelaciones que indicará si la tarifa base es reembolsable. Dicho esto, lo normal es que la aerolínea pida que se use la regla más restrictiva para todo el ticket, es decir, si un segmento es reembolsable y otro no, se asume que todo el ticket es no reembolsable.

Teniendo la información de los segmentos tarifarios, estatus de cupones y regla tarifaria el agente ya puede cotizar el monto a reembolsar por tarifa base. Ahora solo falta un cálculo más.

<sup>17</sup> Fare Rules: Category description - Amadeus Service Hub. (s. f.).

<https://servicehub.amadeus.com/c/portal/view-solution/926104/condiciones-de-tarifa-descripcion-de-la-categoria>

<sup>18</sup> Ticketing System. (2018, 25 junio). Sabre #8 | HOW TO CHECK PENALTY REFUND TICKET AND NOSHOW PENALTY ON SABER GDS ABECUS [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=MhOahB6R8Ck>

A parte de tarifa, el pasajero también paga impuestos, Deal Engine generó 4 categorías para describirlas dependiendo de su reembolsabilidad:

1. SALE: impuestos derivados de la venta del segmento (como el IVA) y que pueden ser porcentuales o de monto fijo. Se reembolsan solo si el segmento al que corresponden es reembolsable.
2. FUEL: generalmente son cargos de combustible que se reportan como impuestos a pagar. Solo son reembolsables si todos los segmentos son reembolsables.
3. USE: impuestos por uso tarifario de un aeropuerto, siempre son reembolsables si no se pisó el aeropuerto.
4. NEVER: impuestos que por reglamento nunca pueden ser reembolsados, o porque son penalidades de cambio que se reportan como impuestos.

Sumando el monto de tarifa e impuestos a reembolsar el agente ahora puede compartir el monto con el pasajero de la reserva.

### *3. Validación de cotización y solicitud de reembolso*

El agente comparte al pasajero la cotización final de reembolso por correo, mensaje, notificación web o por llamada. En caso de que el pasajero esté de acuerdo se procede con la solicitud del reembolso a la aerolínea y en Deal Engine se categorizó las vías de reembolso en 2:

- Settlement Systems: son sistemas contables que ayudan a administrar las transacciones monetarias entre las aerolíneas y las agencias de viaje. Los agentes interactúan con estos sistemas a través de una interfaz web y sus credenciales de usuario. Permiten obtener información como el resumen de ventas de tickets, anciliarios, multas u otros (como reembolsos). Una de las funciones que tienen es un formulario que permite solicitar el reembolso. Existen 2 sistemas contables (ARC<sup>19</sup> y BSPLink<sup>20</sup>) que coinciden en esencia

---

<sup>19</sup> *Our story*. (s. f.). [https://www2.arccorp.com/about-us/our-story/?utm\\_source=Global\\_Navigation](https://www2.arccorp.com/about-us/our-story/?utm_source=Global_Navigation)

<sup>20</sup> *Billing and Settlement Plan (BSP)*. (s. f.). <https://www.iata.org/en/services/finance/bsp/>

con las mismas funcionalidades, pero ARC se usa en Estados Unidos y BSPLink en el resto del mundo.

- GDS: a través de la terminal crítica los agentes también tienen la funcionalidad para solicitar reembolsos<sup>21</sup>.

El agente debe solicitar el reembolso en una de estas vías dependiendo de las políticas de las aerolíneas que son sujetas a cambios dependiendo de su estrategia, lo que nos lleva al siguiente paso:

#### *4. Confirmación de reembolso y transferencia de fondos*

La confirmación del reembolso es distinta en cada vía:

- Settlement Systems: las solicitudes de reembolso deben ser revisados manualmente por un agente de la aerolínea, lo que le da la seguridad a la aerolínea de que el cálculo del reembolso es correcto y minimiza su movimiento de caja. En caso de que el cálculo de la solicitud sea incorrecto, lo rechazan indicando la razón. Los agentes de viaje revisan periódicamente el sistema para corregir las solicitudes rechazadas y notificar a los pasajeros donde su solicitud se autorizó. El desembolso ocurre durante el cierre contable del periodo donde se autoriza el reembolso (generalmente ocurre cada 2 semanas entre aerolíneas y agencias).
- GDS: una vez completado la solicitud, el sistema autoriza el desembolso automáticamente y dispone de los fondos en el cierre contable. Este método es el preferido por aerolíneas y agencias ya que reduce el trabajo operativo y tiempo de respuesta al pasajero, pero no todas las aerolíneas lo permiten, y las que lo hacen suelen excluir ciertas regiones. Esto se debe a que la aerolínea requiere cierto nivel de automatización necesario para manejar y contabilizar estos reembolsos automáticos, y deben confiar en la disciplina de sus agencias. Todos los reembolsos automáticos se auditan por la aerolínea entre 3 a 4 meses después del reembolso, en caso de detectar un reembolso incorrecto, emiten un ADM (esencialmente una multa) donde la agencia

---

<sup>21</sup> *How to refund a ticket (Cryptic) - Amadeus Service Hub.* (s. f.).  
<https://servicehub.amadeus.com/c/portal/view-solution/961528/como-reembolsar-un-billete-criptico->

deberá pagar la diferencia entre su cálculo y el cálculo correcto, mas un monto adicional de penalidad.

Llegados a este punto, el pasajero debió de recibir su reembolso, al menos ser notificado de porqué su reembolso no se puede realizar. Dependiendo del tiempo de respuesta de la agencia y de la aerolínea, este proceso puede llevar desde tres días hasta meses (o años en algunos casos extremos), lo que suele causar una mala experiencia en el pasajero y las agencias reconocen como un punto negro en su *customer journey* (recorrido del cliente). Adicionalmente, es un proceso manual que le lleva al agente entre 5 a 30 minutos completar dependiendo de la complejidad del reembolso, por lo es costoso la mano de obra requerida para las agencias.

Dentro de Deal Engine calculamos que, en promedio, entre el 2% al 6% de los tickets emitidos por una agencia de viajes o aerolínea se solicitan para reembolso por los pasajeros. Dada la complejidad en la descarga e interpretación de las reglas tarifarias y de impuestos, es extremadamente poco conveniente para el pasajero calcular y solicitar su reembolso. Por esta razón, son las propias agencias de viajes y aerolíneas las que lo gestionan, forzándolos a tener mucho personal responsable de recibir las solicitudes de los pasajeros, calcular los reembolsos, conseguir la confirmación del pasajero, solicitar y hacer seguimiento al reembolso.

Por ello nace ARP, un servicio de software que automatiza todas las interacciones que normalmente el agente debe hacer con los GDS y los Settlement Systems. Es decir, es capaz de consultar la información de la reserva, el ticket, sus segmentos tarifarios, reglas tarifarias, impuestos para interpretarlos y determinar el monto exacto a reembolsar. Adicionalmente, puede solicitar el reembolso a través de las 2 vías, y en el caso de los Settlement Systems, hacer el seguimiento hasta que la aerolínea lo autorice o rechace. Los clientes (agencias de viajes o aerolíneas) pueden usarlo a través de una interfaz web desplegado por Deal Engine, lo que simplifica el proceso de reembolso de los agentes a la comunicación con el pasajero, solicitud, validación y solicitud de reembolso a través de la interfaz de Deal Engine:



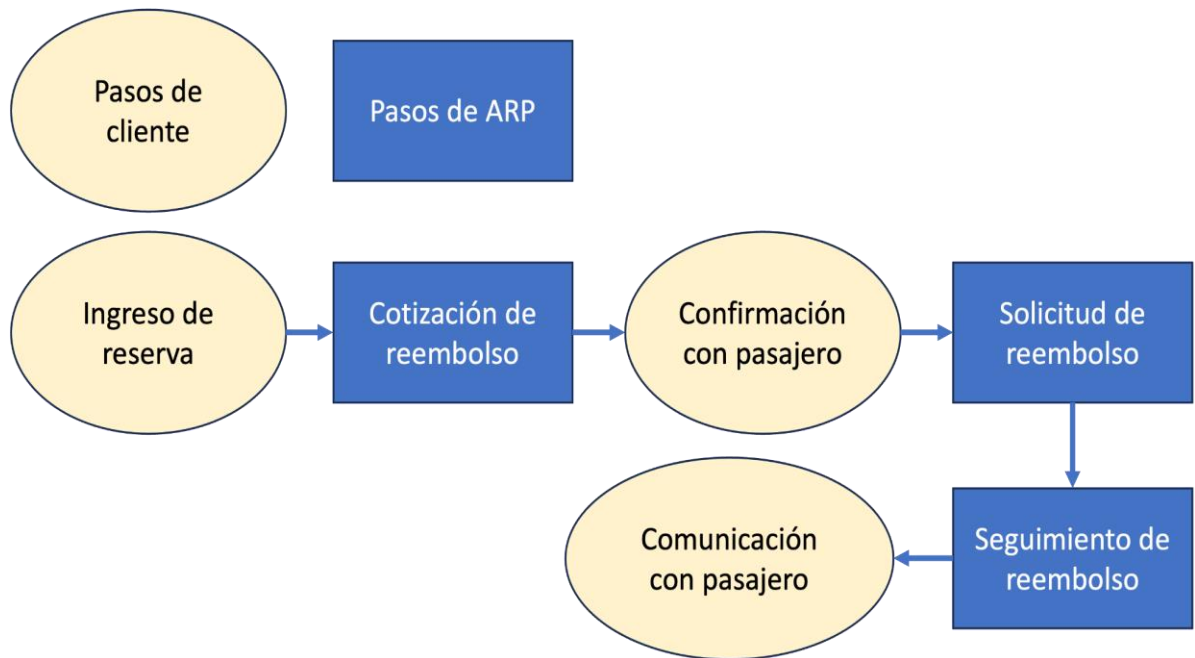


Diagrama 1. Flujo ARP de alto nivel

Fuente: Elaboración propia

## Descripción de los clientes:

Deal Engine trabaja con clientes específicos en la modalidad B2B<sup>22</sup> (Business to Business o Empresa a Empresa). Estas debían ser agencias de viajes o aerolíneas que emitieran al menos 250,000 tickets al año, cada ticket representa el viaje de un pasajero y debían estar contenidos en una reserva la cual podría contener a varios tickets. Este volumen garantiza que el caso de negocio de automatizar sus procesos fuera beneficiosos y estuviera dispuesto a contratar los servicios de Deal Engine.

Dentro de las agencias de viajes Deal Engine distingue 3 categorías:

1. **OTA** (Online Travel Agency o Agencia de Viajes Online): se caracterizan por vender mayoritariamente a través de canales digitales como páginas webs y aplicaciones

<sup>22</sup> Chen, J. (2022, November 3). *Business to Business (B2B)*. Investopedia. <https://www.investopedia.com/terms/b/btob.asp>

móviles, primando el bajo precio y cantidad de ofertas. Sus clientes principales son turistas.

