**SEP TecNM**

**Tecnológico Nacional de México**

Instituto Tecnológico de Acapulco

**Desarrollo de una aplicación móvil que realice los cálculos de los productos en existencia física, en base a las presentaciones que tiene cada uno de ellos, para ayudar a la optimización de tiempos en el proceso de toma física de inventario de Cinépolis La Isla Acapulco.**

**Presenta:**

Abundis Morales Kevin Leonel

Gallardo Cabrera Jesús Erasmo

**Director de proyecto:**

**Ing. Hernández Bravo Juan Miguel**

Acapulco, Gro. Junio, 2020

**Índice**

[**RESUMEN 5**](#_Toc42176005)

[**ABSTRACT 6**](#_Toc42176006)

[**Índice de figuras 7**](#_Toc42176007)

[**Índice de tablas 8**](#_Toc42176008)

[**Capítulo 1: Introducción** 9](#_Toc42176009)

[1.1 Antecedentes del problema 10](#_Toc42176010)

[1.2 Planteamiento del problema 15](#_Toc42176011)

[1.3 Objetivo general 17](#_Toc42176012)

[1.4 Objetivos específicos 17](#_Toc42176013)

[1.5 Justificación 18](#_Toc42176014)

[1.6 Hipótesis 21](#_Toc42176015)

[**Capítulo 2: Estado del arte** 22](#_Toc42176016)

[2.1 Herramientas similares 22](#_Toc42176017)

[2.1.1 Real Inventory 22](#_Toc42176018)

[2.1.2 ABC Inventory 22](#_Toc42176019)

[2.1.3 PartKeepr 22](#_Toc42176020)

[2.1.4 Openbravo 23](#_Toc42176021)

[2.1.5 GLPI 23](#_Toc42176022)

[2.2 Definiciones 23](#_Toc42176023)

[2.2.1 Definiciones de almacén 23](#_Toc42176024)

[2.2.2 Tipos de almacenes 24](#_Toc42176025)

[2.2.3 Definiciones de inventario 24](#_Toc42176026)

[2.2.4 Definición de inventario físico de almacén 25](#_Toc42176027)

[2.2.5 Importancia de los inventarios 25](#_Toc42176028)

[2.2.6 Funciones del inventario 26](#_Toc42176029)

[2.2.7 Tipos de inventario 27](#_Toc42176030)

[2.2.8 Concepto de toma física de inventario 28](#_Toc42176031)

[2.2.9 Finalidad de la toma física de inventario 28](#_Toc42176032)

[2.2.10 Objetivos de la toma física de inventario 29](#_Toc42176033)

[2.2.11 Beneficios de realizar tomas físicas de inventarios de manera periódica 29](#_Toc42176034)

[2.3 Dispositivos móviles 30](#_Toc42176035)

[2.3.1 Clasificación de dispositivos móviles 30](#_Toc42176036)

[2.3.2 Sistema operativo Android 31](#_Toc42176037)

[2.4 Aplicación móvil 31](#_Toc42176038)

[2.5 Ley de Pareto 31](#_Toc42176039)

[2.6 Análisis ABC 32](#_Toc42176040)

[2.6.1 Control de inventarios ABC 33](#_Toc42176041)

[2.6.2 Métodos de valoración o valuación de inventarios 34](#_Toc42176042)

[**Capítulo 3: Marco Teórico** 36](#_Toc42176043)

[3.1 Herramientas utilizadas 36](#_Toc42176044)

[3.1.1 Android Studio 36](#_Toc42176045)

[3.1.2 Firebase 38](#_Toc42176046)

[3.1.3 Enterprise Architect 41](#_Toc42176047)

[**Capítulo 4: Metodología** 42](#_Toc42176048)

[4.1 Modelado de Negocio 42](#_Toc42176049)

[4.2 Casos de Uso 62](#_Toc42176054)

[4.3 Diagrama de Clases 66](#_Toc42176055)

[4.4 Diseño de la Base de Datos. Diagrama de Entidad Asociación. 67](#_Toc42176056)

[4.5 Diccionario de Datos 70](#_Toc42176057)

[4.6 Diagrama de Despliegue 71](#_Toc42176058)

[**Capítulo 5: Implementación** 72](#_Toc42176059)

[5.1 Instalación de Android Studio 72](#_Toc42176060)

[5.1.1 Requisitos previos 72](#_Toc42176061)

[5.1.2 Descarga e instalación de Android Studio 73](#_Toc42176062)

[5.1.3 Creación de un proyecto en Android Studio 75](#_Toc42176063)

[5.2 Integración de Firebase con un proyecto de Android Studio 76](#_Toc42176064)

[5.2.1 Requisitos previos 76](#_Toc42176065)

[5.2.2 Opción 1: Usa el flujo de trabajo de configuración de Firebase console (recomendada) 77](#_Toc42176066)

[5.2.3 Opción 2: Usa Firebase Assistant de Android Studio (se requiere configuración adicional) 80](#_Toc42176067)

[5.2.4 Bibliotecas disponibles 81](#_Toc42176068)

[5.3 Contenido 83](#_Toc42176069)

[5.3.1 Inicio de sesión 83](#_Toc42176070)

[5.3.2 Registro de usuario 85](#_Toc42176071)

[5.3.3 Recuperación de contraseña 87](#_Toc42176072)

[5.3.4 Menú principal 89](#_Toc42176073)

[5.3.5 Menú almacén general 91](#_Toc42176074)

[5.3.6 Menú dulcería 93](#_Toc42176075)

[5.3.7 Menú taquilla 113](#_Toc42176076)

[5.3.8 Presentación de productos 119](#_Toc42176077)

[5.3.9 Acerca de nosotros 121](#_Toc42176078)

[**Conclusión** 123](#_Toc42176079)

[**Bibliografía** 124](#_Toc42176080)

[**Glosario de términos** 126](#_Toc42176081)

# **RESUMEN**

El desarrollo del proyecto planteado en este documento tiene como propósito el crear una aplicación móvil que sirva como apoyo para el conteo físico del inventario de Cinépolis La Isla Acapulco, dado que el proceso actual de toma de inventario es un proceso largo que, en la gran mayoría de las ocasiones, exige a el personal a cargo de esta tarea el quedarse más tiempo que el que dicta su horario laboral para poder terminar correctamente este proceso, esta situación es un potencial problema para la empresa ya que este acto puede darse a malos entendidos y, en el peor de los casos, puede terminar en una demanda por parte del trabajador.

Por esta razón, presentamos esta aplicación móvil que agilizará el proceso de toma física de inventario ahorrando los tiempos de realización del mismo, quitando el proceso que se hace a mano y objeto por objeto, a un proceso digital que tendrá los productos en una base de datos y que solo se seleccionarán las cantidades de cada producto, es decir, si se quiere decir que hay 5 bolsas de takis en presentación de 190gr, en vez de escribir todo lo anteriormente mencionado, únicamente se seleccionará la presentación de takis pertinente y se pondrá la cantidad en existencia de la misma, haciendo que este proceso de manera digital sea más rápido, ágil y certero con respecto a anotarlo con papel y pluma.

Para lograr tal objetivo, nos apoyaremos de herramientas de desarrollo tales como Android Studio, MySQL, PHP, entre otras para el desarrollo de la aplicación en dispositivos que cuenten con el sistema operativo Android, ya que este sistema es el más utilizado por los encargados del proceso de toma física de inventario de Cinépolis La Isla Acapulco.

**Palabras clave:** Conteo Físico, Toma Física de Inventario, Aplicación Móvil, Android Studio y Agilizar Tiempos

# **ABSTRACT**

The purpose of developing the project outlined in this document is to create a mobile application that serves as a support for the physical count of the inventory of Cinépolis La Isla Acapulco, since the current process of taking inventory is a long process that, in the great Most of the times, it requires the personnel in charge of this task to stay longer than the one dictated by their working hours in order to correctly finish this process, this situation is a potential problem for the company since this act can be misunderstood And, in the worst case, it can end in a lawsuit by the worker.

For this reason, we present this mobile application that will streamline the process of taking physical inventory, saving the time it takes, removing the process that is done by hand and object by object, from a digital process that will have the products on a data and that only the quantities of each product will be selected, that is, if it means that there are 5 bags of takis in a presentation of 190gr, instead of writing all of the above, only the relevant presentation of takis will be selected and it will be the amount in stock of it, making this process digitally faster, more agile and accurate with respect to writing it down with paper and pen.

To achieve this objective, we will use development tools such as Android Studio, MySQL, PHP, among others, to develop the application on devices that have the Android operating system, since this system is the most used by those in charge of the process of Physical inventory of Cinépolis La Isla Acapulco.

**Key words:** Physical Count, physical count of the inventory, mobile application, Android Studio and speed up times.

# **Índice de figuras**

[Ilustración 1: Real Inventory 22](file:///C:\Users\Abund\OneDrive\Escritorio\8%20Semestre\15-16%20Taller%20de%20Investigación%20II\Proyecto%20Aplicación%20Móvil%20Cinépolis\Documentación%20proyecto\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc39793136)

[Ilustración 2: PartKeepr 22](file:///C:\Users\Abund\OneDrive\Escritorio\8%20Semestre\15-16%20Taller%20de%20Investigación%20II\Proyecto%20Aplicación%20Móvil%20Cinépolis\Documentación%20proyecto\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc39793137)

[Ilustración 3: Openbravo 23](file:///C:\Users\Abund\OneDrive\Escritorio\8%20Semestre\15-16%20Taller%20de%20Investigación%20II\Proyecto%20Aplicación%20Móvil%20Cinépolis\Documentación%20proyecto\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc39793138)

[Ilustración 4: GLPI 23](file:///C:\Users\Abund\OneDrive\Escritorio\8%20Semestre\15-16%20Taller%20de%20Investigación%20II\Proyecto%20Aplicación%20Móvil%20Cinépolis\Documentación%20proyecto\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc39793139)

[Ilustración 5: Logo Android Studio 36](#_Toc39793140)

[Ilustración 6: Logo Firebase 38](#_Toc39793141)

[Ilustración 7: Opción de Auth (Authentication) en Firebase 39](file:///C:\Users\Abund\OneDrive\Escritorio\8%20Semestre\15-16%20Taller%20de%20Investigación%20II\Proyecto%20Aplicación%20Móvil%20Cinépolis\Documentación%20proyecto\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc39793142)

[Ilustración 8: Opción de Realtime Database en Firebase 40](file:///C:\Users\Abund\OneDrive\Escritorio\8%20Semestre\15-16%20Taller%20de%20Investigación%20II\Proyecto%20Aplicación%20Móvil%20Cinépolis\Documentación%20proyecto\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc39793143)

[Ilustración 9: Enterprise Architect 41](#_Toc39793144)

# **Índice de tablas**

[Tabla 1: Requisitos Android Studio Versión 3.x 37](file:///C:\Users\Erasmo%20Gallardo\Downloads\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc36504116)

[Tabla 2: Requisitos Android Studio Versión 2.x 37](file:///C:\Users\Erasmo%20Gallardo\Downloads\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc36504117)

[Tabla 3: Requisitos Android Studio Versión 1.x 37](file:///C:\Users\Erasmo%20Gallardo\Downloads\Proyecto%20Aplicación%20Móvil%20Conteo%20Físico%20Cinépolis.docx#_Toc36504118)

# **Capítulo 1: Introducción**

El presente proyecto tiene como propósito optimizar los tiempos en el proceso de toma física de inventario de Cinépolis La Isla Acapulco, debido a que este mismo es muy tedioso de realizar y lleva mucho tiempo realizarlo, razón por la cual los encargados al llevarlo a cabo, salen después de su jornada laboral, situación que le debe de importar a la empresa, porque puede provocar que haya demandas en su contra por esta situación, debido a la carga de trabajo que les expone a sus empleados en esta área.

El proceso de toma física de inventario es muy tardado y tedioso, porque deben de contabilizar todos los productos que hay en existencia física, esto significa que entre más productos haya en los almacenes, más tiempo tomará realizar el conteo físico; el conteo lo realizan manualmente usando una calculadora y su resultado lo anotan en bitácoras, las cuales ya debieron de imprimir antes de empezar el conteo, en estas bitácoras los productos ya vienen clasificados por tipo de negocio, los cuales se explicarán durante el desarrollo de este proyecto; además cada producto tiene una cierta unidad de medida en la que se contabiliza, por lo cual los encargados de esta área ya deben de conocerla y también algunos productos están distribuidos en varios tipos de negocio, por lo cual hay que llevar a cabo esas operaciones.

En base a esto, se optó por desarrollar una aplicación móvil que realice estas operaciones automáticamente por ellos, por lo cual ya no tendrán que concentrarse en la unidad de medida de los productos o en la presentación que tienen cada uno de estos productos.

Por ejemplo, una caja de Sabritas de Takis Flamin contiene 25 bolsas de este producto y cada bolsa pesa 190 gramos, con lo que tenemos que multiplicar las 25 bolsas por 190 gramos que nos da como resultado 4,750 gramos, que es lo que habría de este producto por 1 sola caja y tomando en cuenta que sean 32 cajas, se multiplicaría 4,750 por las 32 cajas que hay, dando como resultado 152,000 gramos de Takis Flamin en existencia física, como vemos es algo tedioso, por lo que se busca que esto sea automatizado mediante nuestra aplicación móvil que desarrollaremos especialmente para ellos  y al finalizar de insertar todos los datos, les generé un reporte de las existencias físicas de los productos.

## **Antecedentes del problema**

Solo hay 3 personas encargadas de realizar este proceso en el transcurso de un día. Una de ellas se encarga en el turno de la mañana de cuadrar[[1]](#footnote-1) los tres almacenes principales que hay en Cinépolis La Isla Acapulco, que corresponden al área del almacén principal. Esta área contiene los productos estrella del cine[[2]](#footnote-2), el almacén con los productos de cocina y por último el almacén con los productos del negocio de Coffee Tree[[3]](#footnote-3).

En ellos no debe de haber ningún faltante de producto y en caso de haberlo, se debe de rectificar en los puntos de venta si sobra ese producto y en caso de sobrar producto en los almacenes, se debe rectificar si hace falta ese producto en los puntos de ventas. Para realizar esto se imprime el reporte general de todos los productos que hay en existencia en estos almacenes por sistema (ya hay un software especial para llevar a cabo esto).

Una vez impreso el reporte, la persona encargada procede a contar todo lo que hay en existencia física en los almacenes, lo que conlleva a que haga operaciones manualmente con una calculadora estándar o su celular (todos los encargados de esta área cuentan con celular), que lleva consigo mismo durante todo el proceso y este mismo lo valla anotando con una pluma o lápiz en el reporte que imprimió previamente. Esto implica que la persona encargada pierda tiempo llevando a cabo todas estas operaciones, ya sean sumas, multiplicaciones, divisiones o restas, en las cuales podría equivocarse al hacerlas, lo que provocaría que el encargado volviera a contar nuevamente el producto que estaba contando mientras ocurrió ese error humano.

Otra situación que afecta en este proceso, es que cada producto tiene su propia presentación, por ejemplo, en el reporte dice que hay 42,000 gramos de queso manchego en existencia en sistema, pero en el almacén físicamente hay 3 cajas de queso manchego, que contienen a su vez 4 barras de queso manchego, en la cual cada barra de queso manchego pesa 3,500 gramos; así que se multiplican las 3 cajas de manchego que hay en existencia física, por las 4 barras de queso manchego que tiene cada caja y a su vez se multiplica por los 3,500 gramos que pesa cada barra de manchego, dando así como resultado los 42,000 gramos de queso manchego que hay en existencia en sistema.

Como vemos esto es algo tedioso y se pierde mucho tiempo para realizarlo, porque además de llevar a cabo las operaciones correspondientes, los encargados deben también de aprenderse la presentación de cada uno de los productos que hay en los almacenes.

Por otro lado, las otras dos personas encargadas en el turno de la tarde, deben de realizar el conteo de la dulcería (es donde están los puntos de venta) y los demás negocios que hay en el cine, como lo son la venta de crepas y frappés[[4]](#footnote-4).

Esto lo realizan con el objetivo de obtener las diferencias que hubo en toda la semana correspondiente, ya sean faltantes y sobrantes de cada uno de los productos, porque en base a los faltantes que hubo en la semana, se hacen los vales[[5]](#footnote-5) de cobro a las personas que estuvieron en esa semana en los puntos de venta y los sobrantes se ingresan a sistema.

El conteo de estos negocios lo realizan con unas bitácoras especiales que realizaron ellos mismos, estas mismas las imprimen en papel, lo que lleva a un costo en hojas de papel en papelería (aquí también les dan un presupuesto de lo que pueden gastar en artículos de papelería, en los cuales en algunos casos por falta de presupuesto no pueden pedir algún otro artículo que les haga falta en papelería) y en ellas van anotando lo que hay en existencia física.

En esta parte los encargados del área también deben de saber las presentaciones de cada uno de los productos, pero algo extra en estas áreas es que se manejan porciones de algunos productos. Por ejemplo, para realizar una crepa de queso crema con manzana canela, las porciones correspondientes son: 50 gramos de queso crema y 70 gramos de manzana canela, por lo cual deben de saber el peso de las porciones de cada uno de los productos que tienen esta modalidad, ya que, en los refrigeradores podría haber 80 porciones de queso crema, con lo cual cada porción pesa 50 gramos, así que se multiplican las 80 porciones por 50 gramos y nos daría como resultado 4,000 gramos de queso crema.

Otro problema al realizar el conteo en la tarde, es que algunos productos vienen en recipientes, los cuales tienen un peso y ese peso se debe de restar. Por ejemplo, en el área de frappés, los polvos de frappé vienen en recipientes, este recipiente tiene producto dentro de él, pero el recipiente también tiene un peso en gramos por sí solo, este mismo se pesa en una báscula ya calibrada y se pone una etiqueta de su peso del recipiente (en este caso, ya todos los recipientes tienen su etiqueta correspondiente).

Supongamos que este recipiente contiene polvo de frappé de piña colada, el cual pesa 1560 gramos, pero el peso del recipiente por si solo es de 500 gramos, así que se restan de los 1560 gramos los 500 gramos del peso del recipiente y nos quedaría así, 1060 gramos de polvo de frappé de piña colada realmente, esta cantidad es la que se anota en la bitácora y además de eso, puede que haya una bolsa entera de este polvo de frappé de piña colada, el peso de esta bolsa es de 1600 gramos, igualmente se anota en la bitácora y se suman estas dos cantidades, dando así la cantidad de 2660 gramos de polvo de frappé de piña colada, que es lo que hay en existencia físicamente.

Como vemos esto también es algo tedioso de realizar y conlleva aún más tiempo en realizarlo, porque se tienen que pesar los productos que tienen esta modalidad, provocando que los encargados salgan después de su jornada laboral.

Al finalizar de realizar el conteo físico de todos los productos que hay en existencia física, los encargados que les tocó el turno de la tarde, deben de realizar las últimas operaciones correspondientes de los productos y en caso de haber una diferencia muy exhaustiva, se realiza una última rectificación de ese producto antes de ingresarlo en sistema, para revisar que en realidad hace falta todo ese producto, porque una vez dado de alta en sistema ya no se puede deshacer esa acción y también en caso contrario, lo que es que haya un sobrante muy exhaustivo de algún producto, se realiza este mismo proceso de rectificación.

**Falta de presupuesto en nómina para contratar u autorizar horas extra a otra persona para que apoye en este proceso**. No se puede sobrepasar este presupuesto, porque se asigna una meta que se debe de cumplir, esta meta la asigna el corporativo y si no se llegará a cumplir, se procede a una sanción, que afecta en fines de auditoria y afecta en la calificación que tendrá el cine al cierre del año, lo cual también puede llegar a provocar que despidan a las personas encargadas de cumplir dicho proceso.

**Temporadas altas**. Como lo son las vacaciones de semana santa, las vacaciones de verano y las vacaciones decembrinas, el proceso de toma física de inventario se vuelve aún más tardado y tedioso para los encargados de esta área, ya que se espera una alta afluencia de gente y debido a esta afluencia, se debe de realizar un pedido de productos en mayor proporción a los proveedores correspondientes, para satisfacer la demanda de sus clientes. Un pedido en mayor proporción, significa que habrá más cantidad de productos físicamente en los almacenes y a mayor cantidad de productos en existencia física, se necesitará más tiempo para contabilizarlos.

**Promociones por cierto lapso de tiempo.** El corporativo a través de un correo electrónico, informa sobre las promociones que habrá en ciertas semanas del año. Por ejemplo, actualmente hay una promoción de 2 frappés por 89 pesos, esta promoción se proporciona al comprar tus boletos en taquilla y se entrega un cupón, el cual se debe de redimir en los puntos de venta de dulcería, al entregar este cupón se te hace válida la promoción. Así que, para satisfacer la demanda de este alimento en promoción, se debe de realizar un pedido en mayor proporción a los proveedores correspondientes, lo que significa tener una mayor cantidad de los productos que implican la realización de este alimento en los almacenes, como lo son la leche, la base rich[[6]](#footnote-6) (es con la que se hace la crema batida), el jarabe de chocolate, el granillo y el polvo de los frappés, los cuales tomará más tiempo contabilizarlos.

