

Desarrollo de aplicación mobile de INFERNO®

16 de febrero de 2016

UNIVERSIDAD NACIONAL DE QUILMES
Departamento de Ciencia y Tecnología
Tecnicatura Universitaria en Programación Informática

Trabajo de Inserción Profesional

Javier Presti

Director: Leonardo Gassman, Ing.

Índice

| | |
|--|-----------|
| 1. Introducción | 4 |
| 2. Dominio | 5 |
| 2.1. Carta | 5 |
| 2.2. Edición | 6 |
| 2.3. Mazo | 6 |
| 2.4. Formato | 6 |
| 2.5. Erratas, limitaciones y prohibiciones | 7 |
| 2.6. Reimpresiones | 7 |
| 2.7. Habilidades con palabras clave | 7 |
| 2.8. Ficha | 7 |
| 2.9. Cartas con doble cara | 8 |
| 3. Modo de trabajo | 8 |
| 4. Alcance | 8 |
| 4.1. Búsqueda de cartas | 8 |
| 4.2. Listas de cartas | 9 |
| 5. Diseño de software | 10 |
| 5.1. Arquitectura | 10 |
| 5.2. Carga de datos | 10 |
| 5.3. Múltiples dispositivos | 11 |
| 5.4. Visualización | 12 |
| 5.5. Tipos de buscadores | 13 |
| 5.6. Listado | 15 |
| 5.7. Creación de una lista | 15 |
| 6. Implementación | 16 |
| 6.1. Iteración 1: Carga de datos y Visualización de cartas | 16 |
| 6.1.1. Carga de datos | 16 |
| 6.1.2. Manipulación de imágenes | 16 |
| 6.1.3. Visualización de cartas | 17 |
| 6.1.4. Compartir imagen | 18 |
| 6.2. Iteración 2: Buscadores de cartas | 18 |
| 6.2.1. Buscador básico | 19 |
| 6.2.2. Buscador extendido | 19 |
| 6.2.3. Buscador avanzado | 20 |
| 6.3. Iteración 3: Listas de cartas y validación de mazos | 20 |
| 6.3.1. Listado de cartas | 20 |
| 6.3.2. Creación y validación de listas | 20 |
| 6.4. Iteración 4: Puesta en producción e informe | 22 |
| 7. Trabajo futuro | 22 |

| | |
|-----------------------------------|-----------|
| 8. Conclusiones | 23 |
| A. Campos de una carta | 25 |
| B. Prueba de casos de uso | 27 |
| B.1. Búsqueda de cartas | 27 |
| B.2. Listas de cartas | 30 |

1. Introducción

INFERNO® es un juego nacional de cartas colecccionables (CCG). En los CCG's un jugador debe comprar cartas y combinarlas para armar su propio mazo que competirá contra el de otros jugadores. La elección de cartas a utilizar en un mazo es la que define la estrategia y la forma de jugar con el mismo. Este tipo de juego se renueva constantemente ya que nuevas cartas se van creando comúnmente a través de expansiones o ediciones, que son paquetes que extienden al juego actual. Las cartas se consiguen oficialmente a través de mazos pre-armados o de sobres con contenido aleatorio. Los mismos están disponibles en comiquerías o a través de distribuidores particulares.

El CCG nació con Magic the Gathering® en el año 1993 y que es desde entonces popular. Si bien en un comienzo sólo existía como un juego físico, en los últimos años el género se benefició de un auge con el advenimiento de una gran cantidad juegos digitales¹, muchos de ellos pensados para smartphones. Actualmente el mayor exponente de esta movida digital es Hearthstone, que ya cuenta con más de 40 millones de jugadores registrados².

El juego INFERNO fue creado en Mendoza por la empresa argentina OMG! entretenimiento que comercializa su producto a todo el país y organiza -entre otras cosas- eventos provinciales y nacionales del juego. Cuenta en la actualidad con más de 1000 cartas diferentes distribuidas entre 8 ediciones y con más planeadas para lanzar este año. Este juego se distribuye, al menos por el momento, exclusivamente de manera física. Una herramienta digital para ver y administrar los elementos del juego (cartas, mazos) abriría las puertas a que el producto nacional crezca en popularidad e incentive más a los actuales jugadores.

Un CCG tiene dos grandes partes importantes: la creación del mazo y su uso (o sea, la acción de "jugar"). Para crear un mazo, se deben buscar cartas potencialmente útiles para él. Esto es un trabajo que requiere cierto tiempo, y es aquí donde un software de búsqueda se vuelve muy práctico. Con los filtros adecuados, se pueden armar mazos más fácil y rápidamente y también se pueden descubrir estrategias que finalmente otorguen una posición más favorable para el juego. Asimismo, el poder ir almacenando y guardando mazos y listas de cartas de manera digital le permite al jugador tener una mayor organización y control de sus posibilidades de juego. Crear listas que luego, en muchos casos, se transforman a mazos para jugar es el paso natural que viene luego de la búsqueda de cartas y antes de la compra física de las mismas.

Con respecto a las plataformas utilizadas para el desarrollo de software, hoy en día se está dando mucho en el ámbito mobile ya que las personas usan cada vez más el teléfono y menos la PC. A su vez, Android es el sistema operativo más utilizado en teléfonos móviles³, y más aún en Argentina, donde tiene más

¹<https://playfab.com/blog/the-numbers-behind-the-rise-of-collectible-card-games> [Consulta: 14 febrero 2016]

²<http://www.gamespot.com/articles/hearthstone-reaches-40-million-players-up-10-milli/1100-6432063> [Consulta: 14 febrero 2016]

³http://gs.statcounter.com/#mobile_os-ww-monthly-201601-201601-bar [Consulta: 14 febrero 2016]

del 80 % de cuota entre smartphones⁴. Por esta razón es que Android es una fuerte elección al realizar una aplicación que complemente un juego de cartas en Argentina.

Este informe presenta el Trabajo de Inserción Profesional que da fin a la carrera de Técnico Universitario en Programación Informática. Teniendo el consentimiento del dueño, y dados los puntos anteriormente expuestos, se realizó una aplicación del juego INFERNO para la plataforma Android. El objetivo del software es la búsqueda de cartas y la creación de listas y mazos. **La finalidad del mismo es ser una demostración para obtener financiamiento** para desarrollar una aplicación completa del juego, o proyectos similares sobre otros CCGs. Se detallará las partes de trabajo, el alcance del mismo, su diseño, arquitectura y las herramientas utilizadas, haciendo énfasis en las particularidades técnicas y dificultades encontradas y en cómo se las solucionó. El documento finalizará con las conclusiones obtenidas de este trabajo.

2. Dominio

INFERNO es un juego de cartas colecciónables en donde un jugador arma un mazo de cartas para jugar contra el mazo de otros. El tipo de juego puede considerarse de estrategia, ya que requiere de mucho planeamiento para seleccionar las cartas y luego usarlas en el juego.

En una partida se enfrentan comúnmente 2 jugadores, cada uno con su propio mazo. Al comenzar mezclan sus mazos, toman 5 cartas y comienzan con el primer turno. Al comienzo del turno de cada jugador, él toma una carta y pone la siguiente del tope de su mazo boca abajo en la mesa. Las cartas boca abajo se utilizan como recurso, es decir para pagar el coste de las cartas. El jugador durante su turno juega cartas de su mano y/o habilidades de cartas en juego (como atacar) hasta que decide terminarlo, comenzando el turno del siguiente jugador. Atacar a un jugador es sacarle cartas del mazo. Cuando un jugador no tiene más cartas en el mazo y debe sacar una, pierde.

A continuación se presentarán los conceptos más relevantes de INFERNO.

2.1. Carta

Elemento principal del juego. Cada carta posee características únicas que la diferencian de las demás. En INFERNO pertenecen a una senda -representada como su reino de origen- y se dividen en diferentes tipos, donde cada uno cumple un rol diferente dentro del juego (por ejemplo, una carta de tipo “Arma” es aquella que se equipa en otra de tipo “Aliado”). En general se dice que no hay mejores o peores cartas, sino que son diferentes. Sin embargo, se suele decir que una es más “útil” que otra. Es decir, que es más probable que un jugador querrá incluirla en un mazo. Para ver los campos que contiene, ver anexo A.