2. **TA** (Travel Agency o Agencia de Viajes): parecido a una OTA, pero su canal de ventas suele ser telefónico y a través de sucursales, primando la calidad del servicio a sus clientes. Sus clientes principales suelen ser turistas de alto poder económico y viajeros corporativos.
3. **TMC** (Travel Management Company o Compañía de Gestión de Viajes): son agencias que sirven directamente a empresas y organizaciones para gestionar sus necesidades de viajes corporativos.

Deal Engine sirve a los 3 tipos de agencias de viaje, prefiriendo a las TAs y TMCs para EPS, y OTAs para ARP/ACP.

Dentro de las aerolíneas Deal Engine distingue 2 categorías:

1. **Legacy**: son aerolíneas que usan GDS para hacer disponible su oferta de vuelos, lo que suele encarecer sus precios, por lo que ofrecen mejor calidad de servicio. Ejemplos son Aeroméxico, Delta o American Airlines.
2. **Budget**: son aerolíneas que destacan por tener los menores precios, su venta ocurre exclusivamente a través de sus canales de venta para evitar pagar tarifas de gestión por terceros.

Deal Engine solo trabaja con aerolíneas Legacy, ya que al usar GDS, sus transacciones como la venta, cambios o reembolsos están estandarización dentro de los servicios de los globalizadores. Esto significa que no es necesario integrar los servicios de cada aerolínea, sino que basta con automatizar los servicios de los GDS para poder servir a casi todas las aerolíneas de la industria. Adicionalmente, la mayoría de las agencias de viajes usan aerolíneas Legacy, ya que tienen contratos de objetivos de ventas, por lo que no les conviene usar Budget airlines.

## Descripción de la estructura organizacional:

Durante el proyecto la empresa inició con 3 áreas, las cuales eran Ventas, Operaciones de Negocio e Ingeniería. Yo trabajé como Director de Operaciones desde febrero 2021 hasta agosto 2023 en el área de Operaciones de Negocio y reportaba al Director General.

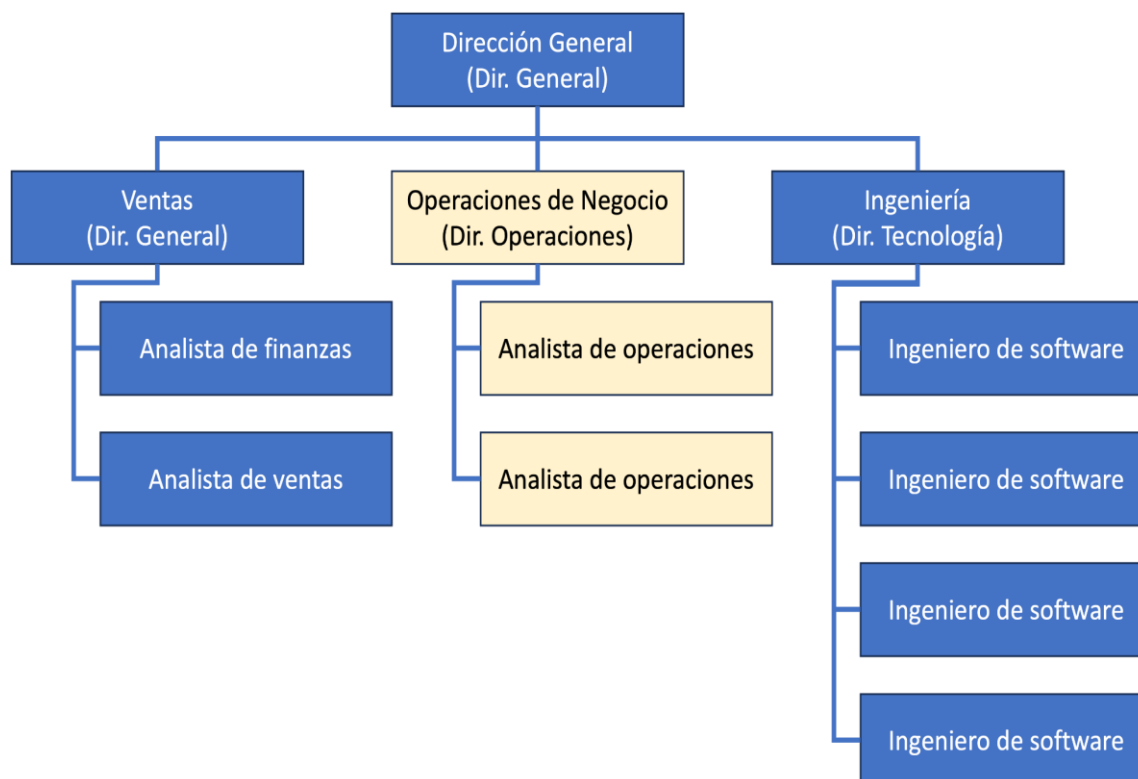


Diagrama 2. Organigrama durante proyecto de Despegar

Fuente: Elaboración propia

El área de Operaciones tenía como objetivo liderar proyectos de integración de los productos en nuevos clientes, aumentar las métricas que impactan los ingresos y gestionar los casos de atención del cliente. Ejemplos de estas acciones son:

1. **Proyectos de integración:** para hacer uso de los productos de Deal Engine, es necesario que los clientes compartan las credenciales de aquellos servicios que necesitan automatizar, por ejemplo, se necesita acceso al sistema de contabilidad de los clientes, el cual permite a la herramienta gestionar automáticamente transacciones como reembolsos o descarga de ventas para análisis. Para ello, el área de Operaciones

debía educar y solicitar estos accesos a los contactos correspondientes en los clientes. Adicionalmente, debían tomar en cuenta los procesos y necesidades específicas de los clientes para gestionar cambios en los productos que cumplieran con sus especificaciones.

2. **Métricas de ingresos:** para poder cobrar los servicios de Deal Engine se debía cumplir con una serie de métricas. Un ejemplo es la gestión de reembolsos de tickets aéreos en las plataformas de contabilidad, la herramienta debe crear solicitudes de reembolso que deben ser autorizadas por las aerolíneas. El equipo de Operaciones se encargaba de detectar aquellas solicitudes que fueron rechazadas, corregirlas y generar cambios en el producto para que estos tipos de reembolsos no volvieran a ser rechazadas en el futuro.
3. **Casos de atención:** el área de operaciones gestiona las comunicaciones del día a día con los clientes, por ejemplo, generando usuarios de las plataformas cuando fueran solicitadas, entregando reportes o gestionando multas de la aerolínea en caso de que la herramienta hubiera actuado de manera incorrecta.

El objetivo de mi posición como Director de Operaciones era asegurar el éxito de los distintos productos que la empresa ofrecía. Crear el equipo para sustentar y expandir estos productos.

Era responsable de diseñar objetivos de producto, definir métricas de éxito, liderar al equipo a conseguir sus objetivos y métricas, manejar relaciones con clientes actuales para darles soluciones, liderar proyectos claves con clientes estratégicos, definir y pedir el desarrollo de nuevas funcionalidades al área de ingeniería, asegurar procesos de colaboración efectivos entre Operaciones de Negocio, Ingeniería y ventas, además del diseño y reclutamiento de los perfiles necesarios para expandir el negocio.

Mis métricas principales fueron:

- Ingresos generados mensualmente
- Número de transacciones por cliente
- Tasa de efectividad de transacciones
- Tasa de uso de los productos por cliente
- Tiempo para integrar clientes estratégicos

Para este reporte de experiencia profesional, en mi rol de Director de Operaciones, fui líder de la integración de un cliente estratégico (Despegar) en el producto de automatización de reembolsos de tickets aéreos (ARP).

# Capítulo II: Descripción de proyecto

## Descripción general

Mi proyecto consistió en asegurar la exitosa integración del producto de Automatización de Reembolsos (ARP o Automated Refund Platform) para el cliente estratégico llamado Despegar en enero del 2022. El objetivo era proporcionar a Despegar un servicio online que cotizará los montos de reembolso de las reservas de sus clientes y solicitará los reembolsos a la aerolínea. El objetivo final era que los pasajeros pudieran autogestionar sus reembolsos a través del portal web de Despegar y tramitar su reembolso sin necesidad de contactar con el área de atención al cliente.

El proyecto me fue entregado por el director general después de que Despegar firmara los acuerdos comerciales con Deal Engine. Este proyecto era la máxima prioridad de Deal Engine en ese momento, ya que Despegar era la Agencia de Viajes Online más grande de Latinoamérica, representaba una oportunidad de incrementar al 50% los ingresos mensuales de ARP, además de que sería una historia de éxito que se podría usar para levantar la siguiente ronda de inversión. Específicamente se calculó que podría representar un ingreso mensual de \$ 25,000 USD a \$ 40,000 USD, lo que también duplicaría los ingresos totales de Deal Engine en ese momento.

Había 4 retos principales para este proyecto:

1. ARP seguía siendo un producto en Beta<sup>23</sup>, lo que significaba que no todas las funcionalidades estaban listas ni la solución abarcaba todos los escenarios de reembolso.
2. Despegar maneja operaciones en 11 países de Latinoamérica y en ese momento solo teníamos clientes en 2 países para ARP, por lo que nuestra solución no abarcaba los escenarios presentes en los demás 9 países y se tendrían que desarrollar.

---

<sup>23</sup> *Beta test*. (2021, 2 septiembre). <https://www.productplan.com/glossary/beta-test/>

3. Despegar fue el primer cliente que solicitó que sus pasajeros pudieran usar las funcionalidades de ARP directamente en su página web, evitando agentes de viajes y consultores externos. Esto significaba mejorar nuestra tasa de éxito en el cálculo y solicitud de reembolso para ofrecer una adecuada experiencia al pasajero, además de acelerar el tiempo de cotización y solicitud para evitar largos tiempos de espera.
4. El cliente requería que la integración debía completarse en un 1 mes, lo que suponía acelerar la integración en comparación integraciones pasadas, que en promedio llevaban 3 meses.

Adicional a los retos anteriores, el director general definió que la métrica de éxito del proyecto era conseguir al menos 20,000 transacciones exitosas mensuales. Dada la naturaleza del producto en ese momento no se especificó un tiempo límite, sin embargo, no debía de tardarse tanto que arriesgara el compromiso del cliente en el proyecto. Tenía la capacidad de pedir todos los recursos que necesitara dentro de los alcances de Deal Engine, siempre y cuando no perjudicara la operación normal de otros productos activos.