**Películas con alta afluencia de gente.** Las películas con alta afluencia de gente afectan también en este proceso, ya que, para satisfacer la demanda de los clientes, de igual manera se debe de realizar un pedido en mayor proporción a los proveedores correspondientes de los productos con mayor demanda, como lo son, las canastas de palomitas, los vasos de refresco, maíz, aceite para las palomitas, jarabes de refresco, sal para palomitas y los demás productos que se utilizan para la realización de las palomitas, que son cosas que nunca debe de hacer falta en los almacenes, porque son los productos principales que ofrece el cine a sus clientes, esto implica que haya mayor cantidad de productos en los almacenes físicamente y se tome más tiempo contabilizarlos.

## **1.2 Planteamiento del problema**

El proceso de la toma física de inventario en Cinépolis La Isla Acapulco es tedioso y lleva mucho tiempo llevarlo a cabo. Los encargados en turno de todas las áreas al realizar este proceso, regularmente salen mucho después de su jornada laboral, debido a que lleva demasiado tiempo realizar el conteo físico de todo lo que hay en existencia en los almacenes del cine.

Si el proceso de toma física de inventario no es realizado en su totalidad, los supervisores administrativos y subgerentes en turno, no pueden llevar a cabo el proceso de “cierre de día”, lo que es básicamente, depositar las ganancias del cine a la cuenta de banco que maneja el Corporativo ubicado en Michoacán y enviarles el reporte de este movimiento. Así que, si los encargados de almacén terminan demasiado tarde el proceso de toma física de inventario, los supervisores administrativos y subgerentes en turno, tienen que esperarlos, por lo que se ven en la necesidad, de ayudarlos en este proceso para que lo terminen en el menor tiempo posible y ellos puedan realizar el proceso de “cierre de día”.

Para poder llevar a cabo el conteo físico de todos los productos que hay en existencia en el cine, los encargados deben de aprenderse las presentaciones que tienen cada uno de ellos, porque si no se las saben, se pierde tiempo en estar averiguando las presentaciones que estos tienen. Por lo que, si eres nuevo en esta área, es el problema más grande que se tiene, ya que el no saber las presentaciones de los productos te quita tiempo el averiguar estas mismas y poder llevar a cabo el proceso de toma física de inventario.

El proceso de toma física de inventario se realiza los días jueves de cada semana, ya que es el día que hay menos afluencia de gente en el cine, pero en los casos de temporada alta, esta situación no es así, porque en temporada alta, sí hay mucha afluencia de gente o en películas que tienen un gran éxito, como lo fue la película de Avengers Endgame.

En estos casos, este proceso tiene una mayor presión, ya que los tiempos para realizarlo se disminuye. Por lo tanto, los encargados se ven en la necesidad de apurarse a realizar este proceso, porque en caso de no hacerlo, saldrán tarde del conjunto.

Otro proceso que se lleva a cabo es el conteo físico de dulcería, este proceso viene dentro del proceso de toma física de inventario (incluye conteo físico de taquilla, almacén y dulcería), pero a diferencia del proceso de toma física de inventario, donde se obtienen las diferencias de toda la semana; en el proceso de conteo físico de dulcería se obtienen las diferencias diarias que hubo en tal día de la semana de esta área, ya que aquí, se encuentran los puntos de venta de los productos del cine. En esta área, es donde siempre hay diferencias de productos, ya sea, por el mal marcado de los productos en punto de venta, que la persona de punto de venta entregue un producto equivocado, que les roben un producto, entre otros motivos más, por ello es que debe de realizarse un control diario de las diferencias que hay en esta área.

El fin de estos procesos, es el obtener las diferencias de los productos y en base a esas diferencias que hubo, determinar qué decisión tomarán los encargados. Por ejemplo, en el caso de los faltantes de productos, se procede a cobrar este mismo y en el caso de los sobrantes, se procede a aumentarlo en sistema. Durante estos procesos, se pueden realizar ajustes, si así lo deciden el encargado de almacén y el gerente, porque, por ejemplo, puede que falte un producto que cueste $31 pesos, pero sobra otro del mismo valor, así que en este caso puede realizarse un ajuste en el sistema durante este proceso.

También otra decisión que los encargados llegan a tomar, es en situaciones donde si hay mucho faltante de un producto en especial, significa que hay algo que está sucediendo, por ejemplo, si hay mucho faltante de jarabe de refresco, puede ser que la máquina de los jarabes está mal calibrada y este arrojando más jarabe del que debe; otra situación sería que el agua mineral no tenga la presión adecuada y se está arrojando una cantidad menor de agua mineral o que las personas en punto de venta no estén sirviendo de manera adecuada los refrescos; por lo que ellos, ante esta situación deben hacer un plan de acción o varios, para solucionar este problema.

Todo esto que describimos anteriormente, conlleva un desgaste físico y mental en ellos, que provoca un menor rendimiento en su trabajo que realizan al siguiente día, porque puede que les toque estar realizando el proceso de toma física de inventario en el turno de la tarde y al siguiente día estar en el turno de la mañana llevando a cabo los demás procesos que le corresponden, como pueden ser la conciliación de inventarios o la asignación de fondos[[7]](#footnote-7) en sistema para las personas que abrirán puntos de venta. Cuando pasa esta situación, los encargados optan por quedarse a dormir en el cine, por lo que no pueden descansar de una manera adecuada que conlleva un desgaste físico y el desgaste mental es causado por el estrés que conlleva realizar estos procesos.

## **1.3 Objetivo general**

* Optimizar el tiempo de realización del proceso de toma física de inventario de Cinépolis La Isla Acapulco.

## **1.4 Objetivos específicos**

* Conocer completamente el proceso de la toma física de inventario de Cinépolis La Isla Acapulco.
* Automatizar las operaciones que son realizadas en papel por encargados a través de una aplicación móvil.
* Disminuir el error humano al momento de realizar operaciones manualmente con la calculadora, lápiz y papel.
* Disminuir el uso de hojas de papel, ya que las bitácoras especiales que ellos tienen, ya no se imprimirán, solo se imprimirán los reportes.
* Poner un apartado donde vengan todas las presentaciones de los productos.
* Organizar los productos por las categorías que vienen en los reportes.
* Generar reportes de existencias físicas de productos de las áreas.
* Llevar a cabo capacitaciones correspondientes a los encargados del área de almacén, de cómo se utiliza y qué funciones realiza la aplicación móvil.

**Nota:** En el caso del almacén general y taquilla no debe de haber diferencias, siempre debe de encontrarse cuadrado y sí existen diferencias, deberán ser ajustadas.

## **1.5 Justificación**

**Justificación tecnológica** **y práctica**

La realización de este proyecto, tiene una importancia muy alta en el área de almacén de Cinépolis La Isla Acapulco, ya que, uno de los principales beneficios que tendrá esta aplicación móvil, será que se agilizará el proceso de toma física de inventario de este establecimiento, lo cual hará que dicho proceso se realice en un tiempo menor, debido a que las operaciones que hacen los encargados de dicha área, serán automatizadas a través de la aplicación móvil, en consecuencia, se espera que los encargados salgan a la hora que termina su jornada laboral o en un tiempo menor al que salían anteriormente sin dicha aplicación móvil, para que así, al siguiente día, procedan a realizar sus demás actividades correspondientes.

La utilidad de esta aplicación móvil será de un uso concurrente, ya que puede usarse para otro proceso que los encargados realizan diariamente, que es el conteo físico del área de dulcería (el conteo físico de almacén y taquilla se hace semanalmente), por lo cual, puede usarse el apartado que tendrá la aplicación móvil de esta área para realizar las operaciones correspondientes y proceder al final a sacar el reporte del conteo diario de tal fecha, para que el encargado vea las diferencias que se tuvieron por día de la semana y llevar un mejor control de dichas diferencias.

**Justificación social**

Los beneficiarios de este proyecto, son principalmente las personas encargadas del área de almacén de Cinépolis La Isla Acapulco, entre los que se encuentran:

* El encargado principal de almacén
* El auxiliar de almacén
* El stick[[8]](#footnote-8) de almacén

Ya que éstos mismos son los que le sacarán el mayor provecho a la aplicación móvil que crearemos especialmente para ellos y la realizaremos en base a sus necesidades, por lo que será personalizada para ellos, con lo que se espera, que tengan un mayor entendimiento de la aplicación móvil, debido a que esta misma, estará familiarizada con lo que ellos hacen regularmente en este proceso.

Otros beneficiarios de este proyecto, son los supervisores de dulcería, ya que estos, deben realizar el reporte de las diferencias diarias que hubo en esta área, reporte el cual debe ser entregado al subgerente de dulcería y al gerente del conjunto, porque en base a este mismo, ellos tendrán que tomar decisiones.

Si la aplicación móvil al implementarse tiene un gran impacto en el conjunto de Cinépolis La Isla Acapulco y se demuestra que con ella se agiliza el proceso de toma de inventario, podrá ser usada en los demás conjuntos que hay en la República Mexicana, beneficiando principalmente a todos los encargados del área de almacén y supervisores de dulcería.

**Justificación económica**

Los encargados del área de almacén, estarían dispuestos a pagar una cantidad de dinero por la aplicación móvil. La razón por la que decimos esto, es porque les hicimos una encuesta a ellos en el trabajo, donde viene incluida la pregunta de si estarían dispuestos de pagar un costo por la aplicación móvil y contestaron que sí, porque a ellos les facilita su trabajo en este proceso de toma física de inventario, el cual, es muy tardado y tedioso de realizar. La aplicación móvil será de gran ayuda para realizar las operaciones de este proceso y evitará que se aprendan las presentaciones de todos los productos, debido a que la aplicación tendrá un apartado especial para describir cada una de ellas (propuesta que me dieron ellos mismos) y también les ayudará en el proceso de conteo físico de dulcería que realizan diariamente.

Así que ellos al tener un mejor control sobre este proceso de toma física de inventario, que es un proceso crucial en esta área, pueden llegar con más facilidad a la meta que les pone el corporativo del presupuesto de costo de venta[[9]](#footnote-9) y al cumplir correctamente la meta mencionada, se les otorga un bono como gratificación.

Otra razón para realizar este proyecto, es que, si este problema lo tiene el conjunto de Cinépolis La Isla Acapulco, también lo tienen los demás conjuntos de la cadena de Cinépolis. Por lo tanto, hay un mercado grande, para vender esta aplicación móvil a los demás conjuntos, para facilitarles este proceso de toma física de inventario y realizarlo en menos tiempo.

En la siguiente información, se muestra el total de cines que operan alrededor del mundo, de la cadena de cines de Cinépolis:

“Cinépolis es una [empresa](https://es.wikipedia.org/wiki/Empresa) [mexicana](https://es.wikipedia.org/wiki/M%C3%A9xico) dedicada a la exhibición de películas fundada en [1971](https://es.wikipedia.org/wiki/1971) en [Morelia](https://es.wikipedia.org/wiki/Morelia), [Michoacán](https://es.wikipedia.org/wiki/Michoac%C3%A1n), bajo el nombre de Organización Ramírez. Hasta el 31 de enero del 2019 operan 939 cines con 5771 salas digitales; 434 se ubican en México y 505 en 14 países: [Estados Unidos](https://es.wikipedia.org/wiki/Estados_Unidos), [Guatemala](https://es.wikipedia.org/wiki/Guatemala), [Honduras](https://es.wikipedia.org/wiki/Honduras), [El Salvador](https://es.wikipedia.org/wiki/El_Salvador), [Costa Rica](https://es.wikipedia.org/wiki/Costa_Rica), [Panamá](https://es.wikipedia.org/wiki/Panam%C3%A1), [Colombia](https://es.wikipedia.org/wiki/Colombia), [Perú](https://es.wikipedia.org/wiki/Per%C3%BA), [Chile](https://es.wikipedia.org/wiki/Chile), [Argentina](https://es.wikipedia.org/wiki/Argentina), [Brasil](https://es.wikipedia.org/wiki/Brasil), [España](https://es.wikipedia.org/wiki/Espa%C3%B1a), [India](https://es.wikipedia.org/wiki/India), [Omán](https://es.wikipedia.org/wiki/Om%C3%A1n), [Baréin](https://es.wikipedia.org/wiki/Bar%C3%A9in), [Arabia Saudita](https://es.wikipedia.org/wiki/Arabia_Saudita).”

(Wikipedia, 2019)

**Justificación ambiental**

Gracias a la aplicación móvil, se disminuirá el uso de hojas de papel, porque las bitácoras especiales que ellos imprimen de apoyo para realizar el conteo de todas las áreas, ya no se imprimirán y solo se imprimirán los reportes, ya que estos son obligatorios para casos de auditoría y/o revisiones en el cine; y las bitácoras especiales, no son obligatorias en casos de revisiones o auditorias.

## **1.6 Hipótesis**

La automatización mediante la aplicación móvil de las operaciones realizadas en la toma física de inventario, provoca que disminuya en un 30% la realización de dicho proceso y reduce el 80% del uso de hojas de papel para este mismo. (Hipótesis Descriptiva).

# **Capítulo 2: Estado del arte**

## **2.1 Herramientas similares**

### **Resultado de imagen para real inventory app2.1.1 Real Inventory**

Ilustración 1: Real Inventory

Esta aplicación para iPhone y iPad, funciona sincronizándose con Cashier Live, un programa para computadora que gestiona inventarios. Con [Real Inventory](https://itunes.apple.com/us/app/retail-inventory/id440020325?mt=8) te será muy sencillo llevar a cabo el recuento de productos que tengas en tu almacén. La aplicación te muestra un control exhaustivo de tus existencias y aporta datos sobre cada producto como, por ejemplo, cuántas unidades hay del mismo, a qué hora lo registraste en el inventario, etc… La parte negativa de Real Inventory es que a pesar de que la app es completamente gratuita la cuenta de Cashier Live, a la que has de vincular Real Inventory, es de pago.

### **2.1.2 ABC Inventory**

Se trata de una **herramienta fácil para gestionar inventarios** en una empresa pequeña, ya que en su versión gratuita [ABC Inventory](http://almyta.com/abc_inventory_software.asp) solo puede ser instalado en un ordenador. Gracias a este programa, tendrás las herramientas básicas para controlar de forma sencilla las existencias de tu almacén, podrás hacer recuento de productos, dividirlos en diferentes categorías para llevar un mayor control.

### **Resultado de imagen para partkeepr2.1.3 PartKeepr**

Ilustración 2: PartKeepr

[PartKeepr](https://partkeepr.org/) es una herramienta especial para gestionar el inventario de empresas de electrónica y te ayudará a mantener los objetos de tu almacén siempre organizados. Aunque es cierto que no ofrece tantas funciones como las aplicaciones y programas mencionados anteriormente, te compensará utilizarla si tu empresa se dedica al **sector de la electrónica** ya que tiene ciertas características pensadas principalmente para este tipo de negocios.

### **2.1.4 Openbravo**

Con la aplicación [Openbravo](http://www.openbravo.com/), basada en la nube, podrás anotar todas las existencias de tu almacén y, además, controlar grandes salidas y entradas de productos. Para cada objeto de tu lista de inventario, puedes asignarle una ubicación, un proveedor y muchos atributos más.

Ilustración 3: Openbravo

### **2.1.5 GLPI**

Ilustración 4: GLPI

****Es una **aplicación dirigida a la gestión de inventario** de aparatos electrónicos. Con [GLPI](http://www.glpi-project.org/spip.php?article43) puedes tener constancia de cuántos equipos tienes en tu empresa y, además, acceder al historial con los últimos movimientos que se han producido en la empresa respecto a los dispositivos electrónicos.

## **2.2 Definiciones**

### **2.2.1 Definiciones de almacén**

Para (Hérnandez, 2007) “El almacén es un elemento importante en la cadena logística, pues si es bien gestionado puede ser capaz de estabilizar la producción con la demanda, ya que intenta sincronizar las distintas carencias entre la fabricación y la demanda, además supone un suministro permanente de materia prima”.

El almacén para (Sánchez, 2012) “es el recinto donde se realizan las funciones de recepción, manipulación, conservación, protección y posterior expedición de productos” (pág. 38). Así mismo, establece que los almacenes brindan las siguientes ventajas:

* Permiten una mejor organización en la distribución de las mercancías.
* Posibilitan una correcta conservación de los productos.
* Posibilitan una utilización racional de la técnica (con la concentración de los almacenes).

### **2.2.2 Tipos de almacenes**

Para (Hérnandez, 2007) “cada tipo de almacén tiene una serie de características diferentes, que sirven para poder clasificarlos y agruparlos atendiendo a una relación de criterios” como los siguientes:

**Según su función en la red logística:** Se pueden separar en almacén de consolidación y almacén de división de envíos o rupturas.

**Según su situación geográfica y la actividad que realicen:** Se pueden distinguir entre: Almacén central, almacén regional y almacén de tránsito.

**Según el tratamiento fiscal que reciben los productos almacenados:** Tenemos los siguientes tipos de almacenes: Almacén con productos en régimen fiscal general y las zonas francas.

**Según el recinto del almacén:** Estos almacenes son del siguiente tipo: Almacén abierto y almacén cubierto.

**Según el grado de mecanización:** Hay varios tipos: Almacén convencional y almacén mecanizado.

**Según la propiedad del almacén como inmueble:** Pueden ser: Almacén en propiedad y alquiler de espacio.

### **2.2.3 Definiciones de inventario**

Para (Hérnandez, 2007) “El inventario es una parte fundamental de todo sistema logístico, es una cantidad almacenada de materiales que se utilizan para facilitar la producción o para satisfacer la demanda del consumidor”.

Para (Guerrero, 2009) “los inventarios comprenden las materias primas, productos en proceso y productos terminados o mercancías para la venta, los materiales, repuestos y accesorios para ser consumidos en la producción de bienes fabricados para la venta o en la prestación de servicios, empaques, envases y los inventarios en tránsito”.

### **2.2.4 Definición de inventario físico de almacén**

“Es una forma de verificación física que consiste en constatar la existencia o presencia real de los suministros, insumos, repuestos y otros bienes o materiales almacenados, apreciar su estado de conservación y condiciones de seguridad” (EGASA, 2019, pág. 2).

### **2.2.5 Importancia de los inventarios**

El manejo de inventarios es uno de los principales problemas de la administración de empresas debido a que es un componente fundamental de la productividad. Si se mantienen inventarios demasiados altos, el costo podría llevar a una empresa a tener problemas de liquidez financiera, ya que un inventario “parado” inmoviliza recursos que podrían ser mejor utilizados en funciones más productivas de la organización. Además, el inventario “parado” tiende a tornarse obsoleto, a quedar fuera de uso y corre el riesgo de dañarse. Por otro lado, según (Fillet, 2003) “si se mantiene un nivel insuficiente de inventario, podría no atenderse a los clientes de forma satisfactoria, lo cual genera reducción de ganancias y pérdida de mercado”. Es vital para toda empresa, industria y comercio llevar inventarios sanos, ya que esto garantizará una mayor confiabilidad en el proceso diario de movimientos de sus productos dentro de los almacenes.

(LÓPEZ, 2015, pág. 22)

### **2.2.6 Funciones del inventario**

El inventario tiene según (Hérnandez, 2007) las funciones siguientes:

* Garantizar un determinado nivel de servicio al cliente.
* Ajustar las curvas de oferta y demanda dando cierta estabilidad a la primera ante una flexibilidad a la segunda.
* Evitar rupturas de inventario por la incertidumbre en el pronóstico de demanda, en los períodos de entrega o en la calidad y cantidad de productos recibidos.
* Protección ante situaciones imprevistas. El inventario permite asegurar la oferta en circunstancias inusuales que pueden conducir a un incremento de la demanda o a la disminución de la oferta o ambos.
* Protección contra los incrementos de precio.
* Hacer frente a posibles errores en la gestión de compras que puede llevar a un estado de desabastecimiento de insumos.
* Asegurar el flujo logístico garantizando el suministro en la producción o el consumo.

Hay dos decisiones básicas de inventario que los gerentes deben hacer cuando intentan llevar a cabo las funciones de inventario, estas decisiones se deben de tomar de forma individual para cada uno de los artículos en el inventario:

* Qué cantidad de un artículo hay que ordenar cuando el inventario de este mismo se va a reabastecer.
* Cuando reabastecer el inventario de este artículo.

### **2.2.7 Tipos de inventario**

1. **Inventario general:** Denominado también inventario masivo, incluye a todos los bienes almacenados en los diferentes ambientes o depósitos de la empresa y sirve de sustento a los Estados Financieros de la entidad. Este inventario requiere de un apropiado programa y se realiza por lo menos una vez en cada ejercicio. El inventario general normalmente se efectúa cerrando las operaciones del almacén por uno o varios días, pudiendo realizarse también en movimiento tomando las precauciones que se señalan en la presente norma.
2. **Inventario selectivo:** Comprende a un grupo de bienes previamente seleccionados. Se realiza periódicamente o cada vez que sea necesario comprobar que los registros de inventario se mantienen al día, así como para verificar que las existencias físicas concuerden con las tarjetas de control visible de almacén y los registros físico - valorados correspondientes. Se realiza sin paralizar las actividades de almacén pudiendo o no bloquear temporalmente la documentación y el despacho del grupo de bienes objeto del inventario por el plazo que dure el mismo.
3. **Inventario al barrer:** Modalidad que consiste en que el equipo de verificadores procede a efectuar el inventario empezando por un punto determinado del almacén efectuando el conteo de todos los bienes encontrados sin excepción.