⁴http://gs.statcounter.com/#mobile_os-AR-monthly-201601-201601-bar [Consulta: 14 febrero 2016]

2.2. Edición

El juego periódicamente (aproximadamente cada 6 meses) lanza una nueva edición con unas 90 a 180 cartas. Éstas se ponen a la venta en sobres cerrados (llamados “boosters”) que contienen cartas de diferentes rarezas con contenido aleatorio. En algunas ocasiones también se venden mazos pre-armados, es decir, con contenido establecido y de público conocimiento. En general las últimas 2 ediciones son las que se consideran “en curso” y que pueden ser conseguidas en un local de venta. Es decir, la edición que se encuentra en producción y el remanente de la anterior. Cartas de ediciones más antiguas en general se consiguen solamente de manera individual y no en boosters.

2.3. Mazo

Conjunto de cartas que pertenece a al menos un formato en particular y por lo tanto puede ser jugado en él. Crear un mazo es armar una estrategia de juego. Cada jugador posee el suyo con el que enfrentará al de otro jugador del mismo formato. Se buscan cartas que entre ellas se complementen o potencien y se define la manera de jugar. Las reglas de creación de mazo están definidas por los formatos. Un jugador comúnmente tiene más de un mazo armado y listo para jugar.

2.4. Formato

Define con qué cartas se puede armar el mazo, principalmente con qué ediciones. También puede establecer restricciones con el uso de alguna carta en particular. Algunos formatos a su vez modifican las reglas de armado de mazo, permitiendo por ejemplo la inclusión de 2 sendas.

En los formatos comúnmente utilizados un mazo debe contener una sola carta especial de tipo Demonio y 50 cartas más que solamente pueden ser de la misma senda que el Demonio y neutrales. A su vez, entre esas 50 cartas no puede haber más de 4 copias de una misma.

A continuación se detallan los formatos con los que se juega actualmente:

Base El más utilizado por los jugadores. Permite el uso de las últimas ediciones únicamente. En este momento comprende a las últimas 4 que son Descenso, Encuentro, Abismo y Origen.

Eterno Comprende todas las ediciones. Se encuentran prohibidas las cartas de ediciones promocionales, es decir ediciones que en general contienen una sola carta y que se lanzan para promover algún evento.

VaP (Viaje a las Profundidades) Conformado por las ediciones Descenso, Encuentro y Abismo.

CdU (Crónicas de Ubanna) Comprende las ediciones Insania, Conquista y Destino. Se encuentran limitadas las cartas Ascenso de Narghot y Estatua Viviente.



Figura 1: Carta reimpressa

2.5. Erratas, limitaciones y prohibiciones

A veces las cartas sufren actualizaciones en sus valores. Esto es conocido como errata. Se publican oficialmente y deberían ser conocidas por los jugadores. Prácticamente siempre al generarse la errata, la carta ya ha sido impresa en su totalidad y por lo tanto todas las copias de la misma presentan el valor modificado desactualizado.

2.6. Reimpresiones

Al salir una nueva edición es posible que algunas de las cartas que la conforman hayan pertenecido también a ediciones anteriores. A esto se le llama reimpresión. Al ser reimpressa una carta puede variar en su ilustración (y por ende su artista), su rareza y es posible que el texto de la carta sufra actualizaciones como una nueva redacción o inclusión de errata. Ver figura 1.

2.7. Habilidades con palabras clave

Algunas habilidades de las cartas son tan comunes que se les pone un nombre. Cuando se hace, en un comienzo se lo muestra en la carta junto con la descripción de la habilidad pero en ediciones posteriores aparece sólo el nombre, funcionando éste como término clave.

2.8. Ficha

Su único propósito es la de representar un elemento dentro del juego. Si bien una ficha se representa en una carta, técnicamente no lo es a los fines de crear un mazo y por tanto no puede ser incluida en él.

2.9. Cartas con doble cara

Las cartas de tipo Demonio poseen valores en sus 2 caras y pueden ser volteadas durante el juego. Este es el único tipo de carta con esta característica y no es mezclada en el mazo.

3. Modo de trabajo

La creación de la aplicación se dividió en iteraciones incrementales. En total tomó 4 iteraciones en donde la segunda se extendió una semana adicional debido a las complejidades que se encontraron a la hora de realizarla. A pesar de esto se pudo cumplir satisfactoriamente con las iteraciones planificadas debido a la fuerte carga horaria que se ha podido asignar como ya estaba previsto.

El objetivo del presente trabajo es presentar una demo de cara a obtener futura financiación. Entonces, el desarrollo se centró en proveer pocas funcionalidades pero que sean completas y tengan un alto atractivo visual y de usabilidad. Es por esto que no estaba planeada la realización de tests automáticos para la presente aplicación, ya que los mismos requieren de horas de programación y no sirven para garantizar los aspectos visuales y de usabilidad que busca esta demo. Por lo tanto se decidió utilizar un mecanismo manual de testeo. Se consideró -al ser una demo- importante que el producto se vea bien y que sea muy práctico para su uso así como que sea completo.

Las iteraciones involucradas se detallan a continuación:

| Iteración | Duración | Objetivo |
|-----------|-----------|--|
| 1 | 2 semanas | Carga de datos y visualización de cartas |
| 2 | 3 semanas | Buscadores de cartas |
| 3 | 2 semanas | Listas de cartas y validación de mazos |
| 4 | 3 semanas | Puesta en producción e informe |

4. Alcance

Los diferentes casos de uso se pueden detallar dentro de 2 grandes categorías que son la búsqueda de cartas y la creación de listas. Las indicaciones para probarlos está dadas en el anexo B.

4.1. Búsqueda de cartas

Para una explicación de los detalles de una carta ver anexo A.

1. Poder buscar una carta según:

- Nombre
- Texto que contiene
- Senda a la que pertenece

- Tipo de carta
 - Edición a la que pertenece
 - Rareza
 - Valores numéricos de Coste/Fuerza/Resistencia
 - Artista de la imagen
 - Formato en el que es válida
 - Habilidades con palabras clave
2. Incluir o excluir las cartas de tipo ficha de la búsqueda
 3. Listar el resultado de la búsqueda:
 - Listado sencillo (con las imágenes de las cartas)
 - Listado con descripción detallada
 4. Aplicar criterios de orden a los resultados según los campos de las cartas
 5. Ver el detalle de una carta buscada:
 - Imagen en tamaño completo y en buena calidad
 - Datos completos de la carta
 - Ediciones en las que fue impresa (detallando la rareza que posee en cada una) y la visualización de las mismas
 - Formatos en los que es válida
 - Descripción de habilidades con palabra clave
 - Cambios en la carta (erratas, limitaciones y prohibiciones)
 - Visualización de ambas partes de la carta donde aplique(2.9)
 6. Compartir una carta en las redes sociales o aplicaciones de mensajería configuradas en el dispositivo

4.2. Listas de cartas

1. Crear lista
2. Agregar carta a lista
3. Ver detalle de cartas, con cantidad de cada una y total
4. Editar o eliminar cantidad de una carta
5. Eliminar lista
6. Dada una lista, saber si cumple con las condiciones para ser mazo y en qué formatos lo es

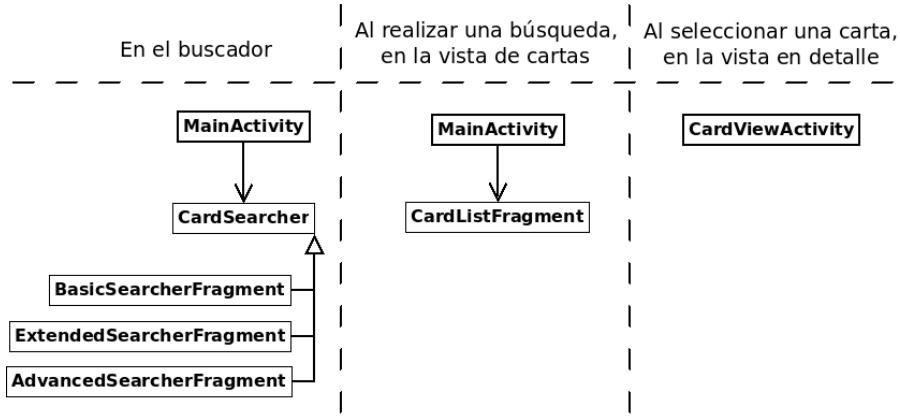


Figura 2: Diagrama simplificado de los diferentes estados en la búsqueda de una carta

5. Diseño de software

5.1. Arquitectura

Se utiliza un Activity que es el encargado de ir cargando los diferentes Fragments. Un Activity en Android tiene un concepto similar al de ventana o marco en el que la aplicación funciona. Puede contener Fragments que suelen ser contenedores de componentes que también poseen comportamiento.