Para abordar este proyecto decidí partirlo en las siguientes etapas:

1. Alinear alcance y etapas
2. Validar y conseguir accesos
3. Diseñar métricas de éxito y pruebas manuales
4. Pruebas automatizadas y despliegue en producción

Tabla 1. Etapas de integración ARP con Despegar

Tema	Objetivos	Participantes	Output
<b>1</b> <b>Alinear alcances y objetivos</b>	<ul style="list-style-type: none"> <li>• Iniciar relación con cliente, conocer a contactos claves</li> <li>• Alinear expectativas, objetivos y métricas de éxito</li> <li>• Alinear etapas del proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Equipo de operaciones</li> <li>• Equipo de reembolsos del cliente</li> <li>• Equipo de ingeniería del cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Plan de trabajo</li> <li>• Etapas y tiempos alineados</li> <li>• Lista de funcionalidades faltantes</li> </ul>
<b>2</b> <b>Validar y conseguir accesos</b>	<ul style="list-style-type: none"> <li>• Integrar los accesos de Despegar a ARP</li> </ul>	<ul style="list-style-type: none"> <li>• Equipo de operaciones</li> <li>• Equipo de ingeniería</li> <li>• Equipo de reembolsos del cliente</li> <li>• Equipo de ingeniería del cliente</li> <li>• Equipo de globalizadores</li> </ul>	<ul style="list-style-type: none"> <li>• Primer cotización y reembolso en todos los globalizadores</li> </ul>
<b>3</b> <b>Diseñar métricas de éxito</b>	<ul style="list-style-type: none"> <li>• Definir las métricas de éxito de las transacciones de cotización</li> </ul>	<ul style="list-style-type: none"> <li>• Equipo de operaciones</li> <li>• Equipo de ingeniería</li> <li>• Equipo de reembolsos del cliente</li> <li>• Equipo de ingeniería del cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Implementación de las métricas de éxito</li> <li>• Pruebas de cotización de Despegar</li> <li>• Implementación de requerimientos adicionales</li> </ul>
<b>4</b> <b>Lanzamiento</b>	<ul style="list-style-type: none"> <li>• Lanzamiento de producto dentro de página web del cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Equipo de operaciones</li> <li>• Equipo de ingeniería</li> <li>• Equipo de reembolsos del cliente</li> <li>• Equipo de ingeniería del cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Primer cotización exitosa de reembolso en página web del cliente</li> </ul>

Fuente: Elaboración propia

## Etapa I: Alinear alcance y etapas

El objetivo fue iniciar la relación operativa con el cliente (Despegar) y acordar las etapas y tiempos del proyecto. Las actividades principales que realicé fueron:

1. Reunirme con el equipo operativo y de tecnología de Despegar
2. Explicarles las capacidades actuales del producto
3. Determinar junto con el cliente las capacidades adicionales que debíamos crear
4. Alinear métricas de éxito
5. Alinear internamente en Deal Engine la metodología de trabajo

Dado los requerimientos del cliente, era crítico poder establecer una metodología de trabajo interno que nos permitiera levantar requerimientos, desarrollarlos, desplegarlos y probarlos rápidamente, por lo que mi primera tarea fue diseñarlo. Para ello tomé en cuenta las existentes metodologías existentes en la industria del software que describiré a continuación.



## *Metodologías de desarrollo de software*

Existen dos filosofías prominentes en el desarrollo de software que son la Metodología de Cascada<sup>24</sup> (Waterfall) y el Desarrollo Ágil<sup>25</sup>.

La Metodología de Cascada busca administrar proyectos mediante la exhaustiva planeación de etapas lineales que se deben completar de inicio a fin para el éxito del proyecto. Su principal ventaja es que una vez planeado, todos los colaboradores tienen visibilidad de todas las etapas, de sus obligaciones, entregables, fechas de entrega, además de que el seguimiento es muy sencillo ya que se puede saber en que etapa se está en el proyecto en cualquier momento. Solo funciona bien para proyectos donde los requerimientos y etapas sean fáciles de planear, se acuerden desde el principio y en general no sea sujeto a muchos cambios, en este caso es una metodología muy eficiente que evita retrabajos. En cambio, no funciona bien para proyectos dinámicos donde solo se conoce el objetivo, pero no se conoce de primera mano que se debe hacer para llegar, ya que hay un costo muy grande a la hora de cambiar la planeación.

Las etapas de esta metodología involucran:

1. Levantar requerimientos
2. Diseño de las etapas
3. Implementación de las tareas
4. Verificación y pruebas de calidad
5. Desplegar solución y continuo mantenimiento

El Desarrollo Ágil es una serie principios que en esencia buscan satisfacer al cliente a través de la entrega continua y rápida de pequeños avances de la solución. Rechaza el modelo de Cascada porque busca ofrecer una manera de trabajar que sea más flexible, colaborativo e iterativo que funciona mejor en proyectos donde el objetivo es claro, pero no tanto los pasos para cumplirlos. En general requiere mucha menor planeación que la Cascada y también es menos precisa a la hora de evaluar los avances, la etapa actual y el tiempo estimado para completar el proyecto. Dicho esto, es más exitoso a la hora de ofrecer una solución de valor

---

<sup>24</sup> Team, A. C. (s. f.). *Waterfall Methodology: Project Management | Adobe Workfront*.  
<https://business.adobe.com/blog/basics/waterfall>

<sup>25</sup> Agile Manifesto. (2001). *Principles behind the Agile Manifesto*. [agilemanifesto.org](https://agilemanifesto.org).  
<https://agilemanifesto.org/principles.html>

para el cliente en un entorno incierto. Se desarrollaron los principios justamente por la insatisfacción que sentían ingenieros prominentes de la industria por el uso de la Cascada, que en muchas ocasiones producía soluciones completas que, hasta el final del proyecto, el cliente se daba cuenta que no servía para cumplir con sus objetivos.

Los principios ágiles no ofrecen una metodología específica, sino que muchas metodologías fueron desarrolladas por empresas e individuos con mayor o menor éxito dependiendo en la naturaleza de su proyecto. A continuación, explicó algunas de las metodologías ágiles más usadas<sup>26</sup>:

- **Kanban**<sup>27</sup>: basado en la mejora continua, esta metodología ayuda a los equipos a tener visualizado las tareas a implementar a través de un tablero de tarjetas que contienen la descripción de la tarea. Estas tarjetas se suelen agrupar en grupos como Completado, En proceso o Por priorizar (conocido como Backlog). Las tarjetas se suelen escribir en el grupo de Por priorizar, después se mueven a En proceso cuando un miembro del trabajo está implementándolo, y finalmente pasa a Completado cuando la implementación está completo, validado, probado y desplegado. También ayuda a dar visibilidad de la disponibilidad de cada miembro del equipo, ya que las tarjetas en proceso se asignan a miembros individuales.
- **Programación Extrema**<sup>28</sup>: consiste en organizar las implementaciones y desplegarlo en unidades semanales llamadas Sprints y busca una alta colaboración de los ingenieros con el cliente, una cultura de cambio e iteración donde se pudiera cambiar y refactorizar el código de manera rápida y sin resistencia al cambio. Esta metodología requiere ingenieros altamente cualificados, con amplia experiencia que les permitiera hacer cambios rápidos y de calidad, y con excelentes habilidades de comunicación y análisis ya que debían entender y colaborar de cerca con el cliente.

---

<sup>26</sup> *Top 7 Agile frameworks*. (s. f.). <https://premieragile.com/types-of-agile-frameworks/>

<sup>27</sup> Martins, J. (2024, 19 enero). ¿Qué es la metodología Kanban y cómo funciona? [2024] • Asana. *Asana*. <https://asana.com/es/resources/what-is-kanban>

<sup>28</sup> Raeburn, A. (2024b, febrero 13). ¿Qué es la programación extrema (XP)? [2024] • Asana. *Asana*. <https://asana.com/es/resources/extreme-programming-xp>

- **Scrum**<sup>29</sup>: es parecido a la Programación Extrema pero abarca soluciones mas amplias que solo la programación, es mas estructurado y requiere equipos mas grandes para operarlo efectivamente. También divide el despliegue de soluciones en Sprints<sup>30</sup>, generalmente documentados usando Kanban. Requiere de varios roles incluyendo un Dueño del Producto, un Maestro Scrum y Desarrolladores, cada uno con responsabilidades y entregables específicos. También busca lograr alta colaboración del cliente en el proceso para poder iterar soluciones.
- **DevOps**<sup>31</sup>: es una metodología que busca la colaboración de todos los participantes involucrados en el desarrollo de software, calidad, administradores de servidores y IT para generar entregas continuas, de alta calidad y poco propensas a fallos.

### *Diseño de Metodología*

Dado la importancia de la velocidad y la naturaleza incierta del proyecto, decidí implementar una metodología de trabajo basándome en los Principio Ágiles y no usar el Método de Cascada. Esto nos permitiría trabajar de cerca con el cliente para asegurar su satisfacción a través de un equipo pequeño que pudiera entregar de manera frecuente y con alta calidad. La metodología que implementé tenía 4 procesos claves:

- 1) **Reuniones semanales con cliente**: implementé una reunión los miércoles que usábamos para enseñar nuevos desarrollos, resolver dudas, actualizar el mapa del producto, recolectar necesidades y desbloquear cualquier problema. En ella participaban del lado de Deal Engine el Director de Tecnología, mi equipo y yo, del lado de Despegar estaba la Manager de Reembolso, el Dueño Técnico del Proyecto de reembolsos y el Manager de Innovación.

---

<sup>29</sup> *What is Scrum?* (s. f.). Scrum.org. <https://www.scrum.org/resources/what-scrum-module#:~:text=Scrum%20helps%20people%20and%20teams,optimize%20for%20their%20specific%20need> s.

<sup>30</sup> Rehkopf, B. M. (s. f.). *Scrum Sprints: Everything You Need to Know* / Atlassian. Atlassian. <https://www.atlassian.com/agile/scrum/sprints>

<sup>31</sup> *¿Qué es DevOps? Explicación de DevOps* / Microsoft Azure. (s. f.). <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>

- 2) **Sprints semanales:** los sprints son espacios de una semana donde a los ingenieros se les encomendaban tareas que debían de publicarse a producción al final de la semana. Estas tareas se investigaban y documentaban una semana antes, ya sea por mí o por analistas en mi equipo, para que pudieran ser desarrolladas por el equipo de ingeniería durante la siguiente semana.
- 3) **Sesión de priorización semanal:** la documentación que yo y mi equipo desarrollábamos eran priorizados y guardados en orden de prioridad dentro de un backlog específico de Despegar. Al final de la semana se tenía una reunión con el director de tecnología para validar las prioridades del backlog y mover las tareas al kanban de la siguiente semana.
- 4) **Pruebas y arreglos semanales:** después de desplegar los cambios desarrollados por el equipo de ingeniería, mi equipo tenía 2 días para probar la funcionalidad y validarlo. De encontrarse algún problema que debía resolverse, se pedía al equipo de ingeniería que se repriorizará este arreglo para poder mostrarlo el miércoles durante la reunión semanal. Adicionalmente, si durante el miércoles el cliente nos hacía llegar algún comentario o queja, se hablaría con el equipo de ingeniería para repriorizar la mejora relacionada con el comentario o queja.