(EGASA, 2019, pág. 2 y 3)

### **2.2.8 Concepto de toma física de inventario**

La toma física de inventario consiste en llevar a cabo un proceso ordenado de verificación física de los bienes, materia prima o producto terminado a una fecha determinada, con el fin de asegurar su existencia real. Los resultados obtenidos de los registros contables, son comparados con la constatación física, a fin de establecer su conformidad. Posteriormente las diferencias que pudieran existir son aclaradas e investigadas por las personas a cargo, y se procede a realizar los ajustes necesarios, según sea el caso. La toma del inventario deberá considerar, además de la constatación física, el estado de conservación, condiciones de utilización, almacenamiento y seguridad del bien o producto objeto de la misma. Este proceso se debe realizar de forma periódica, de acuerdo a las políticas internas de la empresa. Los inventarios se realizan en almacenes, centros de distribución y oficinas del cliente dependiendo el tipo de inventario a realizarse.

(Koneggui, 2019)

### **2.2.9 Finalidad de la toma física de inventario**

“Determinar la existencia física de los suministros, insumos, repuestos y otros bienes o materiales de propiedad de la Empresa que se encuentren almacenados en los diferentes Almacenes o Depósitos. Dicha verificación deberá ser contrastada con las existencias que figuran en los registros contables de la empresa a la fecha del inventario” (EGASA, 2019, pág. 1).

### **2.2.10 Objetivos de la toma física de inventario**

a) Planificar la realización del inventario físico en los almacenes de la Empresa.

b) Preparar el material necesario para la toma de inventario físico.

c) Organizar la toma de inventario físico de almacén de tal manera que el mismo se lleve a cabo utilizando criterios de racionalidad, eficiencia y economía.

(EGASA, 2019, pág. 1)

### **2.2.11 Beneficios de realizar tomas físicas de inventarios de manera periódica**

1. Mantener un inventario contable real
2. Retirar de Stock productos o bienes obsoletos
3. Dar de baja Activos fijos obsoletos
4. Comprobar codificación de los artículos y tener información real de los mismos
5. Recodificaciones y cambios de nomenclatura oportuna
6. Definir responsables o comprobar la efectividad de los encargados de cada uno de los bienes o productos objeto de la toma física.

(Koneggui, 2019)

## **2.3 Dispositivos móviles**

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales.

Una característica importante es el concepto de movilidad, los dispositivos móviles son pequeños para poder portarse y ser fácilmente empleados durante su transporte.

### **2.3.1 Clasificación de dispositivos móviles**

Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, era necesario hacer una clasificación de los mismos, por ello en el 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles.

**- Dispositivo Móvil de Datos Limitados (Limited Data Mobile Device):** teléfonos móviles clásicos. Se caracterizan por tener una pantalla pequeña de tipo texto. Ofrecen servicios de datos generalmente limitados a SMS y acceso WAP.

**- Dispositivo Móvil de Datos Básicos (Basic Data Mobile Device):** se caracterizan por tener una pantalla de mediano tamaño, menú o navegación basada en iconos, y ofrecer acceso a emails, lista de direcciones, SMS, y, en algunos casos, un navegador web básico. Un típico ejemplo de este tipo de dispositivos son los teléfonos inteligentes (“smartphones”).

**- Dispositivo Móvil de Datos Mejorados (Enhanced Data Mobile Device):** se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 pixeles), navegación de tipo stylus, y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos" (Basic Data Mobile Devices) más aplicaciones nativas como aplicaciones de Microsoft Office Mobile (Word, Excel, PowerPoint) y aplicaciones corporativas usuales, en versión móvil, como Sap, portales intranet, etc. Este tipo de dispositivos incluyen los S.O. como Windows Mobile.

### **2.3.2 Sistema operativo Android**

Android es un [sistema operativo móvil](https://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil) desarrollado por Google, basado en Kernel de Linux y otros software de código abierto. Fue diseñado para [dispositivos móviles](https://es.wikipedia.org/wiki/Dispositivo_m%C3%B3vil) con [pantalla táctil](https://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil), como [teléfonos inteligentes](https://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente), [tabletas](https://es.wikipedia.org/wiki/Tableta_(computadora)), [relojes inteligentes (Wear OS)](https://es.wikipedia.org/wiki/Wear_OS), [automóviles (Android Auto)](https://es.wikipedia.org/wiki/Android_Auto) y [televisores (Android TV)](https://es.wikipedia.org/wiki/Android_TV).

## **2.4 Aplicación móvil**

Una aplicación móvil es aquella desarrollada especialmente para ser ejecutada en dispositivos móviles como un teléfono celular, tabletas y similares.Estas aplicaciones tienen características especiales para poder funcionar en estos dispositivos móviles que, por lo general, tienen menos capacidad de procesamiento y almacenamiento que computadoras de escritorio o notebooks.En general estas aplicaciones son conocidas como "apps".

El desarrollo de aplicaciones móviles es un campo de trabajo relativamente reciente. Los desarrolladores de terceros han sido capaces de crear aplicaciones móviles hace menos de una década, por lo que no es de extrañar que las herramientas sigan evolucionando.

## **2.5 Ley de Pareto**

La **ley de Pareto**, también conocida como la **regla del 80/20**, establece que, de forma general y para un amplio número de fenómenos, aproximadamente **el 80% de las consecuencias proviene del 20% de las causas**.

El origen de este principio se encuentra en la observación empírica que realizó el **economista y sociólogo italiano Vilfredo Pareto** (1848-1923) respecto a la distribución de la propiedad en Italia. Así, concluyó que, de hecho, el 80% de la propiedad del país correspondía tan sólo al 20% de la población. Tras estudiar la distribución de la propiedad en otros países, descubrió que seguía el mismo patrón.

**En el mundo empresarial**, la ley de Pareto suele cumplirse, por ejemplo, en los campos de ventas y gastos. Así, en muchos casos se podrá comprobar cómo el 80% de las **ventas** de una empresa proviene de un 20% de sus clientes, o de un 20% de sus productos, o el 80% de sus **gastos**, del 20% de sus proveedores. También se puede aplicar en **logística** (controlando el 20% de los productos almacenados puede controlarse el 80% del valor de los productos del almacén), o en **ingeniería de software** (el 80% de los fallos de un software tiene su origen en un 20% del código de dicho software).

(queaprendemoshoy, 2013)

## **2.6 Análisis ABC**

El análisis ABC consiste en aplicar el principio de Pareto o [regla 80/20](http://www.atoxgrupo.com/website/noticias/regla-80-20-almacenes) para segmentar entidades. Típicamente se aplica en el ámbito del almacén para clasificar el inventario según su importancia. Aunque se pueden seguir diferentes criterios (según cada almacén y tipos de mercancía que manejen), un criterio típico es el valor de inventario de cada referencia, calculado como su demanda anual multiplicada por su coste unitario.

Después, se ordenan de mayor a menor y se agrupan según el porcentaje que representan respecto al total. Así, los porcentajes típicos resultantes serán:

* **Categoría A**: En torno al 20% de las referencias representan aproximadamente el 80% del valor del inventario (regla 80/20).
* **Categoría B**: En torno al 30% de las referencias representan aproximadamente el 15% del valor del inventario.
* **Categoría C**: En torno al 50% de las referencias representan sólo el 5% del valor del inventario.

(ATOX SISTEMAS DE ALMACENAJE, 2017)

### **2.6.1 Control de inventarios ABC**

Una vez realizado el análisis ABC sobre el inventario y habiendo quedado este clasificado en las tres categorías A, B y C, se toman decisiones de gestión de la cadena de suministro basándose en dichas categorías. Entonces, se establecerán estrategias de control de inventario diferentes en función de la importancia de las diferentes mercancías. Es por esto por lo que el análisis ABC aplicado al inventario se llama a veces control de inventario selectivo.

**Categoría A**

Los productos de la categoría A son los más importantes para el negocio, típicamente un 20% de las referencias representando el 80% del valor. Por tanto, deberán destinarse recursos para que su control de inventario sea más exhaustivo y con conteos cíclicos más frecuentes.

**Categoría B**

Los productos de la categoría B tienen una importancia moderada, intermedia entre la categoría A y la categoría C. Típicamente serán en torno al 30% de las referencias, constituyendo el 15% del valor. Es importante hacer un seguimiento de los productos B, puesto que algunos estarán próximos a promocionar a la categoría A, mientras otros estarán próximos a decaer hacia la categoría C.

**Categoría C**

Los productos de la categoría C son relativamente poco importantes. Típicamente representarán en torno al 50% de las referencias, pero constituyendo sólo un 5% del valor. Son productos de poca rentabilidad. Será incluso cuestionable si compensa mantener stock de algunas de estas referencias. No compensa dedicar una gran cantidad de recursos a su control de inventario, puesto que los costes de almacenaje y los costes operativos podría superar fácilmente la baja rentabilidad de estos productos.

(ATOX SISTEMAS DE ALMACENAJE, 2017)

### **2.6.2 Métodos de valoración o valuación de inventarios**

**Método 1: Primeros en entrar, primeros en salir (PEPS)**

FIFO (first in, first out) por sus siglas en inglés. Plantea que los primeros productos que se adquieren son los primeros que se venden, quedando en inventario los artículos más recientes.

Contrario a UEPS, este método plantea la suposición más real en la práctica. Lo más lógico es que se vendan aquellos artículos más con más tiempo en inventario logrando reducir el riesgo de obsolescencia o vencimiento.

Con PEPS, el inventario que queda en el almacén se valora a un precio cercano al valor que fija el mercado (considerando un período inflacionario). Es decir que los artículos que llevan más tiempo en el inventario, tienden a tener menor costo que los adquiridos recientemente.

Esto último es lo que nos lleva a tener un menor costo de ventas y con él un aumento del beneficio obtenido por la empresa, pero también aumento en el pago de impuestos.

**Método 2: Últimos en entrar, primeros en salir (UEPS)**

Last in, first out o LIFO. Este método de valuación es opuesto al PEPS al plantear que las ultimas unidades que ingresaron, serán las primeras en ser vendidas.

En la práctica, no es un método de amplío uso, pero suele ser enseñado en la academia al ser el opuesto a PEPS.

Esto sin mencionar las desventajas ocasionadas a partir de la naturaleza de los artículos. Si una empresa elije operar bajo la modalidad UEPS, su inventario podría quedarse obsoleto o estar sujeto a daños.

Por lo tanto, el método se aleja de la realidad ante variaciones significativas en el precio de los artículos. En un período inflacionario la empresa saldrá inicialmente del inventario más reciente que tiene (las últimas que entraron), lo que nos llevará a tener un inventario infravalorado o subvalorado.

Eso indica que el método UEPS otorga una reducción en la carga tributaria si operamos bajo una economía inflacionaria.

Pero, por otra parte, si la economía atraviesa caída en los precios (deflación) ocurrirá justo lo contrario. Nos llevará a tener un inventario sobrevalorado, pues el más reciente se adquiere a menores precios de lo que se adquirió el más antiguo.

**Método 3: Promedio ponderado**

En este método promediamos el costo de los inventarios en existencia, por ende, con cada unidad que ingresa (las que se compran) se actualiza el promedio.

Dicho de otra forma, con este método dividimos el costo de los artículos puestos a la venta por el número de unidades que tenemos en existencia.

Tengamos en cuenta que los artículos disponibles a la venta los componen aquellos que están por inventario inicial sumados a los que se van comprando. La cifra que obtenemos tras la división se usa para asignar un costo al inventario inicial o un costo de los artículos vendidos.

Así pues, nuestra cantidad de inventario en existencia cuenta con un valor producto de las unidades más antiguas y más nuevas adquiridas.

(IngenioEmpresa, 2018)

# **Capítulo 3: Marco Teórico**

## **3.1 Herramientas utilizadas**

### **3.1.1 Android Studio**

Android Studio es el [entorno de desarrollo integrado](https://es.wikipedia.org/wiki/Ambiente_de_desarrollo_integrado) oficial para la plataforma [Android](https://es.wikipedia.org/wiki/Android). Fue anunciado el [16 de mayo](https://es.wikipedia.org/wiki/16_de_mayo) de [2013](https://es.wikipedia.org/wiki/2013) en la conferencia [Google I/O](https://es.wikipedia.org/wiki/Google_I/O), y reemplazó a [Eclipse](https://es.wikipedia.org/wiki/Eclipse_(software)) como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014. Está basado en el software [IntelliJ IDEA de JetBrains](https://es.wikipedia.org/wiki/IntelliJ_IDEA) y ha sido publicado de forma gratuita a través de la [Licencia Apache 2.0](https://es.wikipedia.org/wiki/Apache_License). Está disponible para las plataformas [Microsoft Windows](https://es.wikipedia.org/wiki/Microsoft_Windows), [macOS](https://es.wikipedia.org/wiki/MacOS) y [GNU/Linux](https://es.wikipedia.org/wiki/GNU/Linux). Ha sido diseñado específicamente para el desarrollo de Android.

(Wikipedia, 2019)



Ilustración 5: Logo Android Studio

**Características Android Studio**

Las siguientes características se proporcionan en la versión estable actual:

* Integración de ProGuard y funciones de firma de aplicaciones.
* Más Especificación a la hora de Programar.
* Renderizado en tiempo real.
* Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
* Soporte para construcción basada en [Gradle](https://es.wikipedia.org/wiki/Gradle).
* [Refactorización](https://es.wikipedia.org/wiki/Refactorizaci%C3%B3n) específica de Android y arreglos rápidos.
* Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario.
* Herramientas [Lint](https://es.wikipedia.org/wiki/Lint) para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
* Plantillas para crear diseños comunes de Android y otros componentes.
* Soporte para programar aplicaciones para [Android Wear](https://es.wikipedia.org/wiki/Android_Wear).
* Soporte integrado para Google Cloud Platform, que permite la integración con Google Cloud Messaging y App Engine.
* Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.

(Wikipedia, 2019)

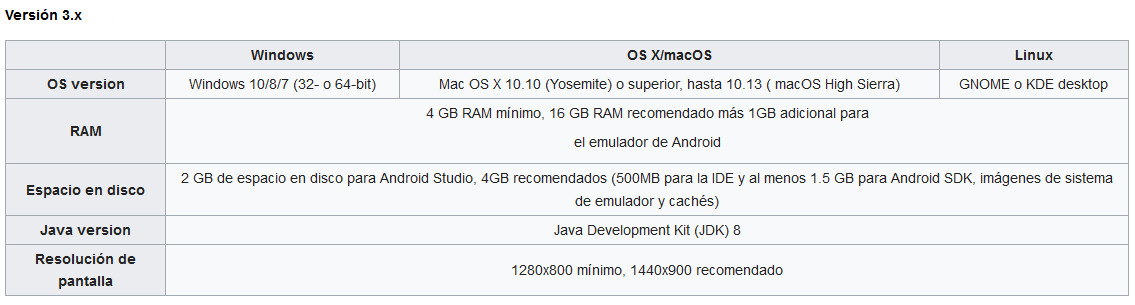
**Requisitos del sistema para usar Android Studio**

Tabla 1: Requisitos Android Studio Versión 3.x

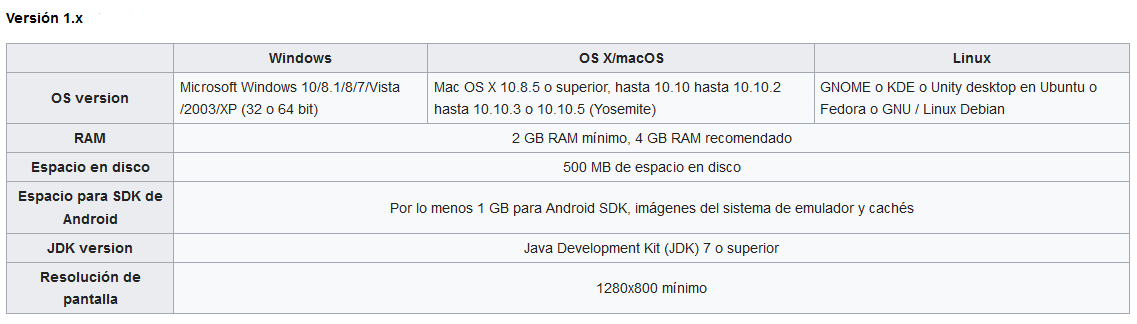


Tabla 3: Requisitos Android Studio Versión 1.x

Tabla 2: Requisitos Android Studio Versión 2.x

### **3.1.2 Firebase**

Firebase es una plataforma para el desarrollo de [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) y aplicaciones móviles desarrollada por [Google](https://es.wikipedia.org/wiki/Google) en 2014.

Es una plataforma ubicada en la nube, integrada con [Google Cloud Platform](https://es.wikipedia.org/wiki/Google_Cloud), que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

Los desarrolladores tendrán una serie de ventajas al usar esta plataforma:

* Sincronizar fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
* Usa un conjunto de herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como [IOS](https://es.wikipedia.org/wiki/IOS), [Android](https://es.wikipedia.org/wiki/Android), [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web), [Unity](https://es.wikipedia.org/wiki/Unity_(entorno_de_escritorio)) y [C++](https://es.wikipedia.org/wiki/C%2B%2B).
* Usa la infraestructura de Google y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.
* Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los [SDK](https://es.wikipedia.org/wiki/SDK) para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

Firebase dota a sus usuarios de una gran documentación para crear aplicaciones usando esta plataforma. Aparte de esto, ofrece soporte gratuito mediante correo electrónico para todos sus usuarios, y además sus desarrolladores participan activamente en plataformas como [Github](https://es.wikipedia.org/wiki/GitHub) y [StackOverflow](https://es.wikipedia.org/wiki/Stack_Overflow), así como poseen un canal de Youtube explicando el funcionamiento de varias de sus herramientas.

Gracias a todas estas funcionalidades, cualquier desarrollador puede combinar y adaptar la plataforma a medida de sus necesidades.

(Wikipedia, 2019)



Ilustración 6: Logo Firebase

**Firebase Auth**

Firebase Auth es un servicio que puede autenticar los usuarios utilizando únicamente código del lado del cliente. Incluye la autenticación mediante proveedores de inicio de sesión como [Facebook](https://es.wikipedia.org/wiki/Facebook), [GitHub](https://es.wikipedia.org/wiki/GitHub), [Twitter](https://es.wikipedia.org/wiki/Twitter), Google, [Yahoo](https://es.wikipedia.org/wiki/Yahoo!) y [Microsoft](https://es.wikipedia.org/wiki/Microsoft); así como los métodos clásicos de inicio de sesión mediante correo electrónico y contraseña. Además, incluye un sistema de administración del usuario por el cual los desarrolladores pueden habilitar la autenticación de usuarios con email y contraseña que se almacenarán en Firebase.

Este servicio busca facilitar la creación de sistemas de autenticación, a la vez que mejora la incorporación, acceso y seguridad para los usuarios. Gracias a esto, el cliente no tiene que preocuparse por desarrollar métodos de autenticación clásicos, ya que Firebase le aporta de manera sencilla, eficaz y segura métodos para gestionar sus usuarios.

También aporta muchas funcionalidades extra, como la recuperación y verificación de cuentas, tanto por correo electrónico como por SMS, y cuotas de registro para los usuarios, todo esto gestionado mediante los servidores de la plataforma.

(Wikipedia, 2019)

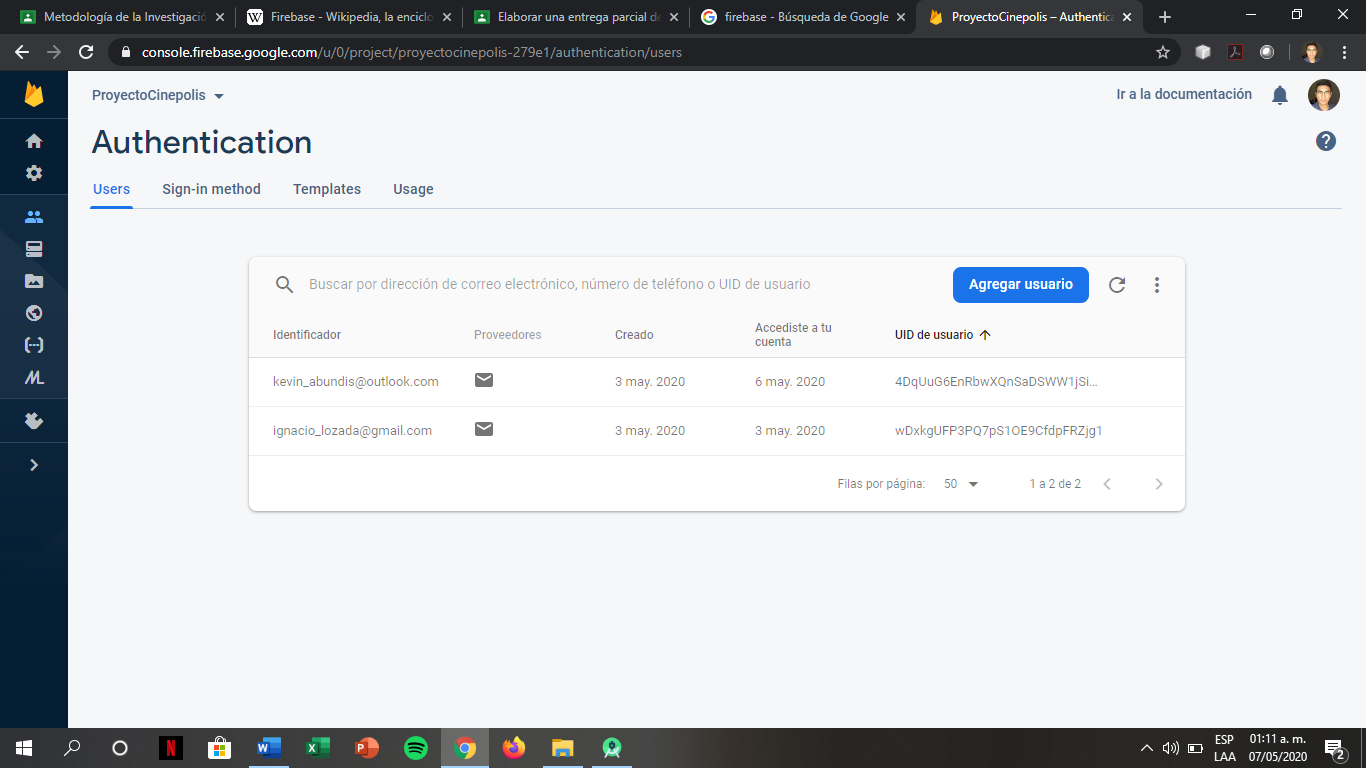


Ilustración 7: Opción de Auth (Authentication) en Firebase

**Realtime Database**

Firebase proporciona una base de datos en tiempo real, back-end y organizada en forma de árbol [JSON](https://es.wikipedia.org/wiki/JSON). El servicio proporciona a los desarrolladores de aplicaciones una [API](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase.​ La compañía habilita integración con aplicaciones [Android](https://es.wikipedia.org/wiki/Android), [iOS](https://es.wikipedia.org/wiki/IOS), [JavaScript](https://es.wikipedia.org/wiki/JavaScript), Java, [Objective-C](https://es.wikipedia.org/wiki/Objective-C), Swift y [Node.js](https://es.wikipedia.org/wiki/Node.js). La base de datos es también accesible a través de una REST API e integración para varios sistemas de Javascript como [AngularJS](https://es.wikipedia.org/wiki/AngularJS), [React](https://es.wikipedia.org/wiki/React), Ember.js y [Backbone.js](https://es.wikipedia.org/wiki/Backbone.js). La REST API utiliza el protocolo SSE (del inglés Server-Sent Events), el cual es una API para crear conexiones de HTTP para recibir notificaciones push de un servidor.