En esta aplicación todas las pantallas que el usuario ve son fragments que van siendo reemplazados a excepción de la vista de una carta, que es otro Activity.

5.2. Carga de datos

Se decidió crear 2 planillas de cálculo con las diferentes características de las cartas. Por “carta” se entiende a aquella que tiene nombre diferente a otra, con excepción de las cartas tipo ficha que se las diferencia por los valores de sus campos. Es decir, si una carta fue impresa en más de una edición, para el programa sigue tratándose de la misma carta (ver 2.6). Esto fue una decisión de diseño.

Ya que al ser reimprima en una edición diferente hay valores de la carta que son o pueden ser diferentes (edición, rareza, artista) se decidió crear una planilla para los datos inmutables de una carta y otra donde se detallan aquellos campos que pueden variar. La información de las planillas se vuelcan a una base de datos embebida en la aplicación. Al momento de consultar por los datos se realizarán los join correspondientes de las tablas.

Se decidió otorgarle un id a cada carta, edición e instancia de carta, que es la conjunción de los anteriores 2 ids. El primero (al que llamaremos “id interno”) es un número de 4 dígitos incremental comenzando en 1001 y el segundo contiene 3 dígitos comenzando por el 101. El id de una carta en la tabla termina siendo la

| id | id interno | id edición |
|---------|------------|------------|
| 1001101 | 1001 | 101 |
| 1002101 | 1002 | 101 |
| 1002114 | 1002 | 114 |
| 1003101 | 1003 | 101 |

Cuadro 1: Primeros ids de las cartas

concatenación de los dos primeros, identificándose por ejemplo la primer carta con el 1001101. Si bien se pudo haber usado números como “0001”, al momento de guardarlo en una tabla y manejarlo habría que tener precauciones extras ya que se almacenaría simplemente como un 1. Forzando a que los primeros ids tengan 4 y 3 dígitos se asegura que una carta esté identificada con un número de 7 dígitos y que siempre sean diferentes.

Al utilizar este id de 7 dígitos las cartas mantienen su orden cronológico al quedarse ordenadas primero por su id interno y luego por su edición. Por ejemplo: la edición con id 114 reedita la carta con id 1002 que sólo había sido editada en la edición con id 101. La reedición tendrá entonces el id 1002114, número que se encuentra entre 1002101 (la carta reeditada) y 1003101 (la siguiente carta de la edición 101). Ver tabla 1.

Con respecto a las imágenes de las cartas, se las quiso nombrar de alguna manera que identifique su contenido, es decir que contenga la edición y el número de carta. Dado que el id de una carta de 7 dígitos mencionado anteriormente posee esta información, se lo puede incluir en el nombre de la imagen. La ventaja de esto es que podemos vincular a una carta directamente con su imagen, sin necesidad de consultar por un campo que contenga el nombre con el que se haya guardado.

Por las razones mencionadas anteriormente es que se decidió utilizar este identificador de carta como el id de los registros de la tabla. Es importante destacar que entonces los ids no son consecutivos, pero sí incrementales.

Se creó un campo para guardar el tipo completo de una carta y otro más por cada subcategoría (ver Tipos en A). Esto favorece la posterior tarea de mostrar datos y de búsqueda respectivamente. Otros campos que se “dividieron” por estos mismos motivos fueron los de coste, fuerza, resistencia, nombre, texto, leyenda y habilidad. Las razones se explican en la sección 6.2.

5.3. Múltiples dispositivos

Se planteó preparar la aplicación para su correcta visualización en múltiples dispositivos. Esto involucra tanto proveer diferentes resoluciones de los iconos como realizar diferentes diseños de pantalla. Las resoluciones provistas para todos los iconos corresponden a dispositivos con densidad de pantalla desde los 160 a los 480 dpi aproximadamente.

Todos los textos de la aplicación se guardan como recurso en el archivo strings.xml, de manera de posibilitar una fácil localización a futuro.



Figura 3: Vista vertical y horizontal de una carta

5.4. Visualización

Con respecto a los diseños de pantalla, se crearon diferentes layouts para las vistas portrait (uso vertical típico en smartphones) y landscape (orientación horizontal común en tablets). Específicamente se organizaron los elementos de la pantalla de diferente manera según la orientación (ver figura 3).

Para realizar una fácil carga de imágenes se nombró a cada una con las letras “zf” seguidas del id de la carta. Entonces, la carta con id 1005101 tendrá la imagen asociada zf1005101.jpg. Si la imagen es del reverso (ver 2.9) se le agrega una “r” al final (por ejemplo, zf1005101r.jpg). De esta manera dado un id es sencillo obtener el nombre de la imagen asociada sin necesidad de realizar ninguna consulta a la base de datos.

Al mostrar una carta, lo más importante y lo que más impacta visualmente es su imagen. Sin embargo, en pantallas con tamaño reducido puede ser incómodo leer las cartas, y el diseño de las mismas no siempre ayuda. En parte por esto es que se consideró importante también detallar en texto los campos de la carta. El propósito de incluirlos es el de proveer fácil lectura de los mismos. Teniendo esto en mente es que se decidió utilizar un color negro sólido de fondo y usar letras blancas de un tamaño generoso. En la vista vertical se decidió que la imagen ocupase toda la pantalla y los textos debajo y en la vista horizontal lado a lado (ver figura 3).

Como se vio anteriormente, una carta puede pertenecer a múltiples ediciones. En vez de mostrarse cada una por separado (como si fuesen diferentes cartas) se decidió dentro de la vista de carta detallar y proveer accesos a las otras ediciones en las que se haya impreso. Por un lado se evitan resultados duplicados al realizar búsquedas y por el otro se enriquece la información proveída de una carta.

Hay cartas que poseen también imagen al dorso. Mostrar ambas imágenes a la vez ocuparía un gran espacio y por eso se decidió que al tocar una imagen con reverso, ésta lo muestre.

Para aquellas habilidades con palabra clave se crean atajos que nos conducen

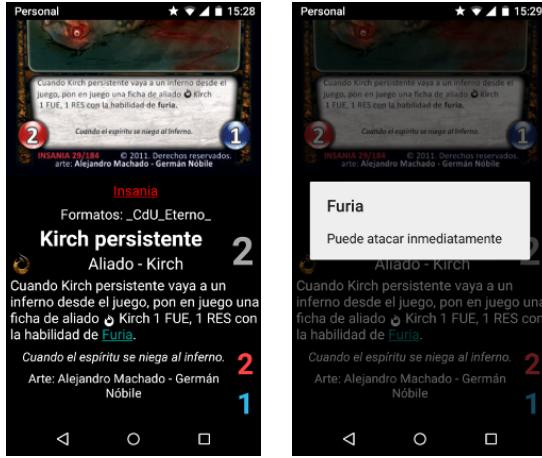


Figura 4: Detalles de una carta

cirán a su descripción. Esto es práctico ya que comúnmente no se leen estas descripciones porque los jugadores ya reconocen los términos clave pero en caso de necesitarse están al fácil alcance. A su vez, los nombres de las sendas son reemplazados por sus iconos, así como aparecen en las cartas. Ver ejemplo en la figura 4.