Con estos sencillos 4 puntos, sentamos la base para poder entregar valor rápidamente al cliente y asegurar el éxito del proyecto en apenas 3 meses.

Entre los retos que encontré fue la resistencia del equipo de ingeniería a aplicar nuevas maneras de trabajar, ya que se tenía un proceso y se tenía la preocupación de reducir esfuerzos en otros frentes. Para resolverlos, alineé con el Director de Tecnología cuál era la máxima prioridad en ese momento para Deal Engine. Ambos acordamos que era el proyecto de Despegar, por lo que se comprometió a que pudiera tener disponible 2 ingenieros y a implementar la metodología para el proyecto. Adicional, le invité a participar en las reuniones semanales con Despegar para que desarrollara un sentido de pertenencia al proyecto y estuviera involucrado en su éxito.

Estos 4 procesos tienen elementos de las metodologías ágiles anteriormente descritas:

- **Kanban:**
  - Usamos un tablero Kanban para tener completa visibilidad y transparencia de las tareas que se debían completar, quien las estaba implementando y cuales habían sido completadas.
  - Este tablero nos permitía organizar las tareas de los ingenieros y medir su rendimiento.
  - Cada vez que el cliente o nosotros necesitáramos añadir un nuevo requerimiento, era muy sencillo colocarlo en el backlog y compararlo a la prioridad de las demás tarjetas, lo que nos daba mucha flexibilidad.
  - El tablero representaba la alineación del equipo completo sobre las prioridades, lo que también permitía una efectiva comunicación de estos hacia la empresa y los directores.
- **Programación Extrema:**
  - Teníamos un equipo pequeño de 2 ingenieros, pero cada uno con mucha experiencia y agilidad en las herramientas de programación que usábamos. Siempre estuvieron involucrados en las reuniones semanales con cliente lo que les permitía colaborar para levantar exactamente el requerimiento.
  - Dada su experiencia e involucración en el proyecto, los ingenieros eran capaces de entregar implementaciones rápidas e iterativamente, prefiriendo entregar pequeños pero continuos avances en vez de tardarse más para desplegar una funcionalidad completa. Aunque pudiera parecer menos eficiente, el cliente agradecía poder ver avances constantes y también tenía la oportunidad de darnos retroalimentación temprana.
  - Cada implementación debía ser rigurosamente probado usando pruebas unitarias y de integración en el código. Esto permitía comprobar de manera automática que sus cambios no habían roto alguna funcionalidad anterior y que funcionara exactamente como lo requería el cliente. Aunque esto suponía añadir tiempo de prueba en el proceso, rápidamente se hizo evidente que era mucho mayor el beneficio que obteníamos de minimizar retrabajos y maximizar la estabilidad del cliente. Adicional esto nos permitía conservar

capital social con el cliente, ya que reducíamos el número de errores que nos hubieran tenido que comentar.

- La discusión, priorización y asignación de tareas ocurría en sprints semanales.
- Conservé la cultura del cambio y refactorización en la metodología. En situaciones normales los ingenieros presentan resistencia al cambio a la hora de cambiar o remover código previamente escrito, pero en la programación extrema el cliente y sus necesidades son lo más importante, por lo que se genera una apertura grande para modificar o remover el código.
- No implementé un proceso de documentación extensiva de las tarjetas y fue porque quería evitar que los ingenieros y analistas pasaran demasiado tiempo documentando y leyendo requerimientos. Esto es especialmente importante ya que muchas veces debíamos quitar o modificar una funcionalidad previamente acordada con el cliente, por lo que no hacía sentido pasar demasiado tiempo en documentación. Esto era posible ya que éramos un equipo pequeño que tenía mucha cercanía con el cliente, por lo que era fácil que todos tuvieran contexto de las tareas y no necesitaran mucha documentación para entender que debían implementar.
- **Scrum:**
  - En las siguientes etapas decidí que para que esta metodología fuera exitosa necesitaba crear un rol que fuera responsable de ejecutarlo correctamente con todas las áreas. Esta idea fue inspirada por la figura del Scrum Master, que en este caso lo asigné Jefe de Producto, quien también actuaba como Product Owner (Dueño de Producto).
  - Al igual que Scrum nuestra planeación se hacía en Sprints y con todos los miembros internos presentes.
  - Se construyó un Roadmap general del producto que actuaba de guía a la hora de alinear prioridades y siguientes pasos. Este mapa cuenta historias de éxito de manera general y evita ser muy específico porque no aspira a ser una secuencia de tareas como usaría el Método Cascada, sino que sirve para generar discusión y darnos una idea aproximada del estatus de un producto.

- **DevOps:** Para soportar una cultura de iteración y constantes despliegues, fue necesario seguir una serie de buenas prácticas:
  - Todo nuestro código al igual que los cambios requeridos se manejaron usando un controlador de versiones<sup>32</sup>. Esto permitía administrar las distintas versiones y cambios en el código al igual que tener control de quien realizaba las modificaciones. De esta manera podíamos revertir cambios si detectábamos algún problema. El controlador de versiones usado fue git<sup>33</sup>.
  - Usamos gitlab<sup>34</sup> como repositorio online de las versiones, esto nos permitía sincronizar las distintas versiones de código entre los ingenieros.
  - Cualquier cambio al código debía ser revisado por otro ingeniero que no hubiera participado en su creación, este proceso de revisión se llevaba a través de gitlab y el cambio no se podía aprobar y desplegar hasta que todos los comentarios del revisor fueran solucionados o alineados.
  - A través de gitlab construimos un proceso CI/CD<sup>35</sup> (Continuous Integration/Continuos Delivery), que es una serie de pasos que buscan permitir cambios continuos en código centralizado en un mismo repositorio online (gitlab), a su vez que permitir el despliegue sencillo de este código, el cual logramos al conectar gitlab con los servidores, lo que nos permitió desplegar cambios con un único botón.
  - Implementamos un proceso de revisión donde todas las nuevas versiones debía ejecutar una serie de pruebas automatizadas (en su momento había cerca de 1,000 pruebas distintas) para asegurar que el cambio no rompiera alguna funcionalidad previamente implementada, adicionalmente el propio cambio debía incluir una nueva prueba. El revisor no podía aprobar un cambio sin que antes las pruebas hubieran sido todas exitosas.

---

<sup>32</sup> Atlassian. (s. f.). *What is version control | Atlassian Git Tutorial*.

<https://www.atlassian.com/git/tutorials/what-is-version-control>

<sup>33</sup> *Git*. (s. f.). <https://git-scm.com/>

<sup>34</sup> *The most-comprehensive AI-powered DevSecOps platform*. (s. f.). GitLab. <https://about.gitlab.com/>

<sup>35</sup> *What is CI/CD?* (s. f.). <https://www.redhat.com/en/topics/devops/what-is-ci-cd>

- Para incrementar la calidad y disponibilidad de nuestros servidores, fue crucial implementar un sistema de observabilidad<sup>36</sup> que nos permitiera saber que estaba ocurriendo en el servidor, que errores estaban ocurriendo y por qué. Usando grafana<sup>37</sup> y prometheus<sup>38</sup> pudimos tener un sistema de alertas y métricas que nos permitían reaccionar y reducir inestabilidades en el sistema. Ambos son herramientas altamente usados en la industria, con excelente documentación, soporte y son de código abierto por lo que no representaban ningún costo de suscripción.
- La mayoría de las métricas importantes del producto las hicimos disponibles en herramientas de inteligencia de negocio para permitir el trabajo de los analistas. Usamos la herramienta de Metabase<sup>39</sup> ya que tiene una excelente documentación, soporte, interfaz gráfica y una versión de código abierto sin ningún costo de suscripción.
- Los servidores requerían de configuraciones específicas y credenciales para que pudieran funcionar las cuales versionamos usando git dentro de un repositorio encriptado. Esto nos permitía cambiar las configuraciones de manera rápida y poder desplegarlo sin hacer cambios en el código. Además, nos permitió controlar quien en el equipo tenía permisos para cambiar configuraciones específicas, incrementando la seguridad de la infraestructura.
- Implementamos una agenda de despliegue, la cual incluía desplegar una vez por día entre las 9 y 12 de la mañana. Esto nos permitía desplegar cada día en un horario donde tuviéramos algún ingeniero en guardia para asegurar que los servidores continuaran funcionando, pudiendo reaccionar y arreglarlo en caso de que algo hubiera salido mal. Adicionalmente, no se podía desplegar de

---

<sup>36</sup> Mireles, Y. (2024, 24 enero). What is observability? *New Relic*. <https://newrelic.com/blog/best-practices/what-is-observability>

<sup>37</sup> Grafana Labs. (s. f.). *Grafana: The open observability platform | Grafana Labs*. <https://grafana.com/>

<sup>38</sup> Prometheus. (s. f.). *Prometheus - Monitoring system & time series database*. <https://prometheus.io/>

<sup>39</sup> Metabase | Business Intelligence, Dashboards, and Data Visualization. (s. f.). Metabase | Business Intelligence, Dashboards, And Data Visualization. <https://www.metabase.com/>



viernes a domingo para evitar que algún problema ocurriera en el servidor y no hubiera un ingeniero disponible para arreglarlo o monitorearlo.

Vale la pena mencionar que la metodología que implementé solo se inspira en otras metodologías ágiles, no las usa completamente. Esto se debe a que en general cada metodología tiene una serie de reglas, entregables y roles muy específicos, su correcta implementación requeriría de un enorme esfuerzo e inversión inicial en conocer y ejecutar los procesos al pie de la letra. Esto no nos permite la flexibilidad y rapidez que el proyecto requería.

En el capítulo de análisis de resultados comentaré sobre los resultados de esta metodología.

## Etapa II: Validar y conseguir accesos

El objetivo de esta etapa es solicitar y validar los accesos/permisos que el cliente nos debía compartir para que la solución funcionara.

Para cotizar un reembolso es necesario poder acceder a la información de las reservas, tickets, reglas tarifarias y de impuestos, información que está disponible en los accesos API<sup>40</sup> de los GDS<sup>41</sup> (Global Distribution System) y sistemas de contabilidad.