La sincronización en tiempo real de esta base de datos permite que los usuarios accedan a la información de sus datos desde cualquier dispositivo en tiempo real, compartiendo una instancia de Realtime Database, y cada vez que un usuario realice una modificación en esta, se almacena dicha información en la nube y se notifica simultáneamente al resto de dispositivos.

Una funcionalidad interesante de esta base de datos, es que, si un usuario realiza cambios y pierde a la vez su conexión a Internet, el SDK de la plataforma usa una caché local en el dispositivo donde guarda estos cambios; y una vez que vuelva a tener conexión, automáticamente se sincronizan los datos locales.

(Wikipedia, 2019)

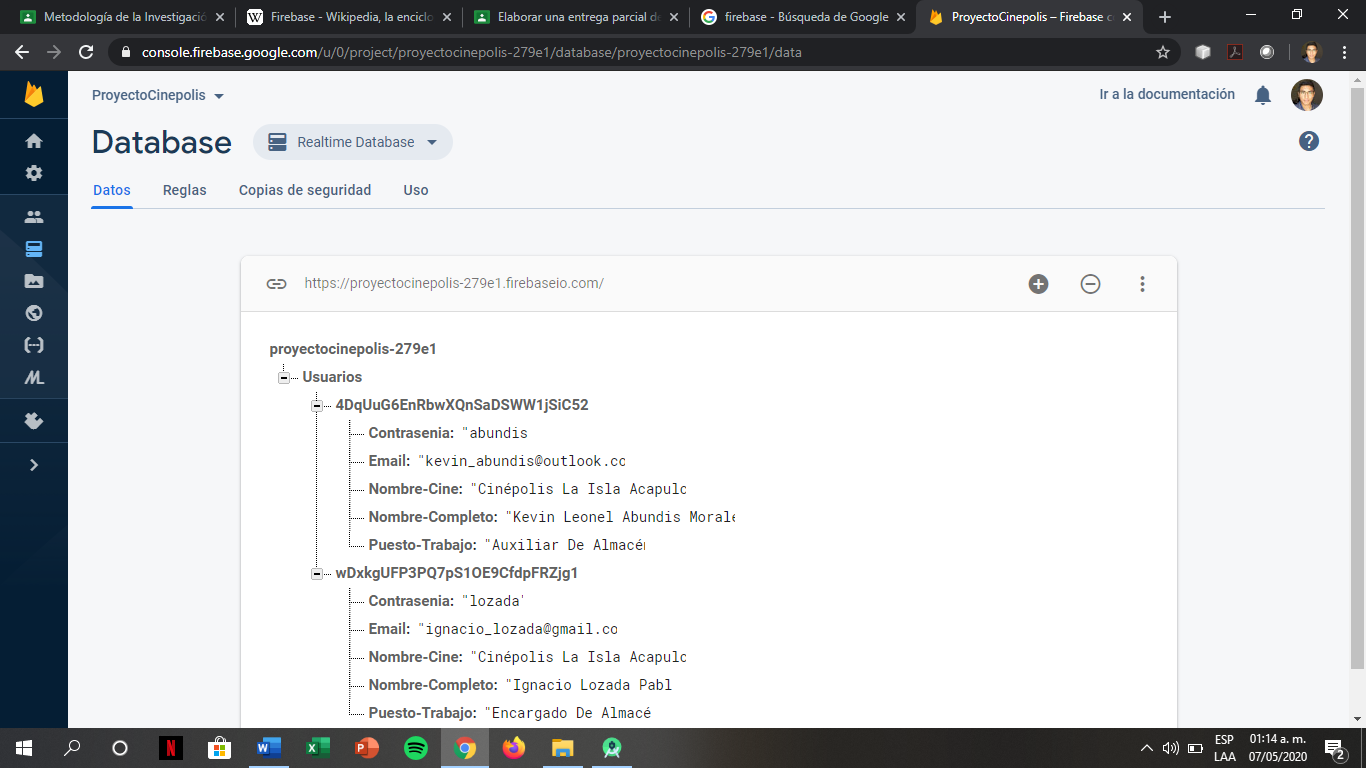


Ilustración 8: Opción de Realtime Database en Firebase

### **3.1.3 Enterprise Architect**

Sparx Systems Enterprise Architect es una herramienta de diseño y modelado visual basada en OMG UML. La plataforma admite: el diseño y la construcción de sistemas de software; modelado de procesos comerciales; y modelado de dominios basados ​​en la industria. Las empresas y organizaciones lo utilizan no solo para modelar la arquitectura de sus sistemas, sino también para procesar la implementación de estos modelos en todo el ciclo de vida de desarrollo de aplicaciones.

(Wikipedia, 2019)

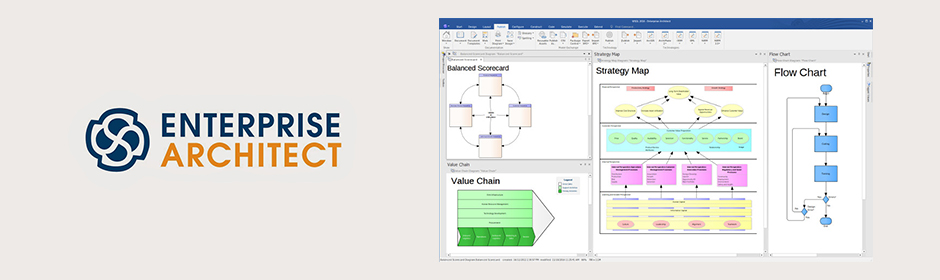
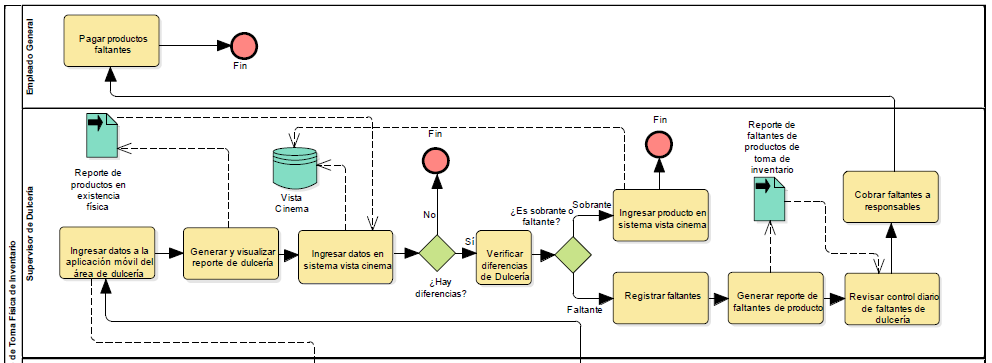


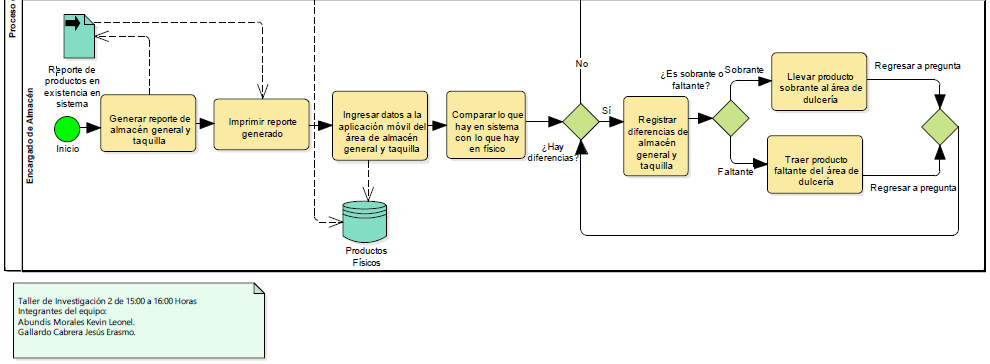
Ilustración 9: Enterprise Architect

# **Capítulo 4: Metodología**

## **4.1 Modelado de Negocio**

BPMN del Proceso de Toma Física de Inventario de Cinépolis La Isla Acapulco.





Documentación del Proceso de Toma Física de Inventario

## Proceso de Toma Física de Inventario

Pool «Pool» owned by 'CollaborationModel', in package 'Cinépolis La Isla Acapulco'

| ELEMENTS OWNED BY Proceso de Toma Física de Inventario |
| --- |
| Empleado General : Lane «Lane» |
| Encargado de Almacén : Lane «Lane» |
| Supervisor de Dulcería : Lane «Lane» |

### Empleado General

Lane «Lane» owned by 'Proceso de Toma Física de Inventario', in package 'Cinépolis La Isla Acapulco'

| ELEMENTS OWNED BY Empleado General |
| --- |
| Pagar productos faltantes : Activity «Activity»  Los empleados generales responsables de los faltantes, tienen que pagarlos. |
| Fin : EndEvent «EndEvent» |

#### Pagar productos faltantes

Activity «Activity» owned by 'Empleado General', in package 'Cinépolis La Isla Acapulco'

Los empleados generales responsables de los faltantes, tienen que pagarlos.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Pagar productos faltantes to «EndEvent» Fin |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Cobrar faltantes a responsables to «Activity» Pagar productos faltantes |

#### Fin

EndEvent «EndEvent» owned by 'Empleado General', in package 'Cinépolis La Isla Acapulco'

### Encargado de Almacén

Lane «Lane» owned by 'Proceso de Toma Física de Inventario', in package 'Cinépolis La Isla Acapulco'

| ELEMENTS OWNED BY Encargado de Almacén |
| --- |
| Comparar lo que hay en sistema con lo que hay en físico : Activity «Activity»  Se compara los productos que hay en sistema contra los datos que arrojó la aplicación móvil que es los productos que hay en físico. |
| Generar reporte de almacén general y taquilla : Activity «Activity»  Este reporte que se genera muestra los productos que hay en existencia en sistema de almacén general y taquilla. Este reporte es generado en el sistema vista cinema. |
| Imprimir reporte generado : Activity «Activity»  Se imprime el reporte generado anteriormente para empezar a realizar el conteo físico del almacén general y taquilla. |
| Ingresar datos a la aplicación móvil del área de almacén general y taquilla : Activity «Activity»  Se deberán ingresar los datos del área de almacén general y taquilla, para que la aplicación realize los cálculos correspondientes. |
| Llevar producto sobrante al área de dulcería : Activity «Activity»  Si hay sobrante, se lleva el producto sobrante al área de dulcería para que los productos que hay en sistema con lo que hay en físico en almacén y taquilla sean iguales. |
| Registrar diferencias de almacén general y taquilla : Activity «Activity»  Si hay diferencias en el almacén general y taquilla se registran en un documento, para después decidir que se hará con esas diferencias. |
| Traer producto faltante del área de dulcería : Activity «Activity»  Si hay faltante, se trae el producto faltante del área de dulcería para que los productos que hay en sistema con lo que hay en físico en almacén y taquilla sean iguales. |
| Productos Físicos : DataStore «DataStore» |
| Reporte de productos en existencia en sistema : DataObject «DataObject» |
| <anonymous> : Gateway «Gateway» |
| ¿Es sobrante o faltante? : Gateway «Gateway» |
| ¿Hay diferencias? : Gateway «Gateway» |
| Inicio : StartEvent «StartEvent» |

#### Comparar lo que hay en sistema con lo que hay en físico

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Se compara los productos que hay en sistema contra los datos que arrojó la aplicación móvil que es los productos que hay en físico.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Comparar lo que hay en sistema con lo que hay en físico to «Gateway» ¿Hay diferencias? |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar datos a la aplicación móvil del área de almacén general y taquilla to «Activity» Comparar lo que hay en sistema con lo que hay en físico |

#### Generar reporte de almacén general y taquilla

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Este reporte que se genera muestra los productos que hay en existencia en sistema de almacén general y taquilla. Este reporte es generado en el sistema vista cinema.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Generar reporte de almacén general y taquilla to «Activity» Imprimir reporte generado |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «StartEvent» Inicio to «Activity» Generar reporte de almacén general y taquilla |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Generar reporte de almacén general y taquilla : Activity, Public  To: Reporte de productos en existencia en sistema : DataObject, Public |

#### Imprimir reporte generado

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Se imprime el reporte generado anteriormente para empezar a realizar el conteo físico del almacén general y taquilla.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Imprimir reporte generado to «Activity» Ingresar datos a la aplicación móvil del área de almacén general y taquilla |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Generar reporte de almacén general y taquilla to «Activity» Imprimir reporte generado |

| CONNECTORS |
| --- |
| DataInputAssociation «DataInputAssociation» Source -> Destination  From: Reporte de productos en existencia en sistema : DataObject, Public  To: Imprimir reporte generado : Activity, Public |

#### Ingresar datos a la aplicación móvil del área de almacén general y taquilla

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Se deberán ingresar los datos del área de almacén general y taquilla, para que la aplicación realize los cálculos correspondientes.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar datos a la aplicación móvil del área de almacén general y taquilla to «Activity» Comparar lo que hay en sistema con lo que hay en físico |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Imprimir reporte generado to «Activity» Ingresar datos a la aplicación móvil del área de almacén general y taquilla |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar datos a la aplicación móvil del área de almacén general y taquilla : Activity, Public  To: Productos Físicos : DataStore, Public |

#### Llevar producto sobrante al área de dulcería

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Si hay sobrante, se lleva el producto sobrante al área de dulcería para que los productos que hay en sistema con lo que hay en físico en almacén y taquilla sean iguales.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Regresar a pregunta  SequenceFlow «SequenceFlow» from «Activity» Llevar producto sobrante al área de dulcería to «Gateway» <anonymous> |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sobrante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Llevar producto sobrante al área de dulcería |

#### Registrar diferencias de almacén general y taquilla

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Si hay diferencias en el almacén general y taquilla se registran en un documento, para después decidir que se hará con esas diferencias.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Registrar diferencias de almacén general y taquilla to «Gateway» ¿Es sobrante o faltante? |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sí  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «Activity» Registrar diferencias de almacén general y taquilla |

#### Traer producto faltante del área de dulcería

Activity «Activity» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

Si hay faltante, se trae el producto faltante del área de dulcería para que los productos que hay en sistema con lo que hay en físico en almacén y taquilla sean iguales.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Regresar a pregunta  SequenceFlow «SequenceFlow» from «Activity» Traer producto faltante del área de dulcería to «Gateway» <anonymous> |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Faltante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Traer producto faltante del área de dulcería |

#### Productos Físicos

DataStore «DataStore» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar datos a la aplicación móvil del área de dulcería : Activity, Public  To: Productos Físicos : DataStore, Public |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar datos a la aplicación móvil del área de almacén general y taquilla : Activity, Public  To: Productos Físicos : DataStore, Public |

#### Reporte de productos en existencia en sistema

DataObject «DataObject» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

| CONNECTORS |
| --- |
| DataInputAssociation «DataInputAssociation» Source -> Destination  From: Reporte de productos en existencia en sistema : DataObject, Public  To: Imprimir reporte generado : Activity, Public |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Generar reporte de almacén general y taquilla : Activity, Public  To: Reporte de productos en existencia en sistema : DataObject, Public |

#### <anonymous>

Gateway «Gateway» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Gateway» <anonymous> to «Gateway» ¿Hay diferencias? |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Regresar a pregunta  SequenceFlow «SequenceFlow» from «Activity» Traer producto faltante del área de dulcería to «Gateway» <anonymous> |
| Name: Regresar a pregunta  SequenceFlow «SequenceFlow» from «Activity» Llevar producto sobrante al área de dulcería to «Gateway» <anonymous> |

#### ¿Es sobrante o faltante?

Gateway «Gateway» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sobrante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Llevar producto sobrante al área de dulcería |
| Name: Faltante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Traer producto faltante del área de dulcería |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Registrar diferencias de almacén general y taquilla to «Gateway» ¿Es sobrante o faltante? |

#### ¿Hay diferencias?

Gateway «Gateway» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: No  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «Activity» Ingresar datos a la aplicación móvil del área de dulcería |
| Name: Sí  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «Activity» Registrar diferencias de almacén general y taquilla |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Gateway» <anonymous> to «Gateway» ¿Hay diferencias? |
| SequenceFlow «SequenceFlow» from «Activity» Comparar lo que hay en sistema con lo que hay en físico to «Gateway» ¿Hay diferencias? |

#### Inicio

StartEvent «StartEvent» owned by 'Encargado de Almacén', in package 'Cinépolis La Isla Acapulco'

### Supervisor de Dulcería

Lane «Lane» owned by 'Proceso de Toma Física de Inventario', in package 'Cinépolis La Isla Acapulco'

| ELEMENTS OWNED BY Supervisor de Dulcería |
| --- |
| Cobrar faltantes a responsables : Activity «Activity»  El supervisor de dulcería es el encargado de llevar a cabo el cobró a los responsables de los faltantes. |
| Generar reporte de faltantes de producto : Activity «Activity»  Se genera el reporte de faltantes de productos de la toma de inventario. |
| Generar y visualizar reporte de dulcería : Activity «Activity»  Se genera y se visualiza el reporte de los productos en existencia física del área de dulcería. |
| Ingresar datos a la aplicación móvil del área de dulcería : Activity «Activity»  Se deberán ingresar los datos ahora del área de dulcería, para que la aplicación realize los cálculos correspondientes. |
| Ingresar datos en sistema vista cinema : Activity «Activity»  Se ingresan los datos de los productos en existencia física en el sistema vista cinema para que realize los cálculos correspondientes. |
| Ingresar producto en sistema vista cinema : Activity «Activity»  Si hay sobrante de producto en el área de dulcería, se ingresa el producto en el sistema vista cinema, así para que lo que hay en sistema y lo que hay en físico coincida. |
| Registrar faltantes : Activity «Activity»  Si hay faltante de producto se registra para posteriormente cobrar esas diferencias. |
| Revisar control diario de faltantes de dulcería : Activity «Activity»  Utilizando el reporte de faltantes de productos de la toma física, se revisa en el control diario de faltantes que se lleva a cabo del área de dulcería, los responsables que tuvieron esos faltantes. |
| Verificar diferencias de Dulcería : Activity «Activity»  Verificar las diferencias exhaustivas, porque al momento de tomar la decisión de ingresarlas al sistema o de cobrar esas diferencias, ya no pueden modificarse los valores en el sistema vista cinema. |
| Reporte de faltantes de productos de toma de inventario : DataObject «DataObject» |
| Reporte de productos en existencia física : DataObject «DataObject» |
| Vista Cinema : DataStore «DataStore» |
| ¿Es sobrante o faltante? : Gateway «Gateway» |
| ¿Hay diferencias? : Gateway «Gateway» |
| Fin : EndEvent «EndEvent» |
| Fin : EndEvent «EndEvent» |

#### Cobrar faltantes a responsables

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

El supervisor de dulcería es el encargado de llevar a cabo el cobró a los responsables de los faltantes.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Cobrar faltantes a responsables to «Activity» Pagar productos faltantes |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Revisar control diario de faltantes de dulcería to «Activity» Cobrar faltantes a responsables |

#### Generar reporte de faltantes de producto

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Se genera el reporte de faltantes de productos de la toma de inventario.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Generar reporte de faltantes de producto to «Activity» Revisar control diario de faltantes de dulcería |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Registrar faltantes to «Activity» Generar reporte de faltantes de producto |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Generar reporte de faltantes de producto : Activity, Public  To: Reporte de faltantes de productos de toma de inventario : DataObject, Public |

#### Generar y visualizar reporte de dulcería

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Se genera y se visualiza el reporte de los productos en existencia física del área de dulcería.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Generar y visualizar reporte de dulcería to «Activity» Ingresar datos en sistema vista cinema |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar datos a la aplicación móvil del área de dulcería to «Activity» Generar y visualizar reporte de dulcería |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Generar y visualizar reporte de dulcería : Activity, Public  To: Reporte de productos en existencia física : DataObject, Public |

#### Ingresar datos a la aplicación móvil del área de dulcería

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Se deberán ingresar los datos ahora del área de dulcería, para que la aplicación realize los cálculos correspondientes.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar datos a la aplicación móvil del área de dulcería to «Activity» Generar y visualizar reporte de dulcería |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: No  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «Activity» Ingresar datos a la aplicación móvil del área de dulcería |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar datos a la aplicación móvil del área de dulcería : Activity, Public  To: Productos Físicos : DataStore, Public |

#### Ingresar datos en sistema vista cinema

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Se ingresan los datos de los productos en existencia física en el sistema vista cinema para que realize los cálculos correspondientes.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar datos en sistema vista cinema to «Gateway» ¿Hay diferencias? |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Generar y visualizar reporte de dulcería to «Activity» Ingresar datos en sistema vista cinema |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar datos en sistema vista cinema : Activity, Public  To: Vista Cinema : DataStore, Public |
| DataInputAssociation «DataInputAssociation» Source -> Destination  From: Reporte de productos en existencia física : DataObject, Public  To: Ingresar datos en sistema vista cinema : Activity, Public |

#### Ingresar producto en sistema vista cinema

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Si hay sobrante de producto en el área de dulcería, se ingresa el producto en el sistema vista cinema, así para que lo que hay en sistema y lo que hay en físico coincida.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar producto en sistema vista cinema to «EndEvent» Fin |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sobrante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Ingresar producto en sistema vista cinema |

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar producto en sistema vista cinema : Activity, Public  To: Vista Cinema : DataStore, Public |

#### Registrar faltantes

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Si hay faltante de producto se registra para posteriormente cobrar esas diferencias.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Registrar faltantes to «Activity» Generar reporte de faltantes de producto |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Faltante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Registrar faltantes |

#### Revisar control diario de faltantes de dulcería

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Utilizando el reporte de faltantes de productos de la toma física, se revisa en el control diario de faltantes que se lleva a cabo del área de dulcería, los responsables que tuvieron esos faltantes.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Revisar control diario de faltantes de dulcería to «Activity» Cobrar faltantes a responsables |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Generar reporte de faltantes de producto to «Activity» Revisar control diario de faltantes de dulcería |

| CONNECTORS |
| --- |
| DataInputAssociation «DataInputAssociation» Source -> Destination  From: Reporte de faltantes de productos de toma de inventario : DataObject, Public  To: Revisar control diario de faltantes de dulcería : Activity, Public |

#### Verificar diferencias de Dulcería

Activity «Activity» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

Verificar las diferencias exhaustivas, porque al momento de tomar la decisión de ingresarlas al sistema o de cobrar esas diferencias, ya no pueden modificarse los valores en el sistema vista cinema.