Inferno cuenta en la actualidad con 20 habilidades con término clave y se van sumando con cada nueva edición. Éstas pueden estar incluidas en el campo texto de una carta y al encontrarse uno de ellos mientras se está viendo el contenido de una carta, es necesario incluir su definición. Para encontrarlas eficientemente el campo texto se guarda codificado en la base de datos. Es decir, se incorporan ciertos símbolos que serán leídos para decodificar su contenido y que serán eliminados al mostrarse en la vista. Por ejemplo, en vez de guardar en la base de datos dentro del campo texto el término clave “Furia” se lo persiste como “#Furia#”. De esta manera el programa intentará encontrar el carácter “#” en busca de palabras clave y una vez encontrado se realizará una búsqueda del término exacto que se encuentra entre los símbolos especiales. Se realiza de esta manera en vez de consultar en la tabla por las habilidades con término clave ya que no sólo es necesario buscarlas, sino ver también en qué parte del texto se encuentran, y persistir esto puede generar problemas ya que los textos de las cartas pueden variar al ser reeditadas.

El mismo procedimiento de buscar un símbolo especial en el texto se sigue para encontrar los términos de las sendas que serán cambiados por su imagen.

5.5. Tipos de buscadores

Se decidió dividir las búsquedas en 3 instancias diferentes de buscadores que las llamamos básica, extendida y avanzada (ver figura 5). La razón de esto fue no sobrecargar la pantalla con opciones que el usuario no necesita.

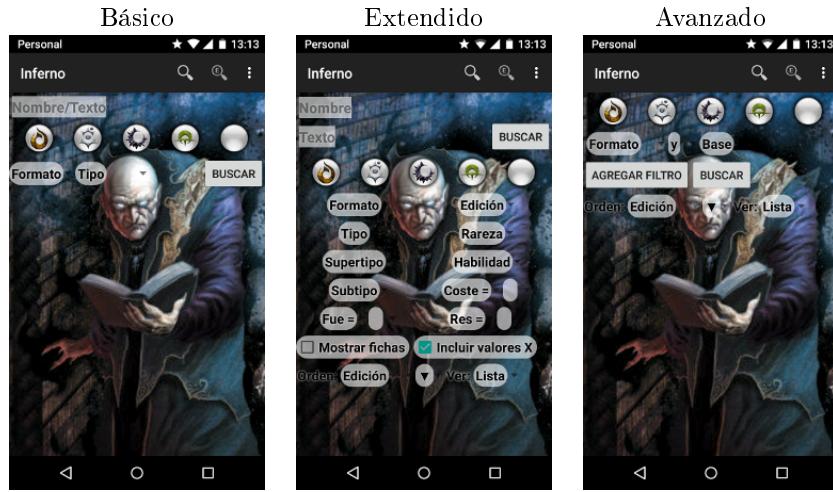


Figura 5: Tipos de buscadores

Al abrirse la aplicación en el buscador, aquí es donde se puso el mayor esfuerzo en lo visual y la usabilidad. Ésta es la puerta de entrada y con lo que los usuarios más van a interactuar, de ahí su importancia. Con respecto a componentes gráficos, se incluyó una imagen de fondo con la ilustración de alguna carta al azar y para generar el mayor impacto se notó que lo mejor era no modificarla agregándole transparencia y retocar en cambio los demás elementos de la pantalla. Finalmente se optó por hacer éstos con fondo blanco con cierta transparencia. Ver comparaciones en figura 6.

El buscador básico es el que se abre por defecto. Se considera que en general con unos pocos campos el usuario estará satisfecho y no es necesario sobrecargar la pantalla de tantos elementos. Con las opciones más importantes la aplicación se ve más elegante y destaca más la imagen de fondo, dándole un mayor atractivo visual. En este buscador se excluyen de los resultados las fichas ya que difícilmente un usuario esté interesado en ellas.

El buscador extendido se utiliza para poder realizar una búsqueda pudiendo utilizar prácticamente todos los campos, a excepción de algunos muy puntuales (artista y leyenda) por los que difícilmente el usuario quiera buscar. Estos campos sólo están disponibles en el último tipo de búsqueda.

El buscador avanzado está pensado para realizar búsquedas muy puntuales. Este último es el más potente pero también el más complicado de usar, siendo la idea ir agregando condiciones a través de premisas y/o/no, teniendo una estructura similar a la que terminará siendo la consulta a realizar. Primero se selecciona un campo y el tipo de condición (y/o/no), y luego el valor a buscar. En aquellos campos en donde se introduce texto (nombre, texto, artista, leyenda) se agrega la posibilidad de realizar una búsqueda parcial (si contiene), total (texto exacto) o si comienza con el término ingresado. Una vez introducidos todos los datos necesarios de un filtro, se selecciona la opción para agregarlo y se agrega al



Figura 6: Comparativa de imágenes de fondo

listado de filtros de la parte inferior de la pantalla. Esta lista es posible editarla eliminando alguno de sus elementos.

5.6. Listado

Se tendrán dos tipos diferentes de listado: grilla y lista. La primera mostrará las imágenes de las cartas y la segunda un resumen de las mismas con sus datos más relevantes, entre ellos su ilustración. Se pensó en tener los artes guardados de manera separada para no tener que andar recortando las imágenes mientras se mostraban los detalles de las cartas. Sin embargo, esto duplicaría la cantidad de imágenes y con ellas el peso de la aplicación se vería muy afectado. Es por eso, y tras comprobar que el recorte funciona muy eficientemente, que se decidió no guardar por separado las ilustraciones. El buscador básico mostrará el listado en lista y los demás darán la opción para elegir el tipo de listado.

5.7. Creación de una lista

Primero se crea y después se agregan elementos desde la vista de carta. Al elegir una lista nos muestra el detalle de las cartas con su cantidad y al seleccionar una vemos todos sus detalles y podemos editar la cantidad. Mientras que se ve el detalle de carta por carta es útil ir viendo cuántas de ellas se encuentran en la lista. Para esto en el Action Bar se decidió crear un botón para cambiar la cantidad o eliminar la carta que se muestra actualmente y el ícono de este botón es la representación numérica de la cantidad actual que se encuentran en la lista. Es decir, cada vez que se cambia la carta que se está viendo, cambia a su vez el ícono de este botón.

6. Implementación

6.1. Iteración 1: Carga de datos y Visualización de cartas

Antes de poder realizar búsquedas de cartas, es necesario tener la base de datos cargada. Esto implica no sólo crear una base de datos e introducirle datos, sino persistirla y luego ver modos de acceso a la misma.

El segundo paso que se realizó fue poder ver los datos de una carta.

6.1.1. Carga de datos

Las tablas se crearon desde planillas de cálculo y se guardaron en una base de datos SQLite⁵ dentro de la aplicación. Esto se realizó utilizando el software Sqliteman⁶.

Para mejorar la velocidad de las consultas se crearon algunos índices con los campos nombre, senda, formato, tipo básico, edición, rareza e id interno. Éstos cubren la gran mayoría de las consultas que se vayan a realizar por los usuarios. El uso de índices haría más lentas las operaciones de update e insert, pero éstas no son utilizadas.

6.1.2. Manipulación de imágenes

Se obtuvieron todas las imágenes de las cartas y se las renombró con el software pyRenamer⁷. Al ser ligeramente de diferentes dimensiones las imágenes, se las cortó utilizando ImageMagick⁸ para que todas sean iguales y así facilitar su manipulación.

Un problema que se tuvo fue que se contaba con 1313 imágenes de cartas que totalizaban 165,5MB. Esto presentaba 2 inconvenientes: por un lado, cada carta pesaba en promedio 129KB (lo cual es bastante si se desea mostrar muchas y de manera bien fluida) y el total de ellas hacía que la aplicación pesase mucho. Cabe destacar que el alojar las imágenes en un servicio externo no era una opción ya que el desarrollo de la aplicación siempre se tuvo en cuenta para que sea completamente offline. La solución fue reducir la calidad de las imágenes, pero a través de medios que garanticen gran reducción sin que el usuario pueda notar mucha diferencia. Primero, con ImageMacick se disminuyó la calidad de las imágenes un 25 % y con lo resultante se utilizó un algoritmo de compresión sin pérdida con Trimage⁹. El resultado fue pasar de un total de 165,5MB a 62,7MB, con un promedio que bajó de 129KB a 48,9KB por imagen y prácticamente con la misma calidad a la vista (ver ejemplo en la figura 7). El disminuir la calidad fue lo que produjo los mayores resultados (165,5MB a 93,4MB, baja de 44 %), pero la compresión resultó ser altamente eficaz igualmente (93,4MB a 62,7MB, baja de 33 %).