Como contexto, una API es un sistema de comunicación que permite a 2 robots transmitir información o solicitar acciones. Por ejemplo, en la página web de Despegar es posible reservar un viaje, esto ocurre, ya que el navegador web de la computadora o celular del pasajero manda una solicitud de reserva a la API de los GDS, quienes generan la reserva en su sistema y responden al navegador con un mensaje de confirmación. Las APIs son tecnologías críticas que permiten automatizar procesos en todo tipo de industrias y aplicaciones, en el caso de la cotización de reembolsos, el servicio de ARP usa las APIs de

---

<sup>40</sup> *¿Qué es una API? - Guía sobre las API para principiantes - AWS.* (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/api/>

<sup>41</sup> Fernández-Paniagua, A. M. del C. (2017). *¿Qué son los Softwares GDS y para qué sirven?* Thinking for Innovation. <https://www.iebschool.com/blog/software-gds-marketing-digital/>

los GDS para consultar información de la reserva (como la tarifa e impuestos pagados), el texto de las reglas tarifarias y el texto de las reglas de impuestos.

Adicionalmente, es conveniente dar contexto sobre cómo se ofrece la oferta de la industria aérea. En Deal Engine definimos 2 canales que son los GDS y NDC (New Distribution Capability):

- 1) Los GDS son empresas que estandarizan la oferta de múltiples aerolíneas y ofrecen una interfaz gráfica/API para que agencias de viajes reserven y administren viajes. Estos servicios se volvieron el de facto estándar de la industria de viajes, ya que permiten a las agencias solo tener que conectarse a alguno de estos GDS en vez de a todas las aerolíneas. Hay 3 grandes jugadores que son Amadeus, Sabre y Travelport. Nuestro cliente trabaja con los 3 GDS en una proporción similar, por lo que era importante incluirlos todos en el proyecto.
- 2) NDC<sup>42</sup> es un estándar que las aerolíneas pueden seguir para desarrollar su API de oferta, el objetivo es quitarse a los GDS como intermediarios para poder ofrecer precios más bajos. La propuesta para las agencias de viaje sería que no tendrían que desarrollar varias conexiones por aerolínea porque todas estas tendrían la misma estructura de API, por lo que desarrollar una conexión, en teoría, funcionaría para todas las aerolíneas. Este estándar sigue siendo nuevo en la industria, por lo que Despegar no lo usaba en las mismas proporciones que los GDS, y, por tanto, quedo fuera del alcance del proyecto.

Este paso siempre es un reto a la hora de integrar clientes, debido a que estos están acostumbrados a trabajar con la interfaz gráfica de los GDS en vez de los accesos API, los cuales permiten hacer peticiones robóticas desde nuestro servicio. Para resolver este reto con Despegar, solicité incluir a los representantes de Amadeus, Sabre y Travelport en nuestras reuniones semanales para que pudieran ayudar a acelerar la creación de los accesos y resolver dudas.

---

<sup>42</sup> *Distribution with Offers & Orders (New Distribution Capability)*. (n.d.). [Www.iata.org. https://www.iata.org/en/programs/airline-distribution/retailing/ndc/](https://www.iata.org/en/programs/airline-distribution/retailing/ndc/)

Adicionalmente, en ese momento teníamos mucha experiencia trabajando con Amadeus y Sabre; sin embargo, no lo teníamos con Travelport. Para hacer frente a este reto, solicité que el cliente nos pusiera en contacto con su agente de Travelport, con el que iniciamos reuniones recurrentes donde nos compartieron manuales, recomendaciones y nos ayudaron a resolver dudas. Esto me permitió documentar la información necesaria para incluirlo en los sprints de desarrollo y llevaría 1 mes completar la integración.

Otro reto que tuvimos fue el validar que los privilegios de los accesos fueran completos. Por seguridad, los accesos tienen limitadas funcionalidades y es nuestra responsabilidad solicitar los privilegios que necesitamos para nuestro servicio. El problema es que activar y desactivar privilegios toma mucho tiempo del lado de los GDS, y es propenso a errores de comunicación. Para solucionarlo decidí crear una nueva posición llamada TAM (Technical Account Manager), quien se encargaría de solicitar y administrar los accesos de nuestros clientes.

Desarrollé el perfil y puse la vacante en redes sociales, entrevisté alrededor de 10 candidatos usando diversas técnicas de entrevistas como STARS<sup>43</sup> (entrevistas basadas en comportamientos), diseñé y apliqué evaluaciones, tuve retroalimentación con pares y al final contraté a uno de ellos.

Una evaluación técnica en el proceso de reclutamiento y selección está diseñada para medir las habilidades y competencias específicas de un candidato en relación con el puesto de trabajo que se está considerando. Este tipo de evaluación se utiliza principalmente en roles que requieren habilidades técnicas o especializadas, como programadores, ingenieros o diseñadores, y puede incluir pruebas prácticas, ejercicios de resolución de problemas o cuestionarios técnicos. El objetivo es identificar de manera objetiva si el candidato posee las capacidades necesarias para desempeñar eficazmente el trabajo y tomar decisiones fundamentadas en la selección de personal.<sup>44</sup>

---

<sup>43</sup> *How to Use the STAR Method for Behavioral Interviews*. (n.d.). Recruitment.

<https://recruitment.com/process/star-interview-questions>

<sup>44</sup> *IA abierta*. (2023). ChatGPT (versión del 14 de marzo) [Modelo de lenguaje grande].

<https://chat.openai.com/chat>

Posteriormente, le ayudé a documentar nuestro conocimiento en los GDS, a desarrollar material para compartir con clientes, desarrollar una serie de herramientas que validaran automáticamente los accesos y privilegios. De esta manera conseguimos que el proceso de conseguir los accesos en Despegar fuera más corto (1 mes) en comparación a otros clientes pasados (en promedio 3 meses), y aseguramos que sería más rápido aún en el futuro.

### Etapa III: Diseñar métricas de éxito y pruebas manuales

El objetivo de esta etapa fue alinear las métricas de éxito del proyecto junto con el cliente, para posteriormente desarrollarlas y usarlas como base para hacer las primeras pruebas manuales. Para ello se utilizó las mismas sesiones mencionadas en las etapas anteriores, donde aprendimos que el cliente quería conectar nuestra solución a su página web para que sus pasajeros pudieran cotizar el reembolso de su reserva sin tener que contactar a un agente telefónico de Despegar. Esto significaba que debíamos preparar 3 puntos que no teníamos en cuenta para proyectos pasados:

- 1) **Precisión:** las cotizaciones debían devolver el monto exacto de reembolso desde la creación. Esto es un reto porque el algoritmo debe interpretar lenguaje humano o tener en cuenta reglas de negocio, el cual podría no ser tan preciso o no estar cargado en el sistema. En el pasado esto no era problemático porque el servicio solo era usado por los agentes de viaje, quienes tenían la posibilidad de hacer pequeñas correcciones en la interfaz, sin afectar el tiempo total de reembolso significativamente. Sin embargo, ahora que los pasajeros serían los primeros que verían la cotización, era importante que vieran el monto correcto.
- 2) **Rapidez:** los usuarios de páginas web están acostumbrados a tiempos rápidos de respuesta, por lo que era importante que nuestro servicio tardara pocos segundos en procesar sus solicitudes para que su experiencia fuera positiva. Esto fue el reto más grande ya que durante la cotización se usaban servicios webs de proveedores que podían ser lentos, por lo que del lado de Deal Engine tuvimos que idear estrategias para reducir el impacto de la lentitud de los proveedores.

- 3) **Estabilidad:** Despegar tiene miles de usuarios online en cualquier instante, por lo que era crítico que tuvieran acceso a la plataforma y que no vieran caídas en el servicio. Hasta entonces nuestros procesos de despliegue o de manejo de excepciones eran muy lentas debido a que no había una presión por tener una plataforma que no fallara, pero en este escenario tendríamos que adaptar nuestros procesos para reducir al mínimo el número de incidencias que nuestro servicio presentara.

Para hacer frente a estos retos, diseñé y confirmé internamente y con el cliente las siguientes métricas:

- 1) **Safe rate (tasa segura):** es una variable categórica que indica si el algoritmo de cotización detectó un error en alguna de las fases del cálculo. Decidí no llamarlo tasa de precisión ya que no hay manera de saber automáticamente si una cotización está bien calculada, pero sí podemos detectar ciertos escenarios donde sabemos que muy probablemente están mal calculados.

Aquellas cotizaciones donde no detectáramos ningún problema se marcan como cotizaciones seguras, lo que significa que el cliente puede arriesgarse a compartirlo automáticamente con sus pasajeros web. Es importante mencionar que, aunque el algoritmo no haya detectado un error, no significa que la cotización sea correcta (haría falta la evaluación de un agente humano para entender si está bien), simplemente consideré que había pocas probabilidades de que estuviera mal, en la práctica solo un 5% de las cotizaciones seguras se descubrieron como no seguras en una posterior auditoría.

Las categorías de esta métrica son:

- a) **OK:** no detectamos ningún error, es seguro que el pasajero vea la cotización.
- b) **OK, no open coupons (no hay cupones abiertos):** detectamos que todos los vuelos fueron usados (esto significa que fueron volados, reembolsados o intercambiados en el pasado) por lo que no se puede reembolsar, es decir, la cotización indica un monto de 0. Esta cotización es segura para compartir con el pasajero.

- c) **Tax calculator error (error de calculadora de impuestos):** el algoritmo no consiguió cotizar el reembolso de uno o varios impuestos, esta cotización no es segura para compartir con el pasajero.
  - d) **Fare rule error (error de regla tarifaria):** el algoritmo no consiguió interpretar las reglas tarifarias, esta cotización no es segura para compartir con el pasajero.
  - e) **Historic pricing error (error de cotización histórica):** el algoritmo no consiguió encontrar una tarifa equivalente en la cotización del reembolso de un ticket parcialmente volado. La cotización histórica es un método de cotización de reembolsos que ciertas aerolíneas exigen cuando un ticket fue parcialmente volado, esto tiene como objetivo evitar fraudes y el algoritmo debe ser capaz de calcularlo. En caso de no poder calcularlo, se desplegará este error en la métrica de Safe rate y se considerará que la cotización no es segura para compartir con el pasajero.
- 2) **Timeout rate (tasa de cotizaciones irresponsivas):** es una variable booleana que indica si una cotización tardó en crearse en menos de 30 segundos. Diseñé esta variable para medir cuantas cotizaciones los pasajeros no eran capaces de ver en la página web, esto se debe a que los navegadores web tienen un tiempo de respuesta de 30 segundos ante cualquier petición, en caso de tardarse más, se cancela la petición y al pasajero no le llegará los resultados de la cotización.
- 3) **Uptime (tiempo de actividad):** esta es una métrica estándar de la industria de software que mide el porcentaje de tiempo que un programa está funcionando y es capaz de responder ante las peticiones de sus usuarios. Decidí implementar esta medición en ingeniería para que se aseguraran de mantener una mínima tasa del 99%, que acordé con el director de tecnología. Esta tasa mínima exige que la herramienta solo pudiera estar fuera de actividad por 1 hora y 40 minutos por semana. En la práctica después de 6 meses conseguimos una tasa del 99.975%, la cual significa que la herramienta solo estuvo fuera de actividad por 2 minutos y 30 segundos semanalmente.