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Verificar diferencias de Dulcería to «Gateway» ¿Es sobrante o faltante? |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sí  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «Activity» Verificar diferencias de Dulcería |

#### Reporte de faltantes de productos de toma de inventario

DataObject «DataObject» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

| CONNECTORS |
| --- |
| DataInputAssociation «DataInputAssociation» Source -> Destination  From: Reporte de faltantes de productos de toma de inventario : DataObject, Public  To: Revisar control diario de faltantes de dulcería : Activity, Public |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Generar reporte de faltantes de producto : Activity, Public  To: Reporte de faltantes de productos de toma de inventario : DataObject, Public |

#### Reporte de productos en existencia física

DataObject «DataObject» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

| CONNECTORS |
| --- |
| DataInputAssociation «DataInputAssociation» Source -> Destination  From: Reporte de productos en existencia física : DataObject, Public  To: Ingresar datos en sistema vista cinema : Activity, Public |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Generar y visualizar reporte de dulcería : Activity, Public  To: Reporte de productos en existencia física : DataObject, Public |

#### Vista Cinema

DataStore «DataStore» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

| CONNECTORS |
| --- |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar datos en sistema vista cinema : Activity, Public  To: Vista Cinema : DataStore, Public |
| DataOutputAssociation «DataOutputAssociation» Source -> Destination  From: Ingresar producto en sistema vista cinema : Activity, Public  To: Vista Cinema : DataStore, Public |

#### ¿Es sobrante o faltante?

Gateway «Gateway» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sobrante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Ingresar producto en sistema vista cinema |
| Name: Faltante  SequenceFlow «SequenceFlow» from «Gateway» ¿Es sobrante o faltante? to «Activity» Registrar faltantes |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Verificar diferencias de Dulcería to «Gateway» ¿Es sobrante o faltante? |

#### ¿Hay diferencias?

Gateway «Gateway» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

| OUTGOING BEHAVIORAL RELATIONSHIPS |
| --- |
| Name: Sí  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «Activity» Verificar diferencias de Dulcería |
| Name: No  SequenceFlow «SequenceFlow» from «Gateway» ¿Hay diferencias? to «EndEvent» Fin |

| INCOMING BEHAVIORAL RELATIONSHIPS |
| --- |
| SequenceFlow «SequenceFlow» from «Activity» Ingresar datos en sistema vista cinema to «Gateway» ¿Hay diferencias? |

#### Fin

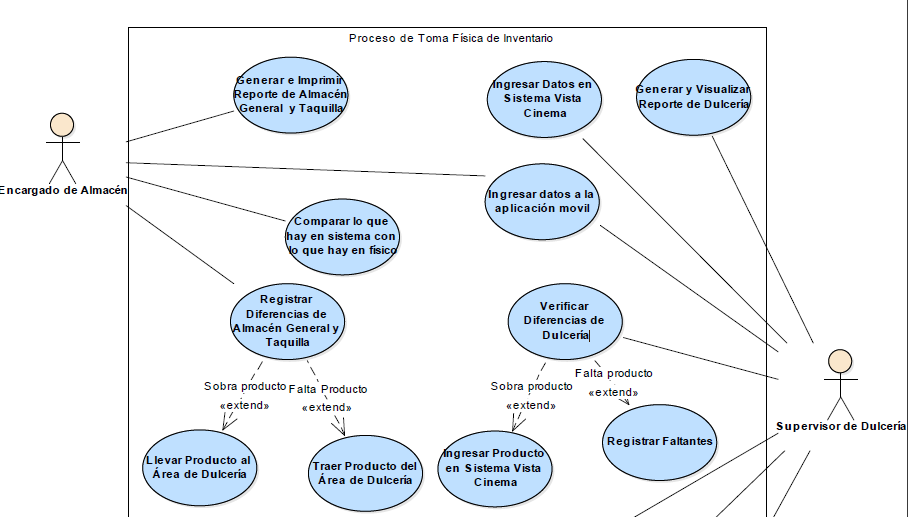
EndEvent «EndEvent» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

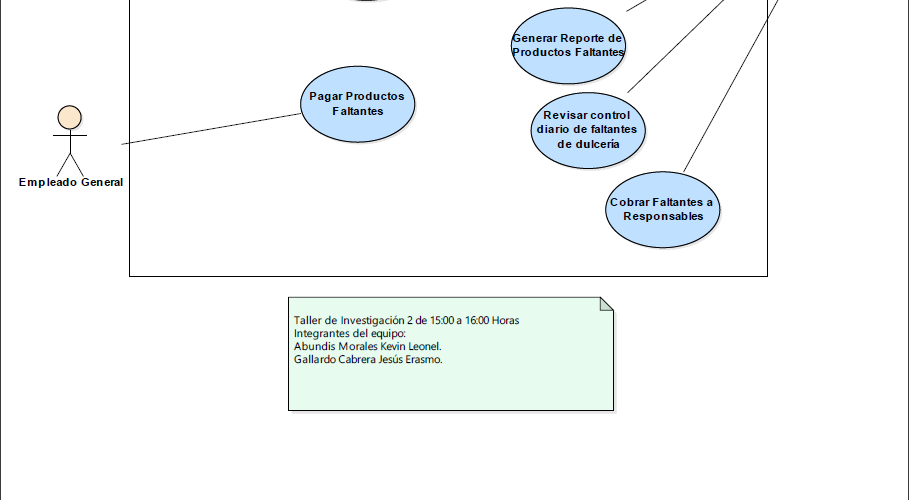
#### Fin

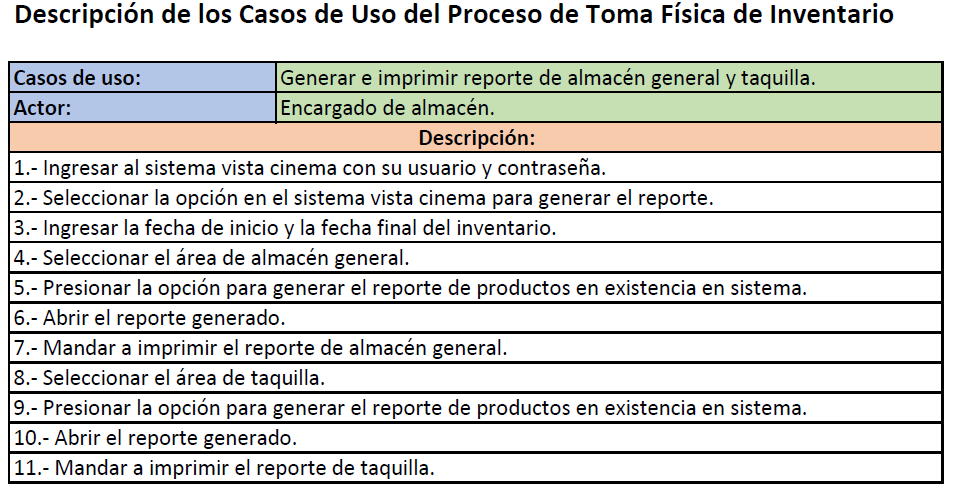
EndEvent «EndEvent» owned by 'Supervisor de Dulcería', in package 'Cinépolis La Isla Acapulco'

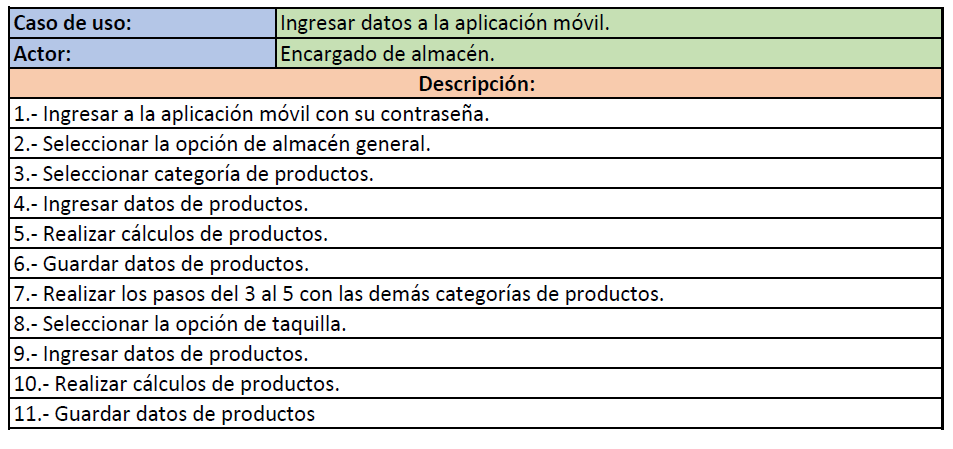
## **4.2 Casos de Uso**

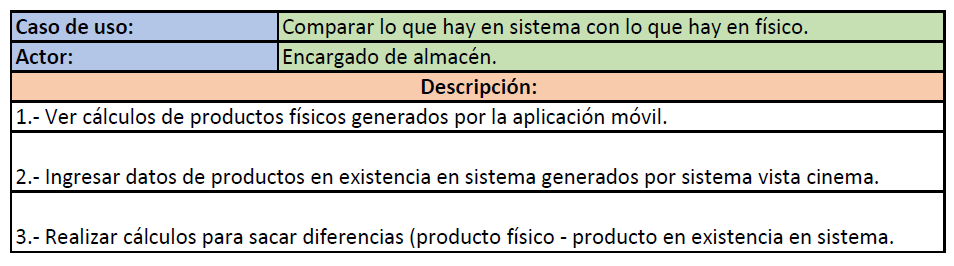
Diagrama de Casos de Uso del Proceso de Toma Física de Inventario de Cinépolis La Isla Acapulco.