⁵<https://www.sqlite.org> [Consulta: 15 febrero 2016]

⁶<http://sqliteman.yarpen.cz> [Consulta: 14 febrero 2016]

⁷<https://launchpad.net/pyrenamer> [Consulta: 14 febrero 2016]

⁸<http://www.imagemagick.org> [Consulta: 14 febrero 2016]

⁹<https://trimage.org> [Consulta: 14 febrero 2016]



Figura 7: Imagen de 252KB (izq) pasada a 49,4KB (der)

6.1.3. Visualización de cartas

La parte más importante es mostrar la imagen de la carta. Dado un id, se busca en la aplicación la imagen correspondiente. Para detallar sus campos de información se realiza una consulta nuevamente utilizando el id. Ya que el id de la carta es el mismo que se utiliza en el del registro de las tablas, esta búsqueda es sumamente rápida. Una vez obtenidos los datos se los vuela la vista.

Se hace una lectura del texto de la carta en búsqueda de palabras clave de habilidades. La búsqueda se realiza de forma eficiente y a través del SpannableString que nos provee Android se crean los “links” de las habilidades con sus descripciones. De la misma forma también se modifican ciertos términos por sus imágenes respectivas (como las sendas de las cartas).

Se detalla aquellas ediciones en las que la carta se encuentra, y tocando en ellas se cambia la imagen y demás campos (rareza, artista) de la carta para mostrar la edición seleccionada. Para saber si una carta posee más de una edición se buscan sus campos por su id interno (el de 4 dígitos), obteniéndose un registro por cada edición de la carta.

En donde aplique, tocando la vista de imagen se cambia la misma por su reverso. Dado el id de una carta se deduce el nombre de la imagen del reverso.

6.1.4. Compartir imagen

Se creó un Navigation Drawer (panel lateral deslizable) para acceder a la opción de compartir imagen. Esta funcionalidad es posible crearla usando intents implícitos. En Android un intent es una descripción de una operación a ser realizada. Puede ser explícito, en donde se hace referencia a la clase que debe ser ejecutada o puede ser implícito, en donde en cambio se provee suficiente información en el intent para que el sistema en tiempo de ejecución determine qué ejecutar. En el caso de compartir imagen, en el intent se especifica que la acción a realizar es la de enviar datos y de tipo imagen jpg. Con esta información sumado a la ubicación del recurso (la imagen a enviar) con Android es todo lo que se necesita para poder compartir la imagen.

6.2. Iteración 2: Buscadores de cartas

Para realizar las búsquedas se creó un Builder con estado conversacional de la consulta. Se usa entonces un objeto de la clase CardsDB que va construyendo la query en base a los métodos que recibe y con ella finalmente realiza la consulta al enviarle el mensaje correspondiente. Por cada filtro que se desea agregar a la búsqueda se le envía un mensaje.

Hay algunas medidas que se tomaron para hacer las búsquedas más sencillas y eficientes.

Se puede realizar una búsqueda por alguna habilidad con palabra clave. Para esto se aprovechó la codificación de las habilidades explicada en el punto 6.1.3.

Es posible consultar por un texto en particular, ya sea la habilidad de la carta (lo que está escrito en la caja de texto) o leyenda como el nombre de la misma o del artista. Buscar textos presenta una dificultad adicional, y sobre todo si se trata del idioma español. Se debe tener en cuenta las diferencias entre mayúsculas y minúsculas y especial cuidado con los tildes. Es decir, escribiendo el término “solo” uno esperaría obtener resultados que contengan por ejemplo “Sólo”. Si simplemente eliminamos los acentos del texto a buscar, también deberíamos eliminar los acentos de los datos de la base de datos para poder realizar la búsqueda, lo cual sería muy ineficiente (más aún cuando varios de estos campos en los buscadores poseen auto-completado y se realizan múltiples consultas a la base). Para lidiar con este problema por cada campo por el que se busca por texto en las tablas se creó un campo adicional con el texto “crudo”, es decir sin acentos y todo en letras minúsculas. De esta manera se eliminan los acentos del texto a buscar, se pasa a minúsculas y la consulta se realiza sobre este campo crudo. Así es como se elimina el problema de los acentos y aparte se puede hacer una búsqueda case-sensitive en la base de datos.

Una cuestión similar a la anterior pasa con los valores “numéricos” de los campos coste, fuerza y resistencia, y es que no son siempre exactamente numéricos sino que pueden contener el valor X (ver figura 8). Esto presentó un problema ya que habría que guardar los valores “numéricos” como texto y tratarlos de esa manera, lo que dificultaría tareas como ordenarlos o contarlos. Esto también se solucionó con la misma idea de tener un campo crudo para realizar



Figura 8: Cartas con valores X

consultas que tenga solamente el valor numérico sin tomar en cuenta si posee o no X.

Se crearon 3 tipos diferentes de búsquedas según la necesidad del usuario.

6.2.1. Buscador básico

En el campo de texto se utilizó AutoCompleteTextView para proveer sugerencias de auto-completado. Se realizan queries por nombre de carta a la base de datos mientras se está escribiendo y se muestran los posibles resultados. No se mejoró la performance de esto ya que funciona de manera muy fluida. De haberse querido mejorar se hubiera guardado dentro de la aplicación la lista de los nombres de las cartas, y se buscaría en ella en vez de estar accediendo a la base de datos por cada letra que ingresa el usuario.

Los “botones” de las sendas son CheckBoxes que cambian su imagen según su estado (tildado o no).

6.2.2. Buscador extendido

La mayor dificultad para este buscador fue crear Spinners con opciones múltiples. Cuando se trabaja con componentes personalizados en muchos casos hay que saber manejar su ciclo de vida correctamente. En este caso en particular queremos que se guarde su estado al ser reconstruido (por ejemplo ante el giro de la pantalla). Para esto se creó un objeto que hereda de BaseSavedState que fue utilizado en el onSaveInstanceState y el onRestoreInstanceState del Spinner, es decir en aquellos métodos que se llaman al guardarse y recuperarse el estado de un View.

En este buscador y el avanzado se agrega la posibilidad de ordenar las cartas y elegir el tipo de listado. La elección de este último se persiste para que quede

como opción por defecto. Es decir, cada vez que se selecciona una opción se guarda en SharedPreferences, que es un map que cada aplicación de Android tiene pensado para ser utilizado como opciones de configuración del usuario.

6.2.3. Buscador avanzado

Lo interesante de este buscador es que posee una estructura sencilla con pocas opciones a la vista, y las mismas van apareciendo y desapareciendo según los elementos seleccionados. En la parte inferior de la pantalla se van detallando los filtros que se van agregando y también se tuvo que tener cuidado de mantener su estado. Para hacer esto último se creó un layout que guarde filtros y mantenga su estado con un BaseSavedState, utilizándolo de la misma manera que con los Spinners con opciones múltiples. Luego desde la vista cada vez que se restaura se le solicitan los filtros al layout y de esta manera se agregan los elementos a la lista. Esta lista básicamente detalla aquellos filtros que han sido agregados, siendo posible la eliminación de alguno a elección.

6.3. Iteración 3: Listas de cartas y validación de mazos

El trabajo de la iteración se dividió en listar cartas y en crear listas.

6.3.1. Listado de cartas

Se comenzó realizando una GridView (lista en grilla) mostrando las imágenes de las cartas. La manipulación de imágenes en Android no es trivial y más aún cuando se desea mostrar tantas juntas. Las tareas que son pesadas como carga de imágenes se hacen en un thread aparte para no limitar la capacidad del thread principal que es el que muestra la vista. Así es como por ejemplo en un comienzo se cargó la imagen de fondo en los buscadores. Pero en este caso se trata de muchas imágenes que el usuario querrá ver de manera fluida. Para conseguir esto se decidió usar una librería externa, se probó Picasso¹⁰ y Universal Image Loader¹¹. Se utilizó la segunda por su mayor capacidad de configuración y performance que demostró en la aplicación. Esta librería junto a la reducción del peso de las imágenes resultó en un scroll fluido por las imágenes.