La implementación de todas estas métricas fue documentada por mi equipo y compartido con ingeniería, quienes las desarrollarían y expondrían en la interfaz y API de la herramienta, para que tanto mi equipo como el cliente tuviera acceso a medirlos. Todo este proceso llevó aproximadamente 3 semanas desplegarlo.

Una API, o Interfaz de Programación de Aplicaciones, es un conjunto de reglas y protocolos que permite que diferentes aplicaciones o sistemas informáticos se comuniquen entre sí y compartan datos o funcionalidades de manera estructurada. Actúa como un intermediario que permite que un programa solicite y obtenga información o servicios de otro sin necesidad de conocer todos los detalles internos de su funcionamiento. Las API son esenciales en el desarrollo de software, ya que permiten la integración de servicios y la creación de aplicaciones que pueden interactuar con diversas plataformas y servicios externos de manera eficiente y coherente. En este caso se usó el lenguaje de programación Scala, porque es un lenguaje de programación especializado en múltiples transacciones concurrentemente y es fácil su desarrollo, por lo que es rápido entregar valor al cliente.

Teniendo estas métricas y los accesos de la etapa 2, generé usuarios en el servicio para el cliente pudiera hacer pruebas manuales.

Esto era importante porque los ejecutivos de reembolsos del cliente debían dar el visto bueno para que sus ingenieros empezaran a desarrollar la conexión con la herramienta. Debido a que el proceso de reembolso es complejo y para asegurar que supieran usar la interfaz, aproveché las sesiones semanales para dar una capacitación en el uso de la herramienta. Con esto asegurado, les pedí que durante su jornada laboral probaran a cotizar reembolsos que los pasajeros les fueran haciendo llegar durante el día con el objetivo de recolectar datos y asegurarnos de que la herramienta tuviera una calidad inicial adecuada. Como condición de éxito para pasar a la siguiente etapa, alinee con el cliente que la herramienta debía tener unas métricas del 90% para el Safe rate y 30% para el timeout.

Esto fue crucial porque las cotizaciones creadas por el cliente eran monitoreadas durante el día por mi equipo, quienes analizarían la calidad y documentarían cambios para el equipo de

ingeniería. Esto permitía mejorar la herramienta incluso antes de que el cliente comunicara cualquier error, o incluso nos permitía resolver errores que el cliente no detectaba.

Dicho esto, decidí darle prioridad a aquellos problemas que el cliente nos comunicaba directamente para darles una buena sensación de avance. Después de 4 semanas de pruebas y 300 casos de cotización, documentación y mejoras del lado de ingeniería, se llegaron a las métricas alineadas, por lo que dio comienzo a la siguiente etapa.

## Etapa IV: Pruebas automatizadas y despliegue en producción

El objetivo de esta etapa fue asegurar el éxito de la conexión entre la API de la herramienta con la página web del cliente. Al cumplir con las métricas acordadas del caso anterior, el cliente dio el visto bueno a sus ingenieros para empezar la conexión.

Para que esto fuera posible, documenté y alineé con el equipo de ingeniería de Deal Engine la creación de una API que pudiera tener la misma funcionalidad que la interfaz, permitiendo al robot del cliente interactuar programáticamente con las funciones de la herramienta.

La buena noticia es que el equipo de ingeniería ya había desarrollado la primera versión de la API, la cual funcionaba como motor para la interfaz gráfica. Por ello, apenas se tardó 2 semanas para desplegar públicamente la API y que el cliente tuviera acceso a ello. Los cambios necesarios fueron la simplificación de ciertos pasos, el renombrado de las funciones para que fueran más intuitivas para clientes externos y desarrollar un módulo de autenticación para usuarios externos.

Una vez publicado públicamente la API, ejecuté una capacitación con el equipo de ingeniería de Despegar, quien tardó 6 semanas en completar la conexión. Para asegurar que este proceso fuera el más rápido posible, se siguió usando las sesiones semanales para que poderles resolver dudas o incluir mejoras en la API, tiempo durante el cual desplegamos ciertas mejoras y arreglos que fuimos identificando con el cliente.

Una vez completado la integración, el cliente desplegó la conexión en su página web, la cual ahora era capaz de mostrar una cotización de reembolso en tiempo real de la reserva de un



pasajero, en vez de enseñarle una pestaña redirigiendo hacia atención al cliente. El resultado nominal fue un éxito, consiguiendo cotizar en aproximadamente 60 mil reembolsos en el primer mes. Las métricas iniciales fueron menores a las deseadas, con un 85% en Safe rate y 40% de timeout.

Para mejorar las métricas, decidí tomar un acercamiento sistémico con el objetivo de implementar un proceso de mejora continua. Esto era muy importante, ya que el éxito del proyecto logró llamar la atención de nuevos clientes, por lo que pronto tendría que alejarme de este proyecto para liderar nuevas integraciones. Para ello, creé el Área de Producto, la cual se encargaría de monitorear y mejorar las métricas del producto, a través de la documentación y colaboración constante con el equipo de ingeniería y el cliente. Diseñé la descripción del rol y la compartí con el director general, el cual me ayudo en conectar con la subgerente de TI de uno de nuestros primeros clientes.

Debida a sus 20 años de la experiencia en el sector de viajes y de los GDS, decidí contratarla en el rol de Head of Product (Gerente de Producto), con la misión de crear un equipo de producto capaz de ejecutar eficazmente su misión de mejorar continuamente las métricas del producto.

1 año después, esta área tiene 5 colaboradores y ha conseguido unas métricas del 98% de Safe rate y 3% de timeout.

Para el cliente se tradujo en una atención del 70% de las solicitudes de reembolso por medios automatizados, que ahora son gestionados por el propio pasajero con una mucho mejor experiencia al cliente. Además, se tradujo en una reducción de los costos significativa para la agencia de viaje pues el desarrollo tecnológico hace el trabajo equivalente a 100 personas del área de servicio.

# Capítulo III: Análisis de resultados

En resumen, creo que el proyecto de integración fue un éxito parcial por múltiples razones. Para analizarlos, categorizaré los resultados obtenidos en resultados financieros, resultados de calidad y resultados de la metodología.

## Resultados financieros

En los primeros meses después de desplegar públicamente ARP con Despegar, se facturó unos \$ 50,000 USD mensuales en promedio, lo que representaba duplicar el ingreso en Deal Engine. Dicho esto, el equipo de Despegar ajustó su robot para ser más eficiente a la hora de consultar nuestro servicio, lo que resultó en estabilizar el ingreso promedio mensual entorno a \$ 30,000 USD. Con este resultado final se puede concluir que fue un éxito ya que era el cliente con mayor facturación tanto del producto como de la empresa.

No solamente eso, demostraba que la herramienta era capaz de manejar grandes volúmenes de reembolsos y que era escalable a cualquier otro cliente. Esto nos permitió seguir vendiendo ARP teniendo un caso de éxito sólido en nuestro arsenal. Adicionalmente, esto significaba que cualquier otra agencia de viajes que no estuviera usando ARP era menos competitivo que Despegar en cuestión de costos y experiencia del usuario durante la gestión posventa.

También en términos de transacciones fue un éxito. Antes de Despegar se realizaban aproximadamente 1,500 transacciones de cotización y solicitud de reembolso al mes, y después de la integración se pasaron a realizar entre 30 y 50 mil transacciones mensuales, entre 15 y 30 veces más transacciones, y también superó el objetivo del director general de 20,000 transacciones adicionales. No solamente es bueno en cuestión de ingresos, adicionalmente nos proporcionó muchos más casos que pudimos usar para evaluar, mejorar y profesionalizar la herramienta.

En término de costo de la integración estimó que en los primeros 3 meses se gastó \$ 15,000 USD cada mes, en su mayoría formado por los sueldos de los ingenieros, analistas, directores involucrados en el proyecto. Adicionalmente hay que factorizar \$ 500 USD mensuales de costo de servidor adicionales a los que ya pagábamos. Después de la publicación del servicio el tiempo dedicado por los ingenieros y directores al proyecto se redujo drásticamente a un estimado de \$ 3,000 USD mensuales. Teniendo en cuenta los ingresos y costos, estimo que la rentabilidad del primer año desde que empezó el proyecto

fue de \$ 258,000 USD. Esto no toma en cuenta el ingreso ni la disminución de costo del futuro.

Mes	1	2	3	4	5	6	7	8	9	10	11	12
Ingreso				\$ 50,000	\$ 50,000	\$ 50,000	\$ 30,000	\$ 30,000	\$ 30,000	\$ 30,000	\$ 30,000	\$ 30,000
Costo	\$ 15,000	\$ 15,000	\$ 15,000	\$ 3,000	\$ 3,000	\$ 3,000	\$ 3,000	\$ 3,000	\$ 3,000	\$ 3,000	\$ 3,000	\$ 3,000
Rentabilidad	\$ (15,000)	\$ (15,000)	\$ (15,000)	\$ 47,000	\$ 47,000	\$ 47,000	\$ 27,000	\$ 27,000	\$ 27,000	\$ 27,000	\$ 27,000	\$ 27,000
Rentabilidad anual	\$ 258,000											

Imagen 4. Impacto neto de la integración con Despegar

Fuente: Elaboración propia

Un resultado que no fue exitoso fue el hecho de que el cliente solo acabó usando ARP para la cotización, pero nunca para la solicitud del reembolso. Aunque ARP tiene la funcionalidad para automatizarles este proceso, el valor que el cliente percibía era mucho menor, pues la tarea de solicitar un reembolso es tediosa, pero no complejo (como si lo es el cálculo). Por esta razón ya habían automatizado en muchos casos la solicitud de reembolso, lo que significaba que esa parte del proceso era innecesario solicitarlo a ARP.

Además, aprendimos que parte del proceso de solicitud que Despegar ejecutaba era la cancelación completa de la reserva en aquellos casos donde el reembolso ocurría antes de los vuelos. Para incrementar la propuesta de valor de esta etapa era necesario realizar una serie de nuevas funcionalidades y mejoras a la experiencia al usuario, pero debido a otras prioridades mas urgentes nunca logré afinar esta etapa.