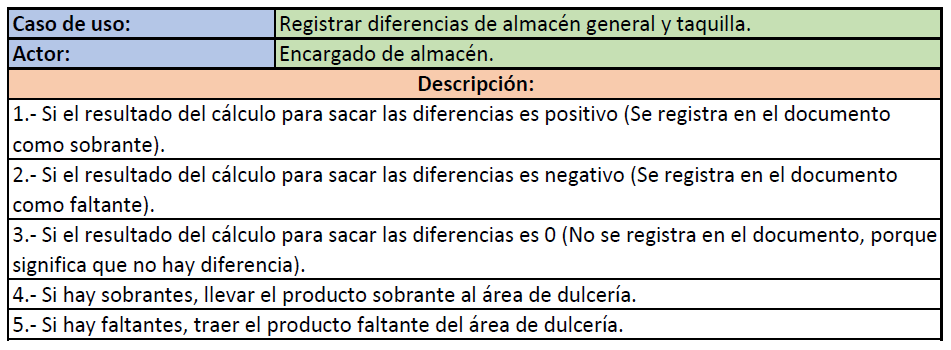


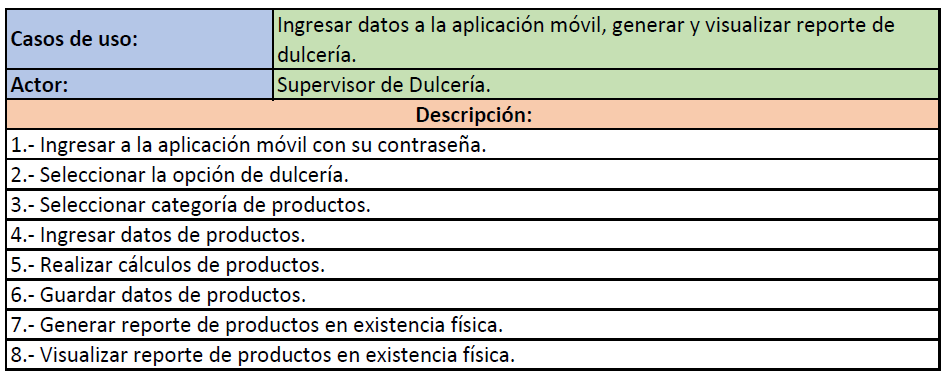


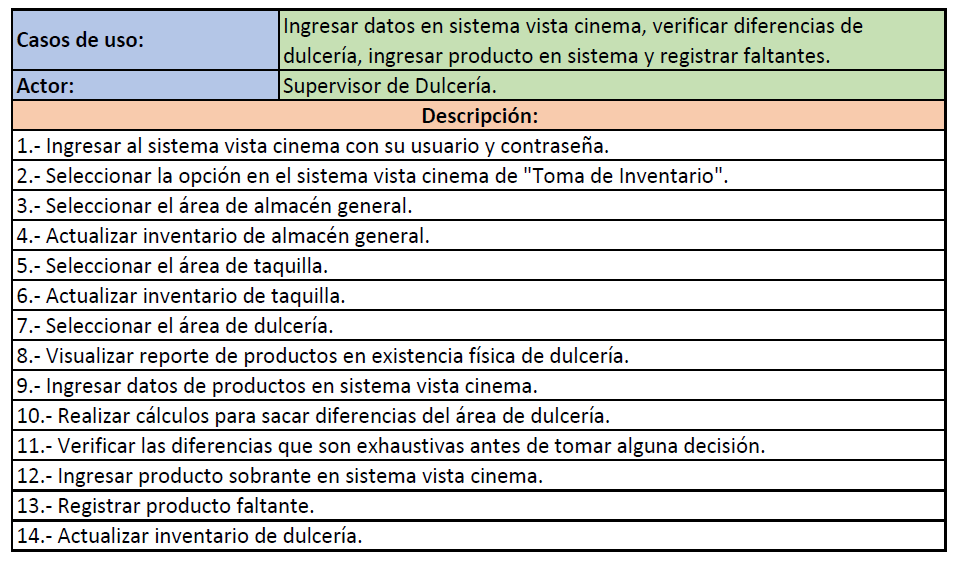


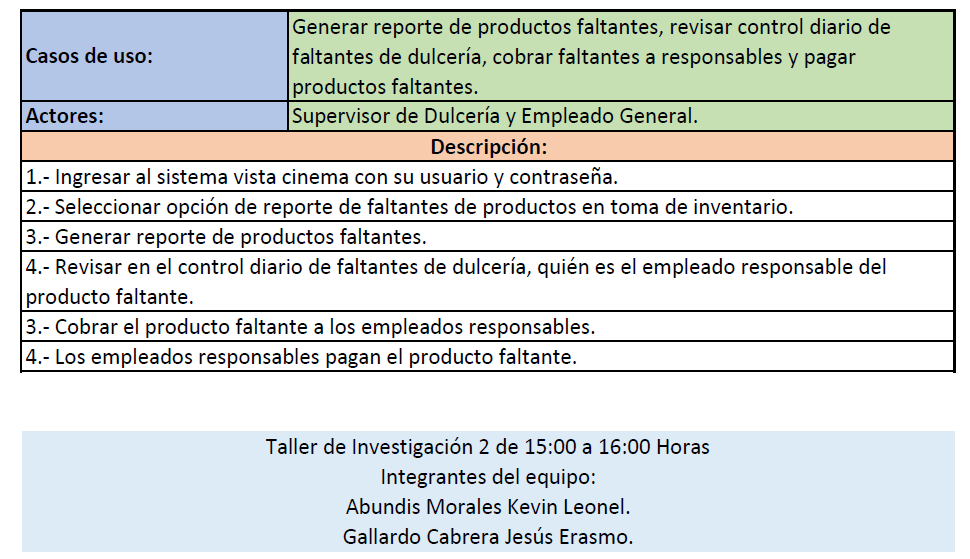




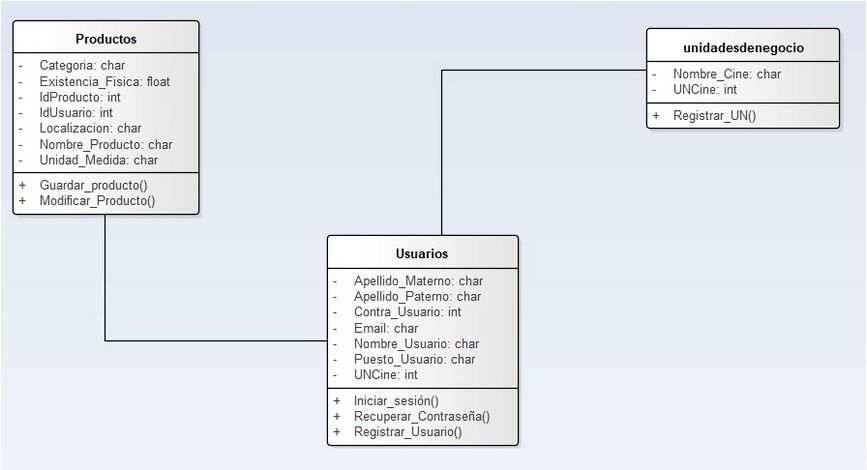






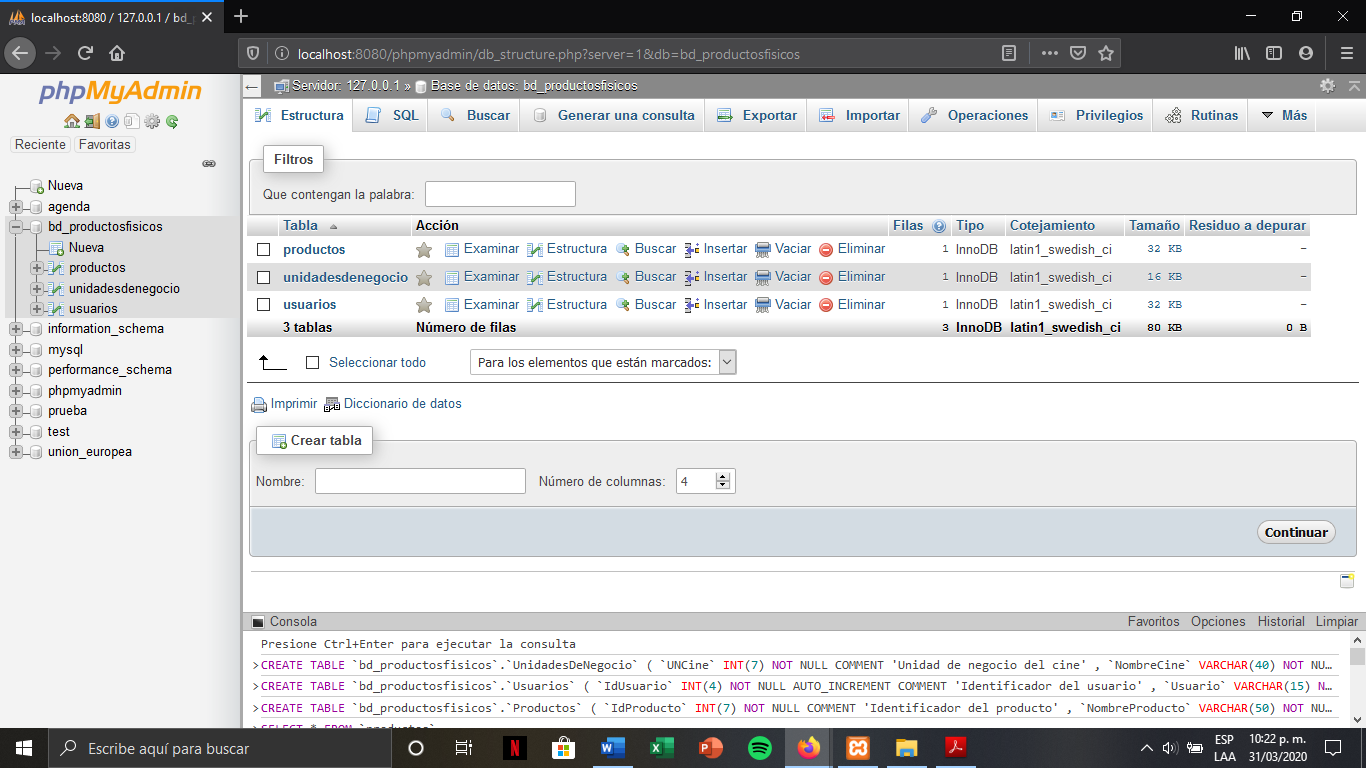


## **4.3 Diagrama de Clases**



## **4.4 Diseño de la Base de Datos. Diagrama de Entidad Asociación.**

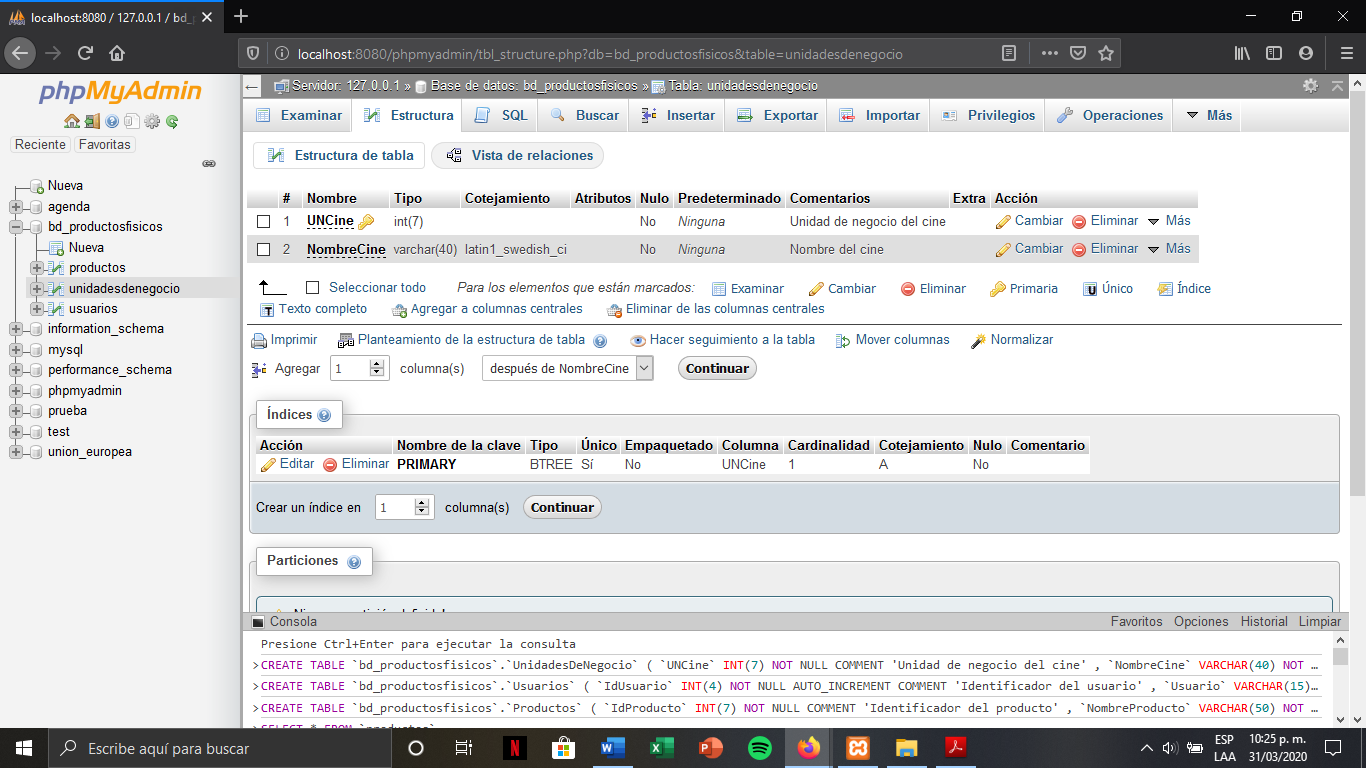
Tablas que contiene la Base de Datos bd\_productosfisicos.



Estructura de la tabla: productos.



Estructura de la tabla: unidadesdenegocio.



Estructura de la tabla: usuarios.

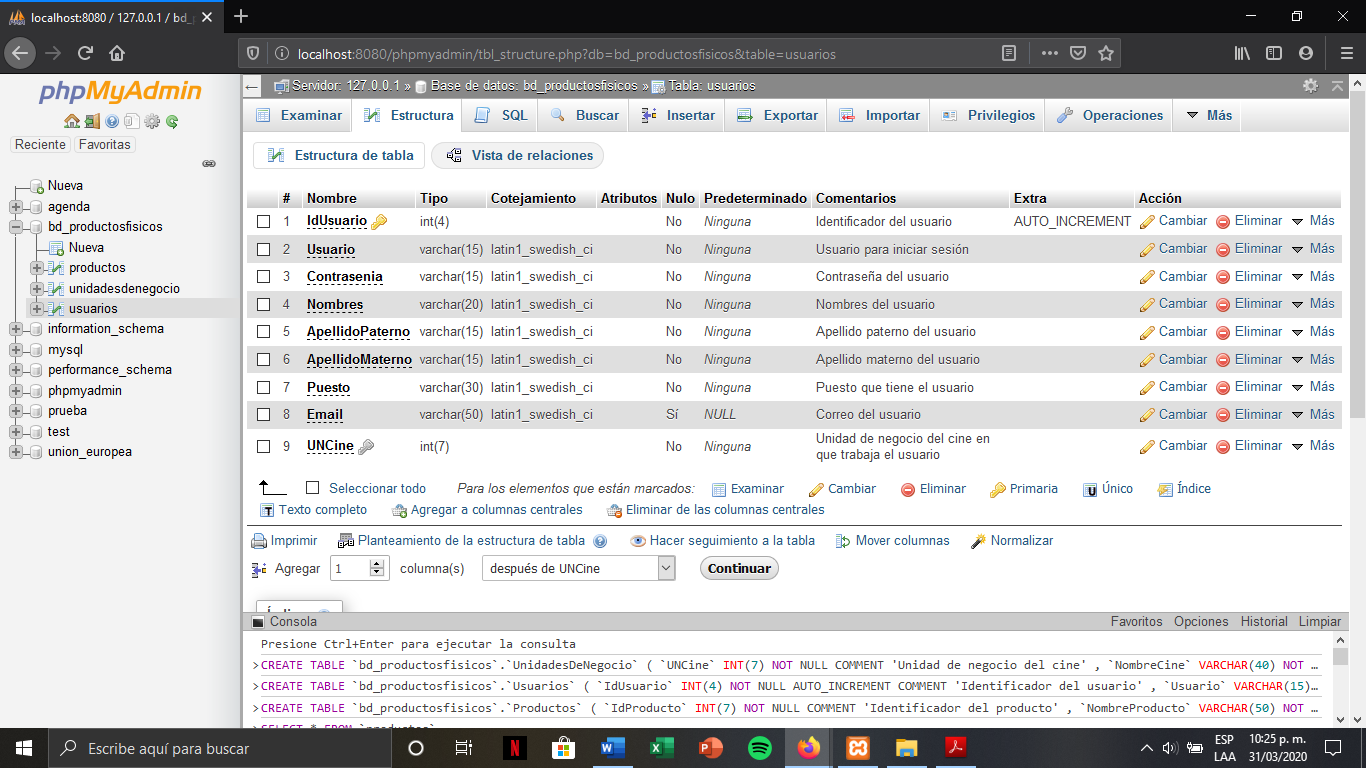
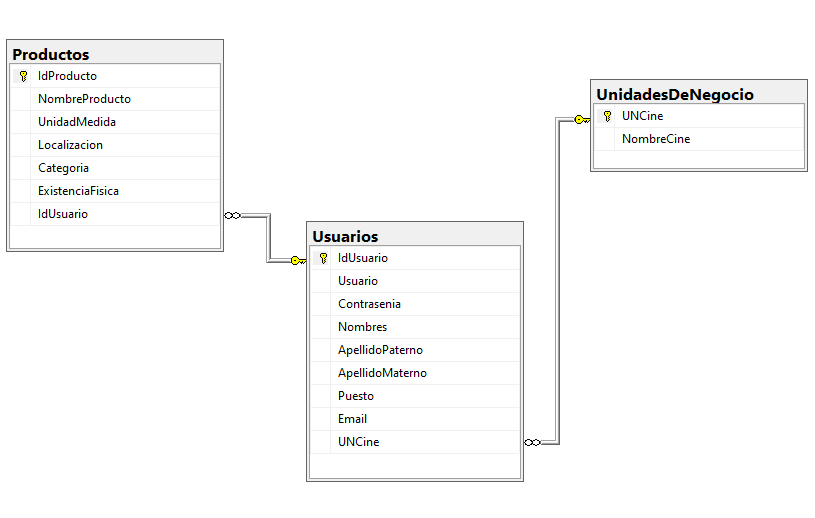


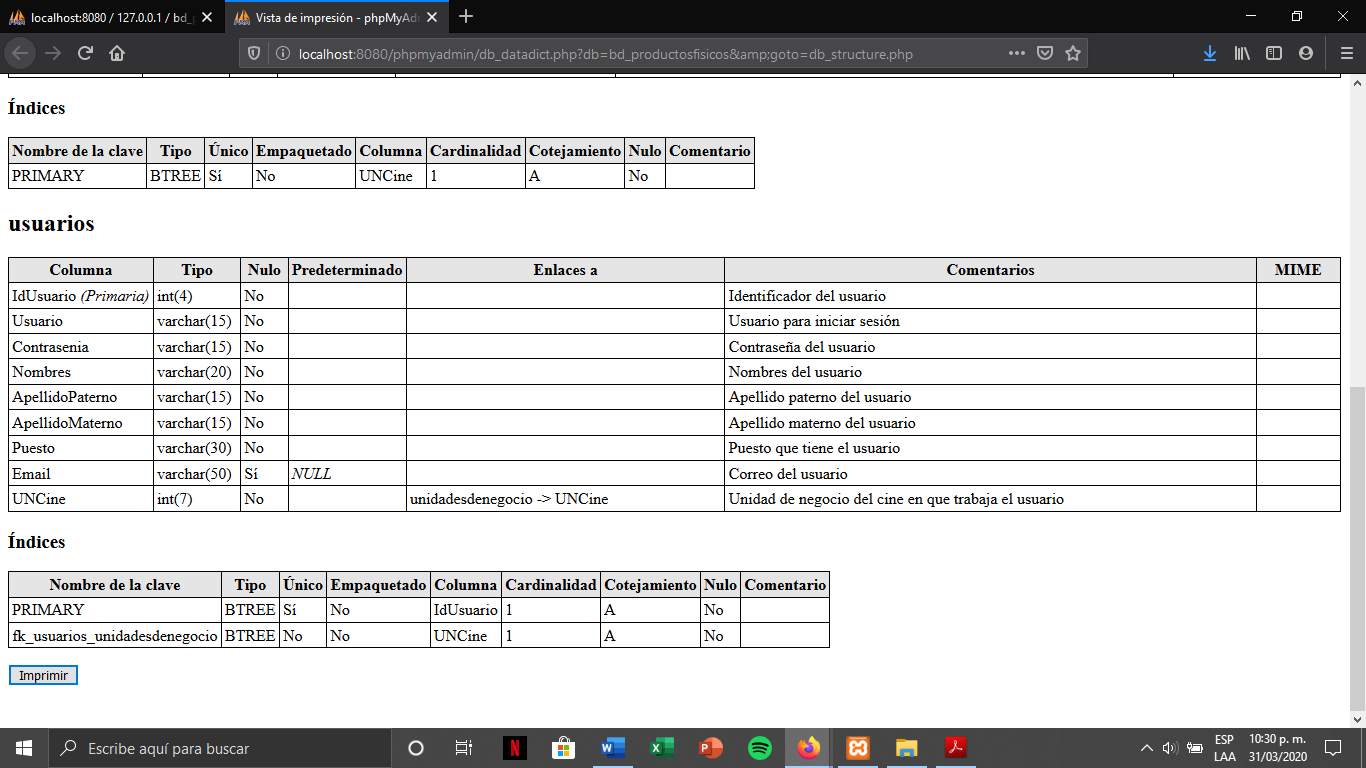
Diagrama Entidad Asociación de la Base de Datos BD\_ProductosFisicos.



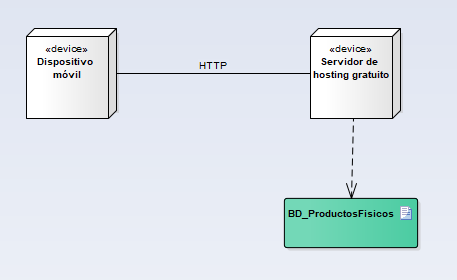
## **4.5 Diccionario de Datos**

Diccionario de datos de la Base de Datos bd\_productosfisicos.





## **4.6 Diagrama de Despliegue**



# **Capítulo 5: Implementación**

## **5.1 Instalación de Android Studio**

### **5.1.1 Requisitos previos**

Requisitos para Windows:

* Microsoft® Windows® 7/8/10 (32- or 64-bit)
* 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
* 2 GB de espacio disponible como mínimo,4 GB recomendados (500 MB para el IDE + 1.5 GB for Android SDK y emuladores)
* 1280 x 800 de resolución de pantalla

Requisitos para Mac:

* Mac® OS X® 10.10 (Yosemite) o superior - 10.13 (macOS High Sierra)
* 3 GB RAM como mínimo, 8 GB RAM recomendado; más 1 GB para el emulador de Android
* 2 GB de espacio disponible como mínimo,4 GB recomendados (500 MB para el IDE + 1.5 GB for Android SDK y emuladores)
* 1280 x 800 de resolución de pantalla

Requisitos para Linux:

* 2 GB de espacio disponible como mínimo,4 GB recomendados (500 MB para el IDE + 1.5 GB for Android SDK y emuladores)
* 1280 x 800 de resolución de pantalla

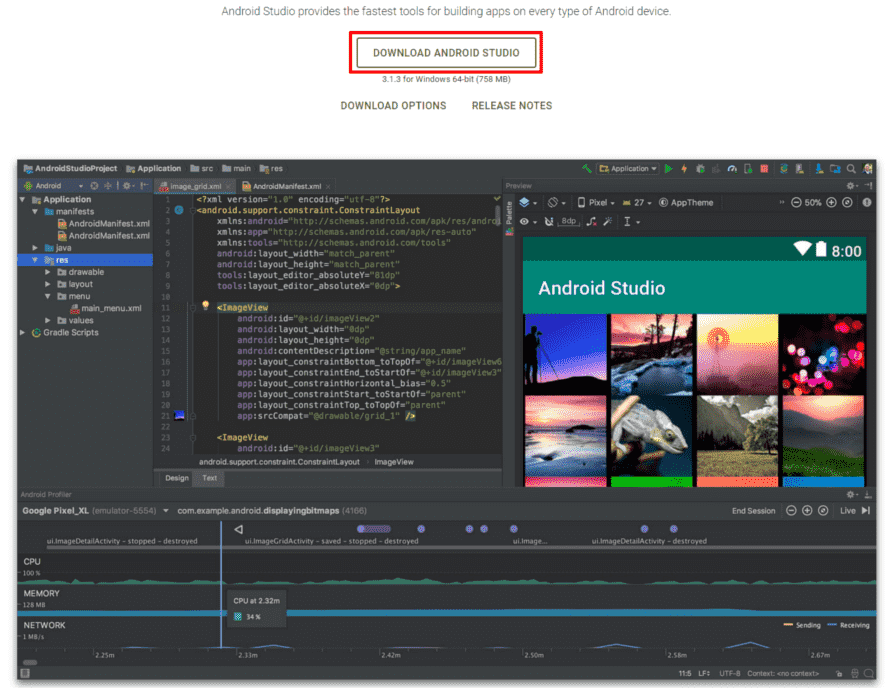
Antes de instalar Android Studio, debemos asegurarnos que tenemos instalada la JDK 6 o superior (no es suficiente tener instalado el JRE stand-alone). Si vamos a desarrollar en la versión Android 5.0 o superior, es necesario instalar la JDK 7. Para saber si tenemos instalada la JDF (y qué versión está instalada), abrimos una ventana del terminal y escribimos:

javac -version

Si la JDK no está instalada o la versión es inferior a la 6, debemos descargar la [**Java SE Development Kit 7**](https://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)e instalarla.

### **5.1.2 Descarga e instalación de Android Studio**

**Paso 1:** Descargar la versión más reciente de Android Studio [**desde la página web principal de desarrolladores de Android**](https://developer.android.com/sdk/index.html).



**Paso 2:** Una vez descargado el instalador correspondiente a nuestro sistema operativo (por defecto la propia web detectará nuestro sistema operativo y nos ofrecerá la mejor versión acorde a él) lo ejecutamos en nuestro equipo para comenzar con la instalación. Lo primero que veremos será el asistente de instalación de Android Studio.



**Paso 3:** El programa intentará detectar y configurar el entorno de desarrollo de Java, el JDK. Si no lo tenemos instalado nos aparecerá una ventana similar a la siguiente, desde donde podemos descargarlo.

**Paso 4:** Una vez descargado e instalado el JDK de Java podemos seguir con la instalación. En la próxima ventana del asistente, debemos elegir los paquetes que vamos a instalar:

* El editor de Android Studio, básico.
* El SDK de Android.
* El emulador de Android.

**Paso 5:** Lo más recomendable es instalar todos para tener todas las funciones disponibles en caso de que queramos utilizarlas. Seguimos con el asistente y llegaremos a la licencia y a los términos de uso, que debemos aceptar para poder seguir con la instalación.

**Paso 6:** En el siguiente paso debemos elegir la ruta donde instalaremos nuestro Android Studio. Debemos elegir una ruta para el programa en sí y otra diferente para instalar el SDK, con bastante espacio disponible ya que las descargas y actualizaciones de los componentes de este suelen ocupar bastante espacio.

**Paso 7:** Continuamos el asistente y antes de empezar el proceso nos preguntará si queremos un acceso directo en el inicio.

**Paso 8:** En el siguiente paso el asistente nos preguntará por la cantidad de memoria RAM que queremos asignar para el uso de máquinas virtuales y emuladores de Android. Cuanta mayor memoria mejor rendimiento tendrán estas, aunque debemos tener en cuenta que ni ningún smartphone cuenta con más de 2GB de memoria (de momento) y que el sistema operativo y las demás aplicaciones de nuestro ordenador también necesitan memoria, por lo que la cantidad que nos aparezca como recomendada será la que debemos dejar.

**Paso 9:** Con esto comenzará ya la copia de los archivos al disco duro. Este proceso puede tardar más o menos tiempo según la velocidad de nuestro sistema.

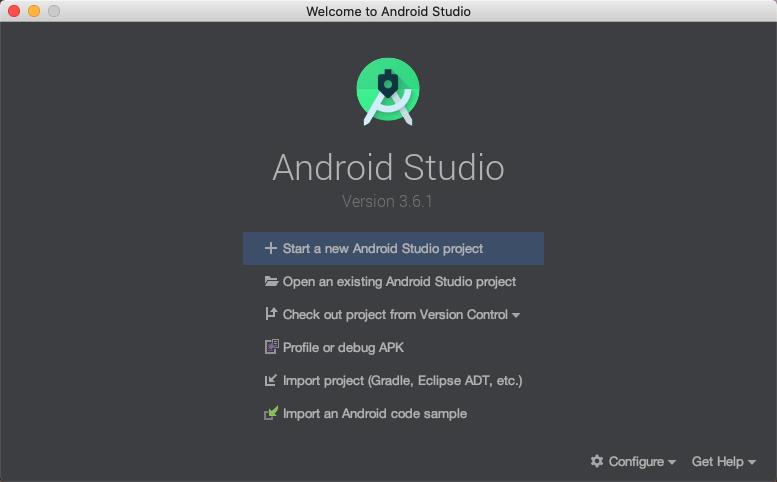
**Paso 10:** Una vez finalice la instalación Android Studio se conectará a Internet y descargará los elementos del SDK necesarios para funcionar correctamente.

Una vez finalice esta copia de datos ya estaremos listos para utilizar nuestro entorno de programación de Android.