Luego se implementó la vista de cartas con un ListView, en donde se muestra el dibujo de la carta y sus datos más relevantes. Nuevamente se utilizó la librería de carga de imágenes. Para obtener el arte de una carta se realiza un corte de la imagen.

6.3.2. Creación y validación de listas

Ante cada cambio en una lista se la pasa por un algoritmo para validarla y decir si es un mazo y en qué formatos se puede utilizar. Se realiza por descarte, comenzando con todos los formatos y de ellos viendo carta por carta a cuáles

¹⁰<http://square.github.io/picasso> [Consulta: 14 febrero 2016]

¹¹<https://github.com/nostra13/Android-Universal-Image-Loader> [Consulta: 14 febrero 2016]

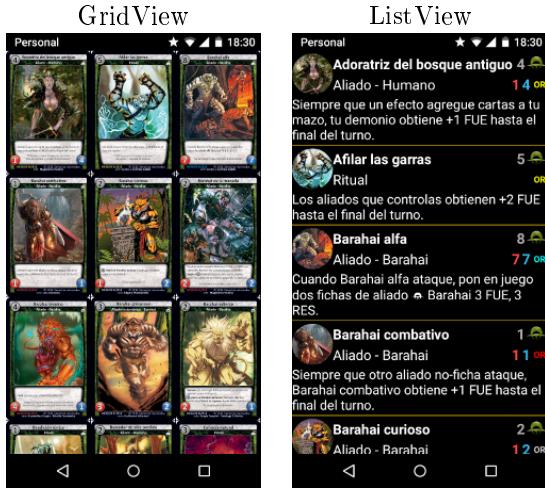


Figura 9: Listado de cartas

pertenece, eliminando los que no hasta terminar la lista o quedarse sin formatos para seguir revisando (es decir, en donde la lista no presenta ningún formato válido).

Se utilizó Sugar ORM¹² para guardar y acceder a las listas creadas. Se decidió utilizar esta librería por su facilidad de uso. Para persistir los objetos de una clase simplemente se puede extender de SugarRecord y esto automáticamente generará una tabla para la clase con un campo por cada atributo de instancia. Los objetos de la clase heredan los métodos save, delete, getId y setId para manejarlos en la base de datos y la clase cuenta con métodos estáticos para realizar consultas dado una cláusula where o con métodos prácticos como findById o listAll, que devuelve todas las instancias persistidas de la clase.

Un problema de Sugar ORM es que no puede persistir relaciones uno a muchos, y un mazo contiene cartas. Para resolver esto se decidió no persistir el atributo "cartas" (agregándole la annotation Ignore) y en cambio persistir un nuevo atributo de tipo String que sea el JSON del primero. Entonces se sobre-escribió el método save para que antes de guardar asigne en el nuevo atributo el valor JSON del atributo cartas. Para transformar un objeto (en este caso una colección) fácilmente a JSON se utilizó la librería Google Gson¹³. También se modificó el getter del atributo "cartas" para que, si es null, antes de devolver el valor se asigne al atributo la colección obtenida desde el JSON. Esto último también se realiza con Gson.

¹²<http://square.github.io/picasso> [Consulta: 14 febrero 2016]

¹³<https://github.com/google/gson> [Consulta: 14 febrero 2016]

6.4. Iteración 4: Puesta en producción e informe

Esta iteración incluyó aquellas tareas que se realizaron para poner a la aplicación en producción.

Primero se decidió agregar banners publicitarios en la parte inferior de la pantalla utilizando AdMob¹⁴, el cual está perfectamente integrado con Android. Para esto hay que registrarse en el servicio y una vez hecho esto se solicitan códigos para poder mostrar las publicidades. Simplemente en la vista se debe incluir al banner y desde la implementación añadirle un código y cargarlo. Google se encarga de, según datos como el lenguaje del dispositivo y el país en el que se encuentra, cargar publicidad pertinente en el banner que hemos incluido.

Finalmente se subió la aplicación a Google Play Store. Para ello hubo que registrarse como desarrollador ante Google (pagando la cuota única de 25 dólares). Un problema interesante que se tuvo con respecto a esto fue al realizar la descripción del programa, asignarle un icono y subirlo. Al tener la aplicación lista, fue suspendida y se envió una advertencia al desarrollador. La única razón que se brindó por email fue “violación de cláusula de Robo de identidad o comportamiento engañoso” sin hacer referencia a qué parte de esta cláusula se había violado. Se dedujo que habría sido por incluir como logotipo de la aplicación el mismo que el del juego. Recibir una advertencia no es algo menor, ya que persisten en el tiempo y al acumular 3 se da de baja al usuario desarrollador, y con él se pierde el control sobre las aplicaciones que se hayan publicado. Es decir, no se pueden actualizar más, ya no se podrán recibir ingresos por ellas y los usuarios no se enterarán de esto y seguirán utilizándola. Sin embargo, es posible apelar por única vez a una suspensión de aplicación y eso es lo que se hizo, mostrando pruebas de que se tienen los derechos para utilizar la imagen del juego. Una vez concedida la apelación se canceló la advertencia que se había recibido.

7. Trabajo futuro

El desarrollo de la aplicación deja mucho lugar para trabajo futuro. Se puede agregar la posibilidad de comprar una carta o lista de cartas, y que el usuario reciba el producto en su domicilio o en la tienda más cercana. De esta manera, la aplicación no sólo le servirá al jugador para planificar su juego, sino también para conseguir aquellas cartas que necesite. Si la carta que busca es de una edición antigua, se podrá detallar aquellas tiendas que la poseen, teniendo en cuenta que para esto se deberá desarrollar alguna interfaz en donde las tiendas detallen sus stocks y éstas deberían mantenerlo actualizado o sino responder on-demand por stock.

También se puede agregar el compartir listas o mazos, por ejemplo exportándolo a un formato de texto que se pueda imprimir o importar para verse dentro de la aplicación.

¹⁴<https://developers.google.com/admob> [Consulta: 14 febrero 2016]

Otro gran trabajo futuro es la posibilidad de poder jugar con la aplicación. Un jugador puede crear un mazo y jugar contra otro, por ejemplo con el propósito de poner a prueba sus estrategias. A esto luego se le podría agregar la posibilidad de jugar contra el programa, desarrollando su inteligencia artificial. El poder jugar puede desencadenar también en la creación de formatos o ediciones sólo disponibles de manera digital.

Ya que una aplicación es accesible desde cualquier parte del mundo, fácilmente INFERO può extenderse hacia los demás países y también se puede trabajar en la localización del programa y las cartas para poder disfrutarse del juego en otros idiomas.

8. Conclusiones

El desarrollo tuvo como finalidad ser una demostración, mostrando las posibilidades de desarrollo sin descuidar la usabilidad y la interfaz. Es por esto que se empleó mucho esfuerzo y dedicación en crear un software que sea amigable y eficiente para usar y que genere un alto atractivo visual. Aún así, se pudo crear una aplicación completa que termina siendo de gran utilidad para el propósito que se tuvo en mente, que es el de crear mazos para el juego INFERO. A su vez deja las puertas abiertas a continuar con su desarrollo, y en un futuro por ejemplo poder jugar a través de ella.

Los principales inconvenientes se encontraron en la carga eficiente de imágenes, persistencia de datos y en mantener el estado de las vistas personalizadas. En los 2 primeros casos la solución vino, al menos en buena parte, en utilizar librerías externas. Si bien en un comienzo hay que decidir cuál utilizar, su aprendizaje nos puede llevar un tiempo y nos podemos encontrar con algún bug, una vez superados estos pasos la recompensa de utilizar una librería es muy grande, sobretodo si se trata de realizar algo que seguro vamos a tener que volver a hacer en el futuro.

Con el objetivo en mente de ser una demostración se trabajó mucho en los requisitos no funcionales, que básicamente involucraron eficiencia, usabilidad y estética de la aplicación. Si bien éstos no están muy ligados al hecho de programar, la habilidad y el conocimiento necesario para poder satisfacerlos hacen a un mejor programador y lo pueden diferenciar del resto. Después de todo, crear una aplicación no es sólo programar, sino transmitir una idea.