Un resultado clave de la integración fue lograr un caso de éxito que demostró que no solamente había mercado para ARP, sino que podía ser un éxito incluso en las agencias más grandes del mundo. Sabiendo eso, el siguiente paso lógico era expandir globalmente lo más rápido posible para lograr atrapar el mercado. Para ello requeríamos crear un equipo más grande compuesto por ingenieros, analistas, gerentes de producto y vendedores, pero eso requiere una inversión muy fuerte de capital. La única solución fue levantar una ronda de inversión, la cual se logró un año después del proyecto de integración, gran parte gracias a los resultados financieros logrados con este proyecto.

## Resultados de calidad

En cuestión de la calidad del cálculo de los reembolsos no conseguí buenos resultados al principio. Después de analizar los resultados decidí implementar un proceso de mejora continua que sería responsabilidad del Área de Producto. Su objetivo sería identificar las razones principales que causaban la baja tasa de éxito, investigar, proponer soluciones, documentarlo, priorizarlo con el equipo de ingeniería y asegurar al despliegue que la mejora funcionó.

El análisis inicial de los resultados de las primeras pruebas masivas del cliente de la herramienta obtuvo los siguientes resultados:

- **Safe rate:** solo el 85% de las cotizaciones se consideraban seguras para compartir con los pasajeros, el cual es inaceptable porque significaba que 1 de cada 10 pasajeros recibiría una cotización incorrecta. Este número debería estar cerca del 99%. Entre los motivos principales que causan el 15% de cotizaciones inseguras se encuentran:
  - **Tax calculator:** el 60% de las cotizaciones inseguras se debían a errores en el cálculo de la reembolsabilidad de alguno de los impuestos pagados en el ticket. Esto se debe a la complejidad en las reglas de cada impuesto, que debía ser descargado en formato, analizado con inteligencia artificial y aplicado correctamente. De por si este proceso es complejo, pero hay que añadir que es muy común que las aerolíneas y regiones tengan reglas de negocio particulares por impuesto que son distintas a las reglas publicadas. Esto significaba que debíamos implementar estas reglas de negocio manualmente en el código.
  - **Fare rule error:** el 20% se debían a ciertas reglas tarifarias que nuestra inteligencia artificial no era capaz de extraer la regla de reembolsabilidad, lo que no nos permitía calcular el monto de reembolso. Al igual que las reglas de impuestos, estos están escritos en texto plano por lo que no hay una manera sencilla de interpretarlo a través de un robot. La única mejora posible es identificar estas situaciones y usarlos para entrenar mejor el modelo de extracción.
  - **Historic Pricing error:** el 15% se deben a fallos a la hora de cotizar la tarifa histórica de segmentos volados. Este error solo puede ocurrir en tickets volados parcialmente donde la aerolínea solicita realizar una cotización histórica. Este proceso es complejo debido a que se debe encontrar una tarifa similar al segmento volado, que en muchos no existe, por lo que se deben aplicar reglas de negocio para seleccionar otros tipos de tarifa. La única solución para el futuro es identificar estas reglas e implementarlas continuamente en el código.
  - **Fare Construction error:** el 5% restante se debe a un fallo de interpretación de la construcción tarifaria, que de nuevo es un texto plano que debe ser analizado por un robot. En general la solución es muy buena a la hora de interpretar las construcciones tarifarias, por lo que priorizaría otros tipos de errores.
- **Timeout:** el 40% de las cotizaciones tardan más de 30 segundos. En su mayoría se debe a que para obtener las reglas de impuesto del GDS debíamos consultar uno por uno las reglas de un impuesto (podrían existir cientos de reglas que apliquen a un

vuelo específico) y hasta que leyéramos el contenido de cada regla sabríamos exactamente cual regla aplica para el ticket. Teniendo en cuenta que estas búsquedas se debían hacer por cada impuesto y vuelo es fácil que tuviéramos que descargar miles de reglas en cada cotización. La solución sería tener acceso directamente a la base de datos estructurado de reglas de impuestos, el cual después del proyecto encontramos en el TTBS<sup>45</sup>, una base de datos estructurada que ofrece la IATA. Esta base nos permitiría buscar la regla de impuesto específico por cada impuesto y vuelo, reduciendo la búsqueda de miles a decenas de reglas.

- **Uptime:** esta métrica fue un éxito ya que consiguió un 99.975% de actividad promedio. Dicho esto, el equipo de ventas identificó que hay potenciales clientes que solicitan un Uptime del 99.999% debido a que ese era la tasa de actividad que ofrecían a sus propios pasajeros. Aunque parezca un incremento pequeño, es exponencialmente complejo mejorar esta métrica porque significaría que durante un año la herramienta solo podía estar 5 minutos 13 segundos fuera de servicio. Es fácil imaginar que un pequeño error humano podría causar una baja mucho mayor acumulado dentro del año. Esto significa que en el futuro se deberá fortalecer los procesos de pruebas de calidad, CI/CD y monitoreo para lograrlo.

## Resultados de la metodología

En general la metodología fue exitosa para el momento y la naturaleza del proyecto. Se consiguió una tasa de despliegues de 1 vez cada 2 días, con un promedio de 3 cambios por despliegue, lo que equivale a 108 cambios entre funcionalidades nuevas, mejoras y arreglos durante todo el proyecto. Teniendo en cuenta el tamaño del equipo y el tiempo fue un logro muy grande. Dicho esto, el éxito fue parcial pues mi evaluación es que esta metodología no era sostenible ni escalable en el tiempo por las siguientes razones:

- El backlog del Kanban acabó con decenas de tarjetas que requería implementar desde cambios grandes hasta pequeñas tareas. Esto dificultaba mucho la priorización de tareas ya que se debían tomar en cuenta muchas necesidades, haciendo mas complejo saber que se debía hacer y cuanto era el esfuerzo necesario. Esto también generó mucha complejidad a la hora de agregar nuevos miembros al equipo y a la hora de delegar el proceso de priorización, pues se debía de explicar todo el contexto de las tarjetas existentes.
- Al avanzar tan rápido, se realizaron muchas tareas que deberían haber sido agrupadas en una única tarea, pero esto no era posible porque en general no había tiempo para detenerse y reflexionar. Para evitar esto en el futuro se debía:

---

<sup>45</sup> *Ticket Tax Box Service (TTBS)*. (s. f.). <https://www.iata.org/en/services/finance/ttbs/>

- Construir un administrador de políticas, que sería una interfaz que tanto el cliente como el equipo de operaciones podría usar para configurar reglas de negocio dinámicamente. Sin ella el equipo de ingeniería tuvo que codificar estas reglas lo que les quitó tiempo de otras tareas, además de que desplegar código siempre toma mas tiempo (por el proceso de CI/CD) que cambiar una variable en una interfaz, que debería tomar segundos o minutos.
- Muchas de las tareas fue arreglar casos extremos dentro del cálculo de la reembolsabilidad de los impuestos. Esto es resultado de la complejidad inherente de la existencia de miles de impuestos, cada uno de distintos países y aeropuertos y cada uno con su propia lógica. Nuestra lógica en ese momento era una aproximación que conseguía una precisión de entre el 80-90% pero esto no era suficiente para el volumen que estábamos manejando. Para evitar hacer parches para cada impuesto y caso extremo, en el futuro se tendría que refactorizar la lógica actual para que fuera más completa y fácil de editar por el equipo de operaciones.
- El equipo de ingeniería de Despegar sufrió mucha frustración al implementar nuestra API debido a que lo sacamos en poco tiempo, pero eso significaba que la documentación no era completa ni fácil de seguir. Es un estándar en la industria dedicar mucho tiempo crear un portal de desarrolladores para que los clientes puedan leer documentación completa, clara y exhaustiva de las APIs públicas, además de poder probar desde el propio portal el servicio. Era importante dedicar tiempo a desarrollar este portal para facilitar la integración de los clientes en el futuro.
- Finalmente, debíamos ir más lento en el futuro y generar espacios donde se pudieran proponer funcionalidades que pudieran pasar el trabajo de los ingenieros a los analistas. Creo que los Sprints semanales no son sostenibles y en el futuro se debían incrementar a 2 semanas o 1 mes.
- Debido a la velocidad no tuvimos mucho tiempo de documentar adecuadamente las complejidades de la infraestructura de servidores ni del CI/CD, esto produce una curva de aprendizaje muy grande para los nuevos ingenieros y evita que se aventuren a mejorarlo. Para el futuro se debería seguir la práctica de DevOps llamada Infraestructura como Código<sup>46</sup>, el cual consiste en describir la infraestructura de servidores dentro de código que está versionado. Esto permite que otra herramienta como Terraform<sup>47</sup> leer este código y recrear esta infraestructura. De esta manera estaría explícito como funciona la infraestructura y los nuevos

---

<sup>46</sup> *What is Infrastructure as Code (IaC)?* (s. f.). <https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac>

<sup>47</sup> *Terraform by HashiCorp.* (s. f.). Terraform By HashiCorp. <https://www.terraform.io/>

ingenieros podrían leerlo y modificarlo, sin necesitar manejar directamente esta infraestructura.

- La Programación Extrema requiere de ingenieros con mucha experiencia que puedan hacer grandes cambios rápidamente y manejar al cliente. Esto obviamente no es sostenible en el tiempo por cuatro razones:
  - El proyecto podría detenerse si uno de estos ingenieros necesitara estar en otro proyecto o decidiera renunciar.
  - En el future necesitaremos contratar ingenieros con menos experiencia y necesitamos que puedan colaborar y aprender en los proyectos de la empresa. Se debe cambiar la metodología para permitirles el tiempo necesario para enfrentar estos retos.
  - No es común encontrar ingenieros de software que además tengan habilidades de manejo de clientes ya que son 2 roles completamente distintos. Para escalar se debe separar estos roles lo que facilitaría la contratación de ambos perfiles.
  - Estar completamente involucrado en un Proyecto urgente requiere mucha energía y compromiso que de ser sostenido acabaría quemando emocionalmente al equipo, disminuyendo la productividad y causando renuncias.
- En esta metodología, la responsabilidad de probar el código y monitorear la calidad cae en los ingenieros. Esto es ineficiente ya que la mayoría de su tiempo debería ser aprovechado en pensar y diseñar funcionalidades. Por ello en el futuro es necesario crear un equipo de Control de Calidad (QA o Quality Assurance) quienes tendrían como responsabilidad principal probar exhaustivamente diferentes escenarios dentro de la solución para identificar errores y levantar requerimientos para arreglarlos.
- A pesar de nuestros procesos y mejores deseos, los servidores podían caerse durante la noche o fines de semana debido a factores externos (un GDS podía dejar de funcionar o una credencial podía expirar). Para escalar, soportar cualquier volumen de transacciones y trabajar con las agencias mas grandes del mundo es crítico reducir al mínimo el tiempo en que el sistema está caído. Para el futuro se debía crear un sistema de guardias para que hubiera un administrador de sistema y un ingeniero disponible las 24 horas del día, 7 días a la semana los 365 días del año.