### **5.1.3 Creación de un proyecto en Android Studio**

Para crear tu nuevo proyecto de Android, sigue estos pasos:

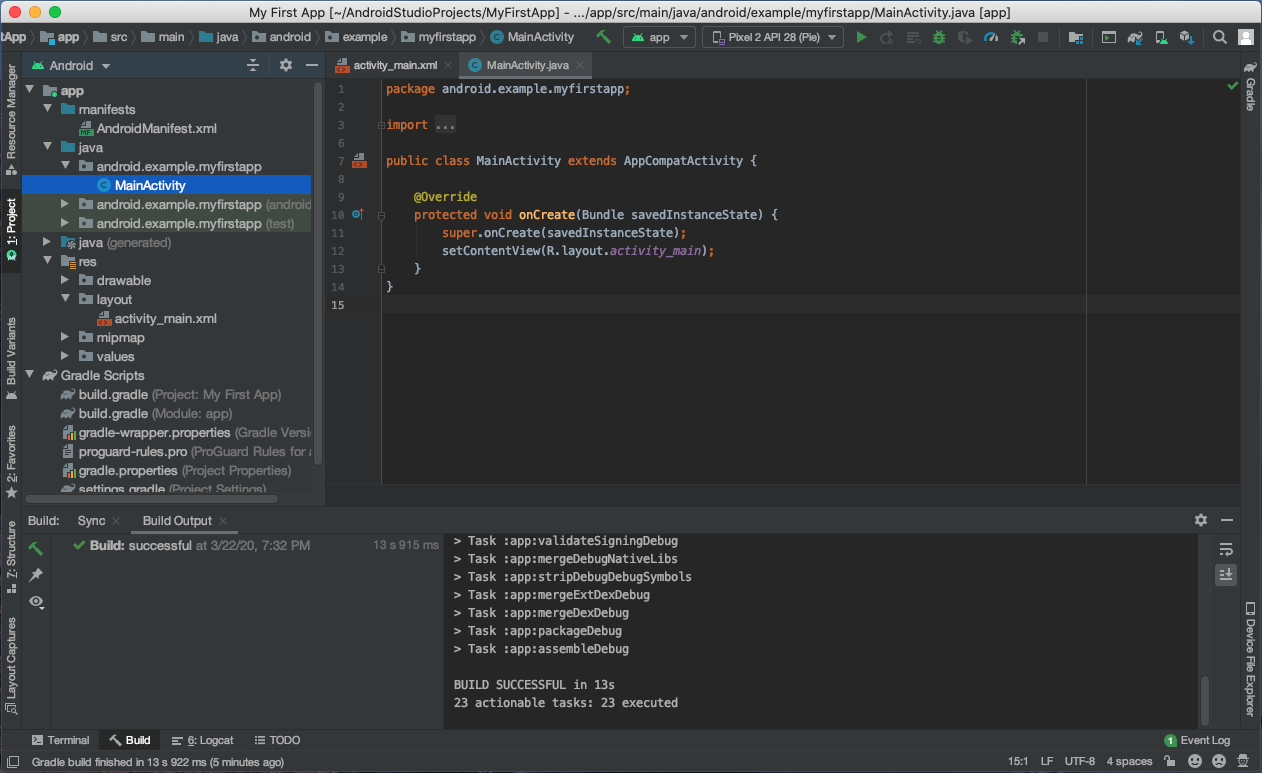
1. Instala la versión más reciente de [Android Studio](https://developer.android.com/studio/?hl=es).
2. En la ventana **Welcome to Android Studio**, haz clic en **Start a new Android Studio project**.



Si ya tienes un proyecto abierto, selecciona **File > New > New Project**.

1. En la ventana **Choose your project**, selecciona **Empty Activity** y haz clic en **Next**.
2. En la ventana **Configure your project**, completa lo siguiente:
   * Ingresa "Mi primera app" en el campo **Name**.
   * Ingresa "com.example.myfirstapp" en el campo **Package name**.
   * Si quieres colocar el proyecto en una carpeta diferente, cambia su ubicación en **Save**.
   * Selecciona **Java** o **Kotlin** en el menú desplegable **Language**.
   * Selecciona la casilla de verificación junto a **Use androidx.\* artifacts**.
   * Deja las otras opciones como están.
3. Haz clic en **Finish**.

Después de un tiempo de procesamiento, aparecerá la ventana principal de Android Studio.



## **5.2 Integración de Firebase con un proyecto de Android Studio**

### **5.2.1 Requisitos previos**

* Instala [Android Studio](http://developer.android.com/sdk?hl=es-419) o actualízalo a su versión más reciente.
* Asegúrate de que tu app para Android satisfaga estas condiciones:
  + Se orienta al nivel de API 16 (Jelly Bean) o superior.
  + Usa Gradle 4.1 o una versión más reciente.
* Configura un dispositivo o emulador para ejecutar la app.
  + Los [emuladores](https://developer.android.com/studio/run/managing-avds?hl=es-419) deben usar una imagen de emulador con Google Play.
* [Accede a Firebase](https://console.firebase.google.com/?hl=es-419) con tu Cuenta de Google.

### **5.2.2 Opción 1: Usa el flujo de trabajo de configuración de Firebase console (recomendada)**

Agregar Firebase a tu app implica realizar tareas en [Firebase console](https://console.firebase.google.com/?hl=es-419) y en tu proyecto abierto de Android (por ejemplo, descargar archivos de configuración de Firebase desde la consola y transferirlos al proyecto de Android).

**Paso 1**: Crea un proyecto de Firebase

A fin de poder agregar Firebase a tu app para Android, debes crear un proyecto de Firebase y conectarlo a la app. Consulta la [Información sobre los proyectos de Firebase](https://firebase.google.com/docs/projects/learn-more?hl=es-419) para obtener detalles acerca del tema.

**Crea un proyecto de Firebase**

**Paso 2**: Registra tu app con Firebase

Cuando tengas un proyecto de Firebase, podrás agregarle tu app para Android.

Revisa la [Información sobre los proyectos de Firebase](https://firebase.google.com/docs/projects/learn-more?hl=es-419#best-practices) a fin de obtener detalles sobre las recomendaciones y consideraciones para agregar apps a un proyecto de Firebase, incluido cómo manejar diversas variantes de compilación.

1. En el centro de la [página de descripción general del proyecto en Firebase console](https://console.firebase.google.com/?hl=es-419), haz clic en el ícono de **Android** para iniciar el flujo de trabajo de configuración.

Si ya agregaste una app a tu proyecto de Firebase, haz clic en **Agregar app** para que se muestren las opciones de plataforma.

1. Ingresa el [ID de aplicación](https://developer.android.com/studio/build/application-id?hl=es-419) de tu app en el campo **Nombre del paquete de Android**.

**Nota:** Asegúrate de ingresar el ID que usa tu app realmente. No puedes agregar ni modificar este valor después de registrar tu app en el proyecto de Firebase.

* + A veces, nos referimos al *ID de aplicación* como un *nombre de paquete*.
  + Encuentra este ID de aplicación en el archivo Gradle, generalmente app/build.gradle (ejemplo de ID: com.yourcompany.yourproject) de tu módulo (de nivel de la app).

1. *(Opcional)* Ingresa otra información de la app si el flujo de trabajo de configuración lo solicita.

El sobrenombre es un identificador interno y conveniente, y solo tú puedes verlo en Firebase console.

**Nota**: Firebase Authentication (cuando se usa el [Acceso con Google](https://developers.google.com/android/guides/client-auth?hl=es-419) o [con número de teléfono](https://firebase.google.com/docs/auth/android/google-signin?hl=es-419)) y [Firebase Dynamic Links](https://firebase.google.com/docs/auth/android/phone-auth?hl=es-419) requieren [información de SHA-1](https://firebase.google.com/docs/dynamic-links?hl=es-419).

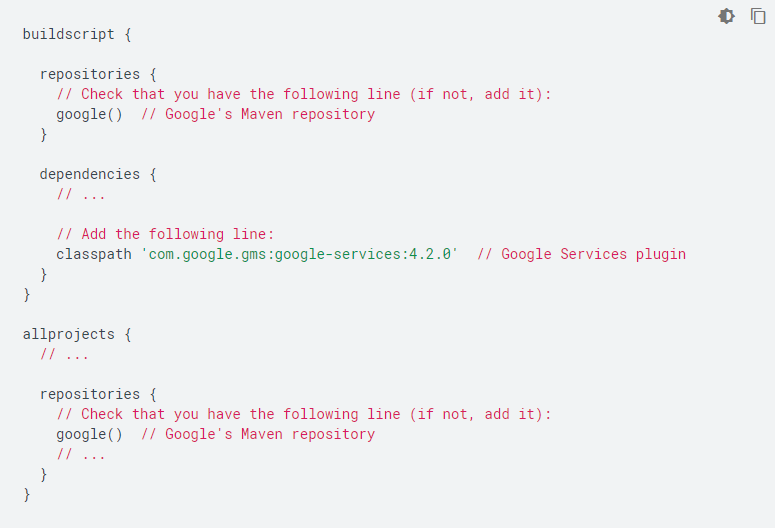
1. Haz clic en **Registrar app**.

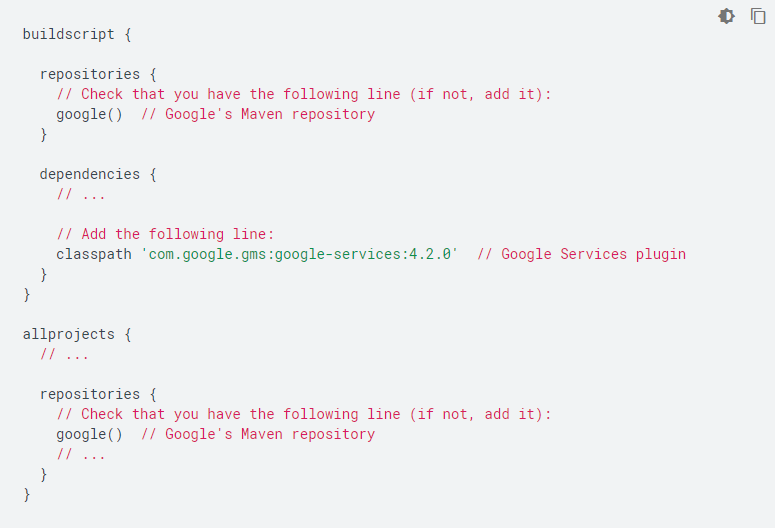
**Paso 3**: Agrega un archivo de configuración de Firebase

1. Agrega el archivo de configuración de Firebase para Android a la app, como se indica a continuación:
   1. Haz clic en **Descargar google-services.json** a fin de obtener el archivo de configuración de Firebase para Android (google-services.json).
      * Puedes volver a descargar el [archivo de configuración de Firebase para Android](http://support.google.com/firebase/answer/7015592?hl=es-419) cuando quieras.
      * Asegúrate de que no se hayan agregado caracteres adicionales, como (2), al nombre del archivo de configuración.
   2. Transfiere tu archivo de configuración al directorio del módulo (nivel de app) de tu app.

**Nota**: El archivo de configuración de Firebase contiene identificadores únicos, pero no secretos, para tu proyecto.

Consulta la [Información sobre los proyectos de Firebase](https://firebase.google.com/docs/projects/learn-more?hl=es-419#config-files-objects) para obtener más detalles sobre este archivo de configuración.

1. Agrega el [complemento de google-services](https://developers.google.com/android/guides/google-services-plugin?hl=es-419) a tus archivos Gradle a fin de habilitar los productos de Firebase en tu app.
   1. Agrega reglas para incluir el complemento de Google Services al archivo Gradle (build.gradle) de nivel de raíz (nivel de proyecto). Además, revisa que tengas el repositorio Maven de Google.



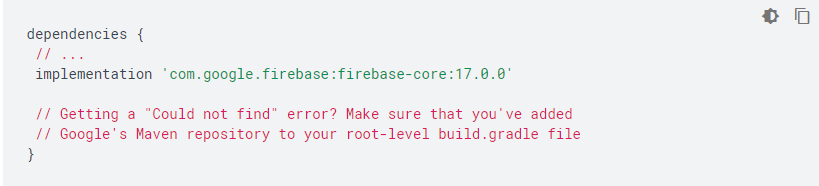
* 1. En el archivo Gradle (generalmente app/build.gradle) de tu módulo (nivel de app), agrega una línea *al final del archivo*.



**Paso 4**: Agrega los SDK de Firebase a tu app

Puedes agregar cualquiera de los [productos de Firebase admitidos](https://firebase.google.com/docs/android/setup?hl=es-419#available-libraries) a tu app para Android. Te recomendamos comenzar con el SDK principal de Firebase (com.google.firebase:firebase-core), que ofrece funciones de Google Analytics para Firebase.

1. En el archivo Gradle (generalmente app/build.gradle) de tu módulo (nivel de app), agrega la dependencia para el SDK principal de Firebase, como se muestra a continuación:



1. *(Opcional)* Agrega las dependencias de las otras [bibliotecas de Firebase](https://firebase.google.com/docs/android/setup?hl=es-419#available-libraries) que deseas usar.

Algunos SDK de Firebase para Android ofrecen una [biblioteca de extensiones de Kotlin](https://firebaseopensource.com/projects/firebase/firebase-android-sdk) alternativa.

1. Sincroniza tu app para garantizar que todas las dependencias tengan las versiones necesarias.
2. Ejecuta la app para enviar la verificación a Firebase de que lo integraste correctamente.

En los registros de tu dispositivo, se mostrará la verificación de Firebase que indica que se completó la inicialización. Si ejecutaste tu app en un emulador que tiene acceso a redes, [Firebase console](https://console.firebase.google.com/?hl=es-419) notificará que se completó la conexión de tu app.

### **5.2.3 Opción 2: Usa Firebase Assistant de Android Studio (se requiere configuración adicional)**

[Firebase Assistant](https://developer.android.com/studio/write/firebase.html?hl=es-419) registra tu app con un proyecto de Firebase y agrega los archivos y el código necesarios del servicio a tu proyecto de Android, todo desde Android Studio.

**Precaución**: Cuando uses Firebase Assistant, sigue cuidadosamente los pasos que se describen en esta sección de la guía para comprobar que todas las dependencias y los ajustes se hayan configurado correctamente.

Recuerda que para conectar tu app con Firebase, la opción que recomendamos es usar el [flujo de trabajo de configuración de Firebase console](https://firebase.google.com/docs/android/setup?hl=es-419#console).

1. Abre tu proyecto de Android en Android Studio.
2. Selecciona **Herramientas > Firebase** para abrir la ventana del **Asistente**.
3. Haz clic para expandir uno de los productos de Firebase que aparecen (por ejemplo, Analytics) y, luego, haz clic en el vínculo del instructivo que se proporciona (por ejemplo, Registrar un evento de Analytics).
4. Haz clic en **Conectar a Firebase** a fin de registrar tu app con un proyecto de Firebase nuevo o existente y para agregar automáticamente los archivos y el código necesarios al proyecto de Android.
5. Comprueba que las [versiones de tus complementos y bibliotecas](https://firebase.google.com/docs/android/setup?hl=es-419#available-libraries) sean las más recientes:
   * Comprueba que la versión de tu complemento de Google Services sea la más reciente (com.google.gms:google-services:4.2.0) en el archivo Gradle (build.gradle) de nivel de raíz (nivel de proyecto):
   * En el archivo Gradle (generalmente app/build.gradle) de tu módulo (nivel de app), comprueba que las [versiones de las bibliotecas de Firebase para Android](https://firebase.google.com/docs/android/setup?hl=es-419#available-libraries) sean las más recientes.

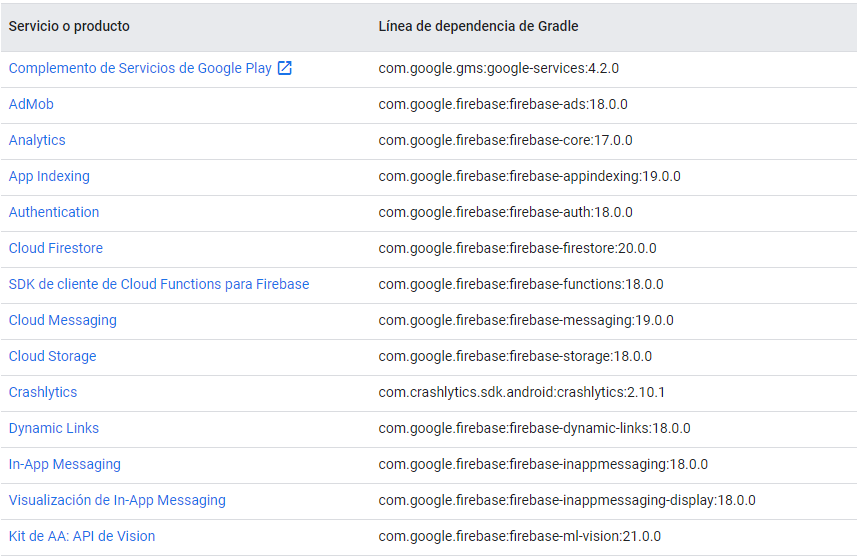
**Nota**: Si usas la versión 3.2 de Android Studio o una anterior, también debes asegurarte de que en cada línea de dependencia solo se haya especificado un número de versión.

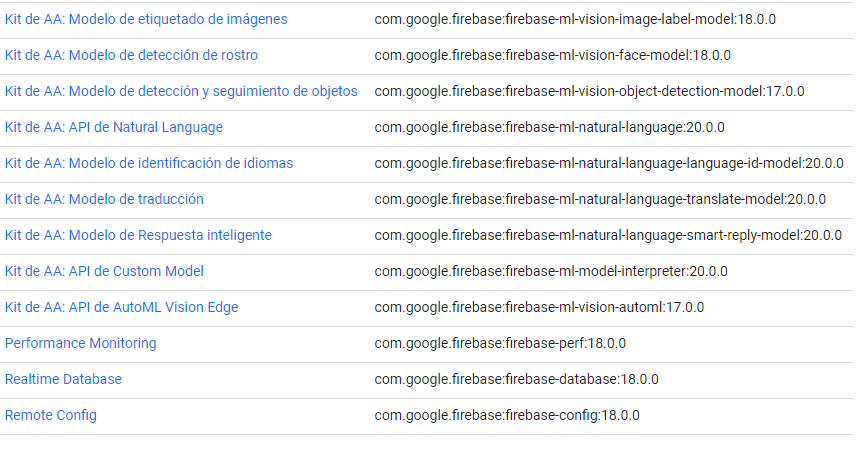
1. Sincroniza tu app para garantizar que todas las dependencias tengan las versiones necesarias.
2. Ve a la [**Configuración del proyecto**](https://console.firebase.google.com/project/_/settings/privacy/?hl=es-419) en Firebase console a fin de [establecer la configuración para compartir datos en Analytics](https://support.google.com/firebase/answer/6383877?hl=es-419).  
   A fin de usar productos como Firebase Predictions o Firebase A/B Testing se debe habilitar la función para compartir datos de Analytics con otros productos de Firebase.
3. Ejecuta la app para enviar la verificación a Firebase de que lo integraste correctamente.

En los registros de tu dispositivo, se mostrará la verificación de Firebase que indica que se completó la inicialización. Si ejecutaste tu app en un emulador que tiene acceso a redes, Firebase console notificará que se completó la conexión de tu app.

### **5.2.4 Bibliotecas disponibles**

Las siguientes [bibliotecas de Android](https://firebase.google.com/docs/reference/android/packages?hl=es-419) están disponibles para los distintos productos de Firebase.





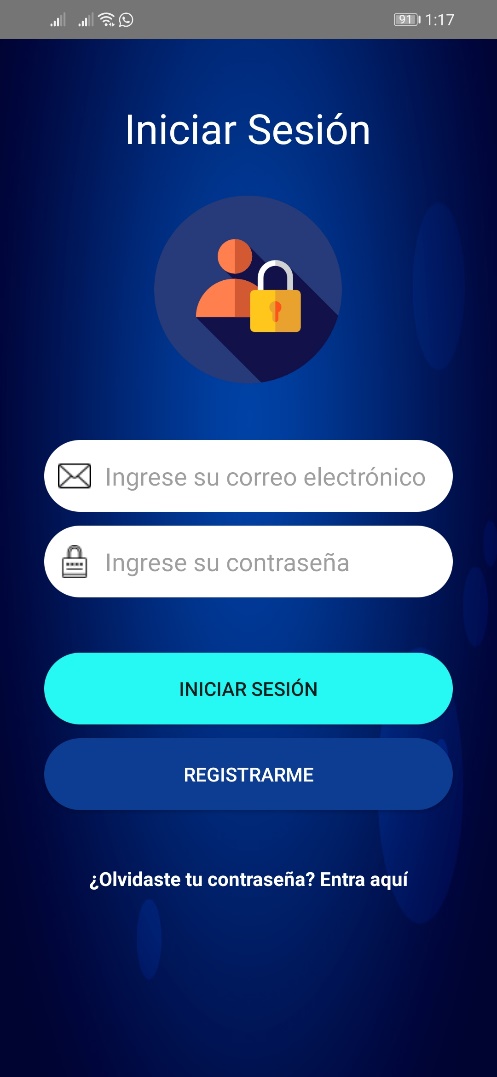
En nuestro caso ocupamos las librerías:

* 'com.google.firebase:firebase-auth:18.0.0'
* ‘com.google.firebase:firebase-database:18.0.0’

para utilizar las herramientas y funcionalidades que nos ofrece Firebase de Authentication y Realtime Database.