Referencias

- [1] McWherter, J. and Gowell, S. Professional Mobile Application Development. Indianapolis, Ind: John Wiley & Sons, Inc, 2012. Print. ISBN: 978-1118240687
 - [2] Lee, and Chaim Krause. Beginning Android 4 application development. Indianapolis, Ind: Wrox/John Wiley & Sons, 2012. Print. ISBN: 978-1118199541
 - [3] Meier, Reto. Professional Android 4 application development. Indianapolis, IN: Wiley/Wrox, 2012. Print. ISBN: 978-1118102275
 - [4] <http://developer.android.com/intl/es/guide/topics/data/data-storage.html> [Consulta: 14 febrero 2016]
 - [5] <http://developer.android.com/intl/es/training/implementing-navigation/nav-drawer.html> [Consulta: 14 febrero 2016]
 - [6] https://www.sqlite.org/lang_createindex.html [Consulta: 14 febrero 2016]
 - [7] <https://www.sqlite.org/queryplanner.html> [Consulta: 14 febrero 2016]
- Repositorio de la aplicación: <https://github.com/presti/tip>
- Web de INFERO: <http://www.inferno.com.ar> [Consulta: 14 febrero 2016]



Figura 10: Campos de una carta

A. Campos de una carta

- Nombre Diferencia a una carta de otra.
- Senda Son cuatro y una Neutral. Todas las cartas de un mazo deben ser de la misma senda que el Demonio, y aquellas neutrales pueden incluirse en cualquier mazo.
- Tipo Definen las reglas por las que se rige una carta y la forma en la que puede ser utilizada. Se divide en tipo básico, supertipo y subtipo.
- Tipo básico Las cartas están principalmente definidas por su tipo básico. Es él quien determina el cómo debe jugarse la carta y establece las reglamentaciones más fundamentales. Todas las cartas pertenecen a al menos un tipo básico, siendo en casi todos los casos a tan sólo uno.
- Supertipo Establece reglas adicionales con respecto al uso de la carta.
- Subtipo Palabra clave que define alguna cualidad de la carta pero que en sí misma no se encuentra ligada a ninguna regla, sino que algunas cartas tienen habilidades que interactúan con otras que tengan un subtipo en particular. Es posible tener más de un subtipo.
- Texto Detalla las habilidades de una carta, que son características que le permiten afectar al juego.

| | |
|--------------------------|--|
| Edición | Conjunto de cartas creadas o reimpresas. Se venden por separado a través de mazos pre-armados o sobres y se imprimen solamente las cartas pertenecientes a la última edición, pasando a ser las anteriores más difíciles de conseguir. |
| Rareza | Define la cantidad que serán impresas de la carta. A mayor rareza, menor cantidad y por ende más difícil de conseguir. |
| Coste/Fuerza/Resistencia | Valores numéricos que influyen en el juego. Éste puede ser variable y especificarse con una X en la carta. |
| Ilustración | Representación gráfica de la carta. Su inclusión es puramente por atractivo visual y no modifica al juego. |
| Artista | Nombre del artista de la ilustración. |
| Leyenda | Texto de ambientación sin funcionalidad en el juego. Le otorga un contexto a la carta. |

B. Prueba de casos de uso

Para instalar el programa, desde la aplicación Google Play Store del dispositivo buscar el término “*inferno peltashield*” y seleccionar la aplicación de nombre *Inferno* con desarrollador *PeltaShield* que posee el logotipo como se muestra en la imagen a la derecha. Al ingresar al detalle seleccionar *Instalar* y aceptar el diálogo que nos muestra. Una vez terminada la instalación abrir la aplicación llamada *Inferno* identificada con el logotipo.

En el anexo siguiente se presentan ejemplos concretos para probar la aplicación.

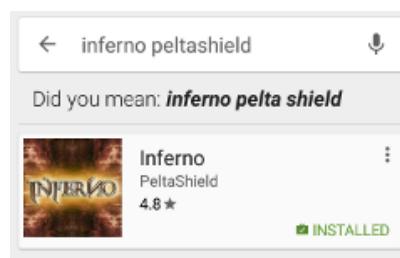


Figura 11: Inferno en Google Play Store

B.1. Búsqueda de cartas

Disponemos de tres instancias diferentes de buscadores que llamamos Básica, Extendida y Avanzada. Para acceder a la búsqueda básica, debemos dirigirnos al botón que se halla en el derecho de la barra superior de la pantalla al inicio de la aplicación. A la derecha del mismo, hallamos el ícono de búsqueda extendida, y correlativo a tal, encontramos el ícono de búsqueda avanzada.

Se buscará la carta de la derecha “*Dacal, regente de Trion*”.

Una vez seleccionados los filtros correspondientes, la búsqueda se realiza tocando el botón “Buscar”.

En el buscador avanzado el término “En el campo X seleccionar Y” significa “Seleccionar X del menú que se encuentra en el extremo izquierdo y luego elegir Y del menú que se encuentra más a la derecha en la misma fila”. Recordar que en este buscador los filtros se agregan con el botón “Aregar filtro” y, a menos que se indique lo contrario, el campo y/o no debe seleccionarse como “y” u “o”.

1. Poder buscar una carta según campos:



Figura 12: Carta a buscar

- Nombre: En la búsqueda básica, ingresar “*Dacal*” en el cuadro de texto que se halla en el extremo superior izquierdo. Con la búsqueda extendida, ingresar “*Dacal*” en el cuadro de texto *Nombre*. En la búsqueda avanzada, seleccionar el campo *Nombre* e ingresar “*Dacal*” en el cuadro de texto, dejando el menú en “contiene” o en “empieza con”. Al buscar aparece el detalle de la carta buscada.
- Texto: En la búsqueda básica, ingresar “*Reptil*” en el cuadro de texto que se halla en el extremo superior izquierdo. Con la búsqueda extendida, debemos ingresar “*Reptil*” en el cuadro de texto llamado *Texto*. En la búsqueda avanzada, seleccionar el campo *Texto* e ingresar “*Reptil*” en el cuadro de texto, dejando el menú en *contiene* o en *empieza con*. Al buscar aparece el detalle de la carta buscada.
- Senda: Se puede filtrar desde cualquiera de los 3 buscadores. En todos los casos, seleccionar el cuarto ícono -el de color verde- que se encuentra en la parte superior del buscador, cambiando su color de fondo de gris a anaranjado. Al realizar la búsqueda aparece el listado de cartas de la senda seleccionada (199 cartas en el buscador básico y extendido, 203 en el avanzado) entre las que se encuentra la carta buscada.
- Tipo: En el buscador básico seleccionar *Aliado* en el menú *Tipo*. En el buscador extendido seleccionar *Aliado* en el menú *Tipo*, *Único* en *Supertipo* y *Humano* en *Subtipo*. En el buscador avanzado seleccionar *Aliado* en el campo *Tipo básico*, *Único* en *Supertipo* y *Humano* en *Subtipo*. Al realizar la búsqueda aparece el listado de cartas que poseen los tipos seleccionados (435 cartas en el buscador básico, 16 en el extendido y avanzado) entre las que se encuentra la carta buscada.
- Edición: Desde el buscador extendido o avanzado, en el campo *Edición* seleccionar *Destino*. Al realizar la búsqueda aparece el listado de cartas de la edición *Destino* (128 cartas en el buscador extendido y 132 en el avanzado) entre las que se encuentra la carta buscada.
- Rareza: Desde el buscador extendido o avanzado, en el campo *Rareza* seleccionar *Épica*. Al realizar la búsqueda aparece el listado de las 76 cartas de la rareza seleccionada entre las que se encuentra la carta buscada.
- Habilidades con palabra clave: Desde el buscador extendido o avanzado, en el campo *Habilidad* seleccionar *Vínculo*. Al realizar la búsqueda aparece el listado de las 41 cartas con *Vínculo* entre las que se encuentra la carta buscada.
- Coste: Desde el buscador extendido, en el campo junto a *Coste=* seleccionar *3*. En el buscador avanzado, en el campo *Coste* seleccionar *3*. Al realizar la búsqueda aparece el listado de las 224 cartas con coste 3 entre las que se encuentra la carta buscada.
- Fuerza: Desde el buscador extendido, en el campo junto a *Fue=* seleccionar *3*. En el buscador avanzado, en el campo *FUE* seleccionar