# Conclusión

El proyecto de integración de ARP en Despegar fue un éxito ya que conseguí adaptar la solución a las necesidades de automatización del cliente, incrementando el ingreso mensual en un 60% y liderando un equipo multidisciplinario compuesto por analistas, ingenieros y miembros claves del cliente a través de una metodología de desarrollo de software iterativo, flexible y rápido.

Dicho esto, aprendí que la metodología no era sostenible debido a que dependía de un equipo altamente experimentado y comprometido completamente con el proyecto, lo que no sería posible en el futuro, especialmente si los siguientes proyectos incluían clientes mucho más pequeños, a los que debía servir, pero donde no podía justificar tener los mismos recursos.

También aprendí sobre las limitaciones de la solución, la cual estaba muy lejos de estar aceptable al principio para Despegar, y que seguía sin estar listo para integrarlo con otros clientes de manera rápida. Faltan funcionalidades, manejar casos extremos, tener funcionalidades de administrador que permitieran cambiar reglas de negocio sin requerir al equipo de ingeniería. Cuando acabé el proyecto estaba claro que debía repetir el mismo proceso con otro cliente debido a todo lo que faltaba, en el futuro se debía desarrollar mucho más el producto para que la integración fuera en menos de una semana.

Finalmente, me di cuenta de que nuestro equipo actual era muy pequeño y faltaba procesos claves para poder escalar a tener muchos más clientes de manera exitosa. Debíamos profesionalizar los procesos de calidad, tener ingenieros en guardia, reducir la curva de aprendizaje y complejidad en el manejo de la infraestructura, además de crear una metodología de trabajo que resultara menos exhausta para los colaboradores.

Considero que, a través del proyecto de integración de Despegar, he demostrado contar con los conocimientos, habilidades, y actitudes que son propios del ingeniero industrial. A lo largo de este trabajo he mostrado que cumplo los objetivos educacionales, los atributos y perfil del egreso desde mi experiencia profesional como Director de Operaciones.



# Bibliografía

- <sup>1</sup> *Travel and Tourism: Contribution to Global GDP 2023* | Statista. (2023, 7 noviembre). Statista. <https://www.statista.com/statistics/233223/travel-and-tourism--total-economic-contribution-worldwide/>
- <sup>2</sup> Fernández-Paniagua, A. M. del C. (2017). *¿Qué son los Softwares GDS y para qué sirven?* Thinking for Innovation. <https://www.iebschool.com/blog/software-gds-marketing-digital/>
- <sup>3</sup> Editor. (2019, 12 abril). History of flight booking: CRSs, GDS distribution, travel agencies, and online reservations. *AltexSoft*. <https://www.altexsoft.com/blog/history-of-flight-booking-crss-gds-distribution-travel-agencies-and-online-reservations/>
- <sup>4</sup> Raymond, L., & Bergeron, F. (1997). Global distribution systems: A field study of their use and advantages in travel agencies. *Journal of Global Information Management (JGIM)*, 5(4), 23-32.
- <sup>5</sup> Jara, A. (n.d.). *Digital transformation for the travel industry*. Deal Engine. <https://deal-engine.com/>
- <sup>6</sup> Statista. (2022, 20 septiembre). *Latin America: Market share of main online travel agencies 2019*. <https://www.statista.com/statistics/1235726/latin-america-leading-online-travel-platforms/>
- <sup>7</sup> *What is Software as a Service (SaaS)?* (n.d.). Salesforce.com. <https://www.salesforce.com/mx/saas/>
- <sup>8</sup> Grant, Mi. (2021, September 15). *What You Should Know About Startups*. Investopedia. <https://www.investopedia.com/terms/s/startup.asp>
- <sup>9</sup> Dawes, J. (2022, November 18). *Deal Engine Raises \$5.3 Million to Help Airlines Save Post-Booking*. Skift. <https://skift.com/2022/11/18/deal-engine-raises-5-3-million-to-help-airlines-save-post-booking-travel-startup-funding-this-week/>
- <sup>10</sup> *Connect to Amadeus Travel APIs | Amadeus for Developers*. (s. f.). Amadeus IT Group SA. <https://developers.amadeus.com/blog/what-is-pnr-booking-reference>
- <sup>11</sup> *How to retrieve/display a PNR (Cryptic) - Amadeus Service Hub*. (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/453392470/como-recuperar/ver-un-pnr-criptico-#>
- <sup>12</sup> *Amadeus Customer Profiles: How to create a profile from a PNR (Cryptic) - Amadeus Service Hub*. (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/838421/perfiles-de-clientes-de-amadeus-como-crear-un-perfil-a-partir-de-un-pnr-criptico->
- <sup>13</sup> *How to display the PNR History (CRYPTIC) - Amadeus Service Hub*. (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/813131/como-mostrar-el-historial-pnr-criptico->
- <sup>14</sup> Price, S. A. (s. f.). *Amadeus Central Ticketing Reissue/Exchange*. ppt download. <https://slideplayer.com/slide/16156344/>

- <sup>15</sup> *How to display an e-ticket record (Cryptic) - Amadeus Service Hub.* (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/939515/como-mostrar-un-registro-de-boleto-electronico-cryptic->
- <sup>16</sup> *Fare calculation symbols and explanations - Amadeus Service Hub.* (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/774026/fare-calculation-symbols-and-explanations>
- <sup>17</sup> *Fare Rules: Category description - Amadeus Service Hub.* (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/926104/condiciones-de-tarifa-descripcion-de-la-categoria>
- <sup>18</sup> Ticketing System. (2018, 25 junio). *Sabre #8 | HOW TO CHECK PENALTY REFUND TICKET AND NOSHOW PENALTY ON SABER GDS ABECUS* [Video]. YouTube. <https://www.youtube.com/watch?v=MhOahB6R8Ck>
- <sup>19</sup> *Our story.* (s. f.). [https://www2.arccorp.com/about-us/our-story/?utm\\_source=Global\\_Navigation](https://www2.arccorp.com/about-us/our-story/?utm_source=Global_Navigation)
- <sup>20</sup> *Billing and Settlement Plan (BSP).* (s. f.). <https://www.iata.org/en/services/finance/bsp/>
- <sup>21</sup> *How to refund a ticket (Cryptic) - Amadeus Service Hub.* (s. f.). <https://servicehub.amadeus.com/c/portal/view-solution/961528/como-reembolsar-un-billete-criptico->
- <sup>22</sup> Chen, J. (2022, November 3). *Business to Business (B2B).* Investopedia. <https://www.investopedia.com/terms/b/btob.asp>
- <sup>23</sup> *Beta test.* (2021, 2 septiembre). <https://www.productplan.com/glossary/beta-test/>
- <sup>24</sup> Team, A. C. (s. f.). *Waterfall Methodology: Project Management | Adobe Workfront.* <https://business.adobe.com/blog/basics/waterfall>
- <sup>25</sup> Agile Manifesto. (2001). *Principles behind the Agile Manifesto.* Agilemanifesto.org. <https://agilemanifesto.org/principles.html>
- <sup>26</sup> *Top 7 Agile frameworks.* (s. f.). <https://premieragile.com/types-of-agile-frameworks/>
- <sup>27</sup> Martins, J. (2024, 19 enero). ¿Qué es la metodología Kanban y cómo funciona? [2024] • Asana. *Asana.* <https://asana.com/es/resources/what-is-kanban>
- <sup>28</sup> Raeburn, A. (2024b, febrero 13). ¿Qué es la programación extrema (XP)? [2024] • Asana. *Asana.* <https://asana.com/es/resources/extreme-programming-xp>
- <sup>29</sup> *What is Scrum?* (s. f.). Scrum.org. <https://www.scrum.org/resources/what-scrum-module#:~:text=Scrum%20helps%20people%20and%20teams,optimize%20for%20their%20specific%20needs.>
- <sup>30</sup> Rehkopf, B. M. (s. f.). *Scrum Sprints: Everything You Need to Know | Atlassian.* Atlassian. <https://www.atlassian.com/agile/scrum/sprints>

- <sup>31</sup> *¿Qué es DevOps? Explicación de DevOps | Microsoft Azure.* (s. f.). <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>
- <sup>32</sup> Atlassian. (s. f.). *What is version control | Atlassian Git Tutorial.* <https://www.atlassian.com/git/tutorials/what-is-version-control>
- <sup>33</sup> *Git.* (s. f.). <https://git-scm.com/>
- <sup>34</sup> *The most-comprehensive AI-powered DevSecOps platform.* (s. f.). GitLab. <https://about.gitlab.com/>
- <sup>35</sup> *What is CI/CD?* (s. f.). <https://www.redhat.com/en/topics/devops/what-is-ci-cd>
- <sup>36</sup> Mireles, Y. (2024, 24 enero). *What is observability? New Relic.* <https://newrelic.com/blog/best-practices/what-is-observability>
- <sup>37</sup> Grafana Labs. (s. f.). *Grafana: The open observability platform | Grafana Labs.* <https://grafana.com/>
- <sup>38</sup> Prometheus. (s. f.). *Prometheus - Monitoring system & time series database.* <https://prometheus.io/>
- <sup>39</sup> *Metabase | Business Intelligence, Dashboards, and Data Visualization.* (s. f.). Metabase | Business Intelligence, Dashboards, And Data Visualization. <https://www.metabase.com/>
- <sup>40</sup> *¿Qué es una API? - Guía sobre las API para principiantes - AWS.* (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/api/>
- <sup>41</sup> Fernández-Paniagua, A. M. del C. (2017). *¿Qué son los Softwares GDS y para qué sirven? Thinking for Innovation.* <https://www.iebschool.com/blog/software-gds-marketing-digital/>
- <sup>42</sup> *Distribution with Offers & Orders (New Distribution Capability).* (n.d.). Wwww.iata.org. <https://www.iata.org/en/programs/airline-distribution/retailing/ndc/>
- <sup>43</sup> *How to Use the STAR Method for Behavioral Interviews.* (n.d.). Recruitment. <https://recruitment.com/process/star-interview-questions>
- <sup>44</sup> *IA abierta.* (2023). ChatGPT (versión del 14 de marzo) [Modelo de lenguaje grande]. <https://chat.openai.com/chat>
- <sup>46</sup> *Ticket Tax Box Service (TTBS).* (s. f.). <https://www.iata.org/en/services/finance/ttbs>
- <sup>46</sup> *What is Infrastructure as Code (IaC)?* (s. f.). <https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac>
- <sup>47</sup> *Terraform by HashiCorp.* (s. f.). Terraform By HashiCorp. <https://www.terraform.io/>