## **5.3 Contenido**

### **5.3.1 Inicio de sesión**

Esta es la primera ventana que aparecerá cuando se ejecute la aplicación móvil, donde el empleado del cine tendrá que iniciar sesión con un correo electrónico y una contraseña. En caso de no tener una cuenta hay una opción para registrarse y en caso de que el usuario haya olvidado su contraseña, hay una opción que le ayudará a recuperar su contraseña.

Parte del código asociado:

//Login de usuario  
 firebaseAuth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 //Actualizamos la contraseña del usuario en la base de datos en tiempo real en caso de que este la haya restablecido,  
 //ya que solo se verá reflejado el cambio en la pestaña de autenticación, pero no en la de la base de datos en tiempo real  
 actualizarPasswordUsuarios();  
  
 //Mensaje de que el usuario ha iniciado sesión exitosamente  
 Toast.*makeText*(MainActivity.this, "Bienvenido", Toast.*LENGTH\_SHORT*).show();  
  
 //Aquí estamos enviando el usuario a la otra actividad  
 Intent intencion = new Intent(getApplication(), Menuprincipal.class);  
  
 //Iniciamos la otra activity  
 startActivity(intencion);  
  
 //Finalizamos la activity actual, para que cuando el usuario presione el botón de atras ya no se visualize de nuevo  
 finish();  
  
  
 }else {  
  
 if (task.getException() instanceof FirebaseAuthInvalidUserException) {  
 //Mensaje de que el usuario no existe  
 Toast.*makeText*(MainActivity.this, "El usuario no esta registrado", Toast.*LENGTH\_SHORT*).show();  
  
 }else{  
 if (task.getException() instanceof FirebaseAuthInvalidCredentialsException) {  
 //Mensaje de que la contraseña en incorrecta  
 Toast.*makeText*(MainActivity.this, "Contraseña incorrecta", Toast.*LENGTH\_SHORT*).show();  
  
 }else{  
 //Mensaje de que no se pudo iniciar sesión por alguna otra razón  
 Toast.*makeText*(MainActivity.this, "No se pudo iniciar sesión", Toast.*LENGTH\_SHORT*).show();  
 }  
 }

### **5.3.2 Registro de usuario**

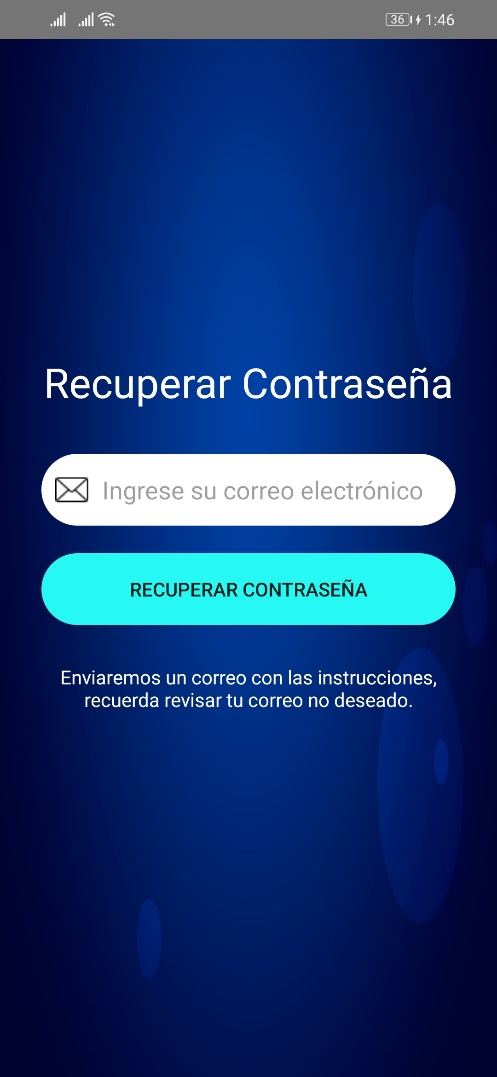
En esta ventana el empleado del cine podrá registrar una cuenta para poder iniciar sesión en la aplicación móvil, los datos que se requieren son los siguientes:

* Nombre completo del empleado.
* Nombre del cine donde trabaja.
* Puesto de trabajo del empleado.
* Correo electrónico.
* ****Contraseña que definirá el empleado.

Parte del código asociado:

//Creando nuevo usuario  
 firebaseAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 //checando si hay registro  
 if(task.isSuccessful()){  
 //Recibimos el id que tiene el usuario actual  
 String id = firebaseAuth.getCurrentUser().getUid();  
  
 //Creamos el conjunto de valores con los datos del usuario que vamos a registrar  
 Map<String, Object> map = new HashMap<>();  
 map.put("Nombre-Completo",nombre);  
 map.put("Nombre-Cine", cine);  
 map.put("Puesto-Trabajo", puesto);  
 map.put("Email", email);  
 map.put("Contrasenia", password);  
  
  
 //Creamos un nodo hijo llamado usuarios para almacenar en este los datos del usuario que se esta registrando  
 databaseReference.child("Usuarios").child(id).setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task2) {  
 if(task2.isSuccessful()){  
 //Mensaje de que el usuario ha sido registrado  
 Toast.*makeText*(Registrar\_Usuario.this, "Se ha registrado el usuario correctamente", Toast.*LENGTH\_SHORT*).show();  
  
 //Aquí estamos enviando el usuario a la otra actividad  
 Intent intencion = new Intent(getApplication(), MainActivity.class);  
  
 //Iniciamos la otra activity  
 startActivity(intencion);  
  
 //Finalizamos la activity actual, para que cuando el usuario presione el botón de atras ya no se visualize de nuevo  
 finish();  
  
 }

### **5.3.3 Recuperación de contraseña**

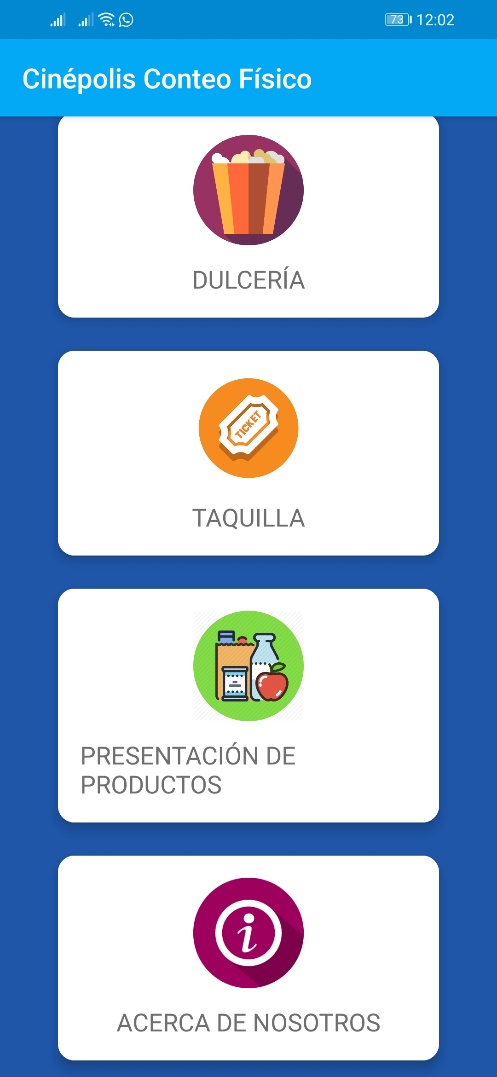
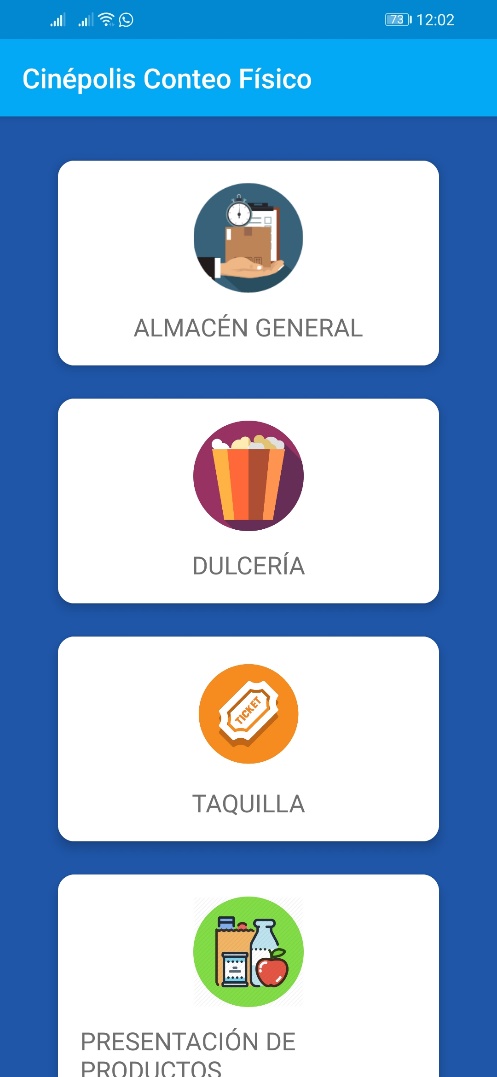
En esta ventana de la aplicación móvil, el empleado del cine podrá recuperar su contraseña en caso de que este mismo la haya olvidado, el único dato que se requiere es el de su correo electrónico con el que realizo el registro, al cual se le enviará un correo con las instrucciones que debe seguir para poder recuperar de manera exitosa su contraseña.

Parte del código asociado:

//Establecemos el idioma en el que se mandara el correo  
 firebaseAuth.setLanguageCode("es");  
  
 //Recuperar contraseña  
 firebaseAuth.sendPasswordResetEmail(email)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()){  
 //Mensaje de que el correo para restablecer la contraseña ha sido enviado  
 Toast.*makeText*(RecuperarContrasenia.this, "Se ha enviado un correo a "+email+" para restablecer tu contraseña", Toast.*LENGTH\_LONG*).show();  
  
 //Aquí estamos enviando el usuario a la otra actividad  
 Intent intencion = new Intent(getApplication(), MainActivity.class);  
  
 //Iniciamos la otra activity  
 startActivity(intencion);  
  
 //Finalizamos la activity actual, para que cuando el usuario presione el botón de atras ya no se visualize de nuevo  
 finish();  
 }else{  
 if (task.getException() instanceof FirebaseAuthInvalidUserException) {  
 //Mensaje de que el usuario con el correo proporcionado no existe  
 Toast.*makeText*(RecuperarContrasenia.this, "El correo electrónico que ingresaste no se encuentra registrado", Toast.*LENGTH\_SHORT*).show();  
 }else{  
 //Mensaje de que no se pudo restablecer la contraseña  
 Toast.*makeText*(RecuperarContrasenia.this, "No se pudo enviar el correo para restablecer contraseña", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 progressDialog.dismiss();  
 }  
 });  
  
}

### **5.3.4 Menú principal**

Una vez haya iniciado sesión el empleado del cine, lo mandará a esta ventana, donde se muestra un menú con cinco opciones:

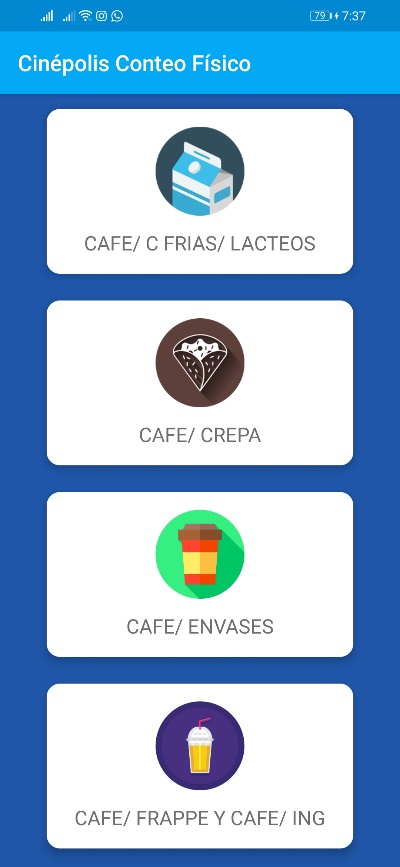
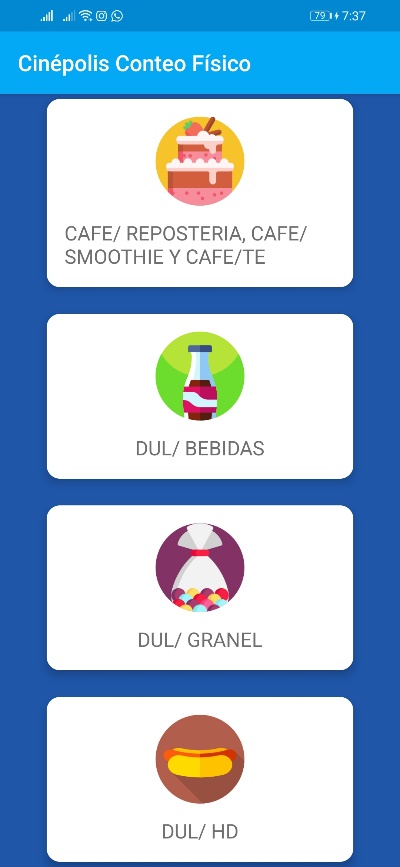
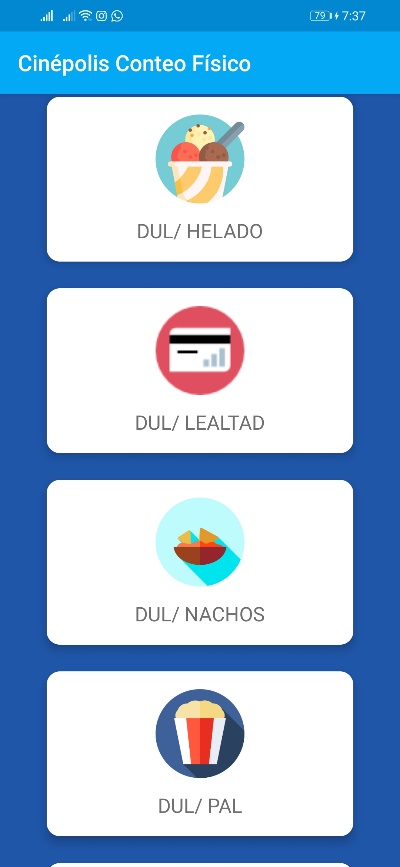
* Opción 1: Almacén General – Es donde el empleado podrá registrar el conteo físico de todos los productos que haya en el área de almacén general.
* Opción 2: Dulcería – Es donde el empleado podrá registrar el conteo físico de todos los productos que haya en el área de dulcería.
* Opción 3: Taquilla – Es donde el empleado podrá registrar el conteo físico de todos los productos que haya en el área de taquilla.
* Opción 4: Presentación de productos – Es donde el empleado podrá consultar información sobre las presentaciones en las que vienen los productos, en caso de que a este mismo se le haya olvidado. La presentación es la cantidad de producto que viene en una caja, paquete o bolsa. Por ejemplo, en una caja de nachos, vienen 25 bolsas de estos mismos, donde cada bolsa pesa 144 gramos, entonces la caja completa pesa 3600 gramos (144x25).
* Opción 5: Acerca de nosotros – Es donde el empleado podrá consultar información sobre las personas que desarrollaron la aplicación móvil (que en este caso seriamos: Erasmo y Kevin).

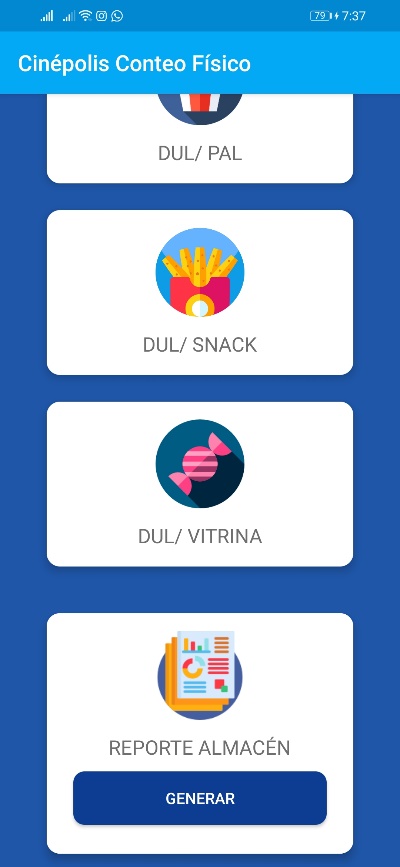
Parte del código asociado:

//Método para mandar a llamar el activity del menu de almacén  
 public void Iralmacen (View view){  
 Intent iralmacen = new Intent(this, Menu\_Almacen.class);  
 startActivity(iralmacen);  
 }  
  
 //Método para mandar a llamar el activity de dulceria  
 public void Irdulceria (View view){  
 Intent irdulceria = new Intent(this, Menu\_Dulceria.class);  
 startActivity(irdulceria);  
 }  
  
 //Método para mandar a llamar el activity de taquilla  
 public void Irtaquilla (View view){  
 Intent irtaquilla = new Intent(this, Taquilla.class);  
 startActivity(irtaquilla);  
 }  
  
 //Método para mandar a llamar el activity de presentaciones  
 public void Irpresentaciones (View view){  
 Intent irpresentaciones = new Intent(this, Presentaciones.class);  
 startActivity(irpresentaciones);  
 }  
  
  
 //Método para mandar a llamar el activity de acerca de nosotros  
 public void IrAcercadenosotros (View view){  
 Intent iracercadenosotros = new Intent(this, Acercadenosotros.class);  
 startActivity(iracercadenosotros);  
 }  
}

### **5.3.5 Menú almacén general**

En este menú podremos elegir entre una de las siguientes categorías que existen en el área de almacén general para realizar el conteo físico y finalmente nos da la opción para generar el reporte de dicha área:





Parte del código asociado:

//Método para mandar a llamar el activity de almacén 1

public void Iralmacen1 (View view){

Intent iralmacen1 = new Intent(this, Almacen1.class);

startActivity(iralmacen1);

}

//Método para mandar a llamar el activity de almacén 2

public void Iralmacen2 (View view){

Intent iralmacen2 = new Intent(this, Almacen2.class);

startActivity(iralmacen2);

}

//Método para mandar a llamar el activity de almacén 3

public void Iralmacen3 (View view){

Intent iralmacen3 = new Intent(this, Almacen3.class);

startActivity(iralmacen3);

}

//Método para mandar a llamar el activity de almacén 4

public void Iralmacen4 (View view){

Intent iralmacen4 = new Intent(this, Almacen4.class);

startActivity(iralmacen4);

}

//Método para mandar a llamar el activity de almacén 5

public void Iralmacen5 (View view){

Intent iralmacen5 = new Intent(this, Almacen5.class);

startActivity(iralmacen5);

}

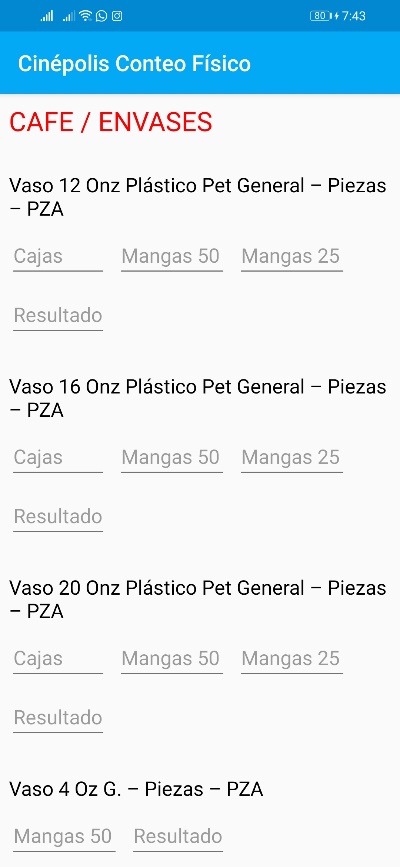
* CAFÉ/ C FRIAS/ LACTEOS



* CAFÉ/ CREPA



* CAFÉ/ ENVASES



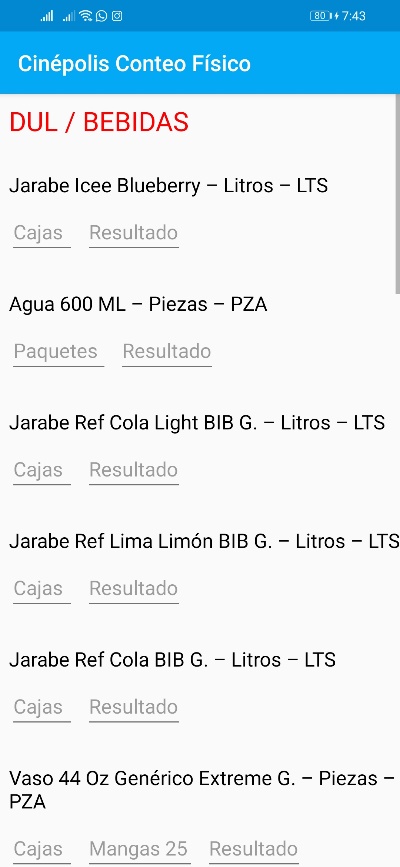
* CAFÉ/