3. Al realizar la búsqueda aparece el listado de cartas con fuerza 3 (84 cartas en el buscador extendido y 86 en el avanzado) entre las que se encuentra la carta buscada.
 - Resistencia: Desde el buscador extendido, en el campo junto a *Res=* seleccionar 4. En el buscador avanzado, en el campo *RES* seleccionar 4. Al realizar la búsqueda aparece el listado de las 63 cartas con resistencia 4 entre las que se encuentra la carta buscada.
 - Artista: En el buscador avanzado seleccionar el campo *Artista* e ingresar “zubeldia” en el campo de texto, dejando el menú en *contiene*. Al realizar la búsqueda aparece el listado de las 22 cartas con el término “zubeldia” como artista entre las que se encuentra la carta buscada.
 - Formato: En cualquier buscador en el campo *Formato* seleccionar *CdU*. Al realizar la búsqueda aparece el listado de cartas de formato *CdU* (436 cartas en el buscador básico y extendido, 448 en el avanzado) entre las que se encuentra la carta buscada.
2. Incluir o excluir las cartas de tipo ficha de la búsqueda: El buscador básico no dispone de esta opción, sino que por defecto siempre excluye las cartas de tipo ficha de la búsqueda. En el buscador extendido existe un CheckBox que dice “Mostrar fichas”, dejarlo desactivado. En el caso del buscador avanzado, en el campo *Tipo Básico* seleccionar *Ficha* y en el campo *y/o/no* elegir *no*. Al realizar la búsqueda aparece el listado de las 1030 cartas que no son fichas.
 3. Listar el resultado de la búsqueda:
 - Con imágenes: Con el buscador básico no es posible ver este tipo de resultado. En los otros dos seleccionar *Grilla* en el menú desplegable que se encuentra a la derecha del texto “Ver.” y realizar una búsqueda. Aparecerá la grilla con las imágenes de las cartas.
 - Con descripción detallada: Con el buscador básico se accede siempre a este tipo de resultado al realizar una búsqueda. En los otros dos casos seleccionar antes *Lista* del menú desplegable que se encuentra a la derecha del texto “Ver.”. Al buscar aparecerá la lista de las cartas con su descripción.
 4. Aplicar criterios de orden a los resultados según los campos de las cartas: Para ordenar de manera descendente por fuerza, en los buscadores extendido y avanzado seleccionar *Tipo* del menú desplegable junto al texto “Orden.”. En el menú contiguo dejar la dirección de la flecha hacia arriba. Al realizar la búsqueda aparece el listado de las cartas ordenadas según su tipo.
 5. Ver el detalle de una carta buscada: Buscar por el término “macizo” como nombre. Se mostrará la carta “Barahai Macizo” con su imagen en tamaño

completo y en buena calidad, formatos CdU y Eterno en los que es válida y valores de todos sus campos y al final se describen las erratas, limitaciones y prohibiciones de la carta.

- Errata: Buscar la carta de nombre “*Elixir de Saviaesencia*”. Al final de la descripción de la carta se verá detallada la errata que posee.
 - Limitación: Buscar la carta de nombre “*Estatua viviente*”. Cerca del final de la descripción de la carta se verá detallado que está limitada en CdU.
 - Prohibición: Buscar la carta de nombre “*Hadouken, sede nacional 2013*”. Al final de la descripción de la carta se verá que está prohibida (“baneada”) en todos los formatos.
 - Descripción de habilidades con palabra clave: Buscar la carta de nombre “*Azzam, jefe del desierto*”. En su descripción tocar sobre *Furia* o *Vínculo* para acceder a sus descripciones.
 - Visualización de ambas partes de la carta donde aplique(2.9): En una búsqueda de cualquier tipo seleccionar *Demonio* en el campo *Tipo*. Luego de buscar elegir cualquier resultado. Tocar sobre la imagen para ver el reverso y anverso de la carta.
 - Ediciones en las que fue impresa (detallando la rareza que posee en cada una) y la visualización de las mismas: Realizar una búsqueda por nombre escribiendo “*realidad*”. Debajo de la imagen se detallan las ediciones en las que la carta fue impresa y el color de fuente de cada una denota su rareza. Para ver las diferentes ediciones tocar sobre los nombres de ellas, o tocar sobre la parte inferior de la imagen.
6. Compartir una carta en las redes sociales o aplicaciones de mensajería configuradas en el dispositivo: Buscar la carta de nombre “*Azzam, jefe del desierto*”. En la vista de detalle de carta seleccionar la opción *Compartir* del menú desplegable desde el extremo derecho de la pantalla. Luego elegir Gmail para enviarla por email.

B.2. Listas de cartas

Para ver las listas de cartas, desde un buscador acceder desde el menú que se encuentra en el extremo superior derecho de la pantalla a “*Listas de cartas*”.

1. Crear lista: En las listas de cartas, en el campo de texto superior escribir *Fuertes* y aceptar el texto ingresado o tocar sobre el botón “+”. La lista de nombre *Fuertes* aparece listada en pantalla.
2. Agregar carta a lista: Crear la lista *Fuertes*. Realizar una búsqueda, seleccionar una carta y al ir a la vista de detalle de carta seleccionar la opción “*Agregar a lista*” del menú desplegable desde el extremo derecho de la pantalla. Luego elegir la lista *Fuertes*. La carta es agregada a la lista y se muestra un mensaje notificándolo.

3. Ver detalle de cartas, con cantidad de cada una y total: Crear la lista *Fuertes*, agregarle una vez la carta de nombre “*Zustatek, guardián del bosque*” y dos veces la carta llamada “*Araña gigante*”. En las listas de cartas, al lado del nombre *Fuertes* se detalla entre paréntesis que posee 3 cartas. Al seleccionar la lista se accede a las 2 cartas diferentes que le hemos agregado (*Zustatek* y la *Araña*) junto a la cantidad respectiva de cada una. Al seleccionar “*Araña gigante*” se ve su detalle y en la parte superior derecha de la pantalla la cantidad de la misma en la lista (2).
4. Editar o eliminar cantidad de una carta: Crear la lista *Fuertes*, agregarle la carta de nombre “*Zustatek, guardián del bosque*” una vez y la carta llamada “*Araña gigante*” dos veces. En las listas de cartas, seleccionar *Fuertes* y luego “*Araña gigante*”. En la parte superior derecha de la pantalla aparece su cantidad. Tocar sobre ella y desde el diálogo que se abre tocar sobre “+” 2 veces para aumentar la cantidad a 4 y seleccionar *Ok*. La cantidad ahora se mostrará como 4. Para eliminar la carta, desde el diálogo elegir *Eliminar* o tocar sobre “-” hasta llevar la cantidad a 0 y seleccionar *Ok*. La carta es eliminada del listado y volveremos a la vista de las cartas de la lista.
5. Eliminar lista: Crear la lista *Fuertes*. En las listas de cartas, tocar sobre la “X” junto al nombre *Fuertes* y confirmar el diálogo. La lista es eliminada y desaparece de la pantalla.
6. Dada una lista, saber si cumple con las condiciones para ser mazo y en qué formatos lo es: Crear la lista *Mazo*. Ir al buscador básico, seleccionar *CdU* en Formato y la cuarta senda tocando sobre el ícono verde. Tocar Buscar. Seleccionar el tercer elemento (“*Zustatek, guardián del bosque*”) y añadir a la lista *Mazo*. Agregar de la misma manera las siguientes 13 cartas de la lista. Volver al buscador para ir a las listas de las cartas. Seleccionar la lista *Mazo*. Seleccionar y editar la cantidad de cada carta a 4, excepto “*Brújula de Estari*” que se debe editar a cantidad 2 y “*Zustatek, guardián del bosque*” que debe dejarse en 1 existencia. Al volver a las listas de cartas, junto al nombre *Mazo* se detalla que es válido en los formatos *CdU* y Eterno (*Et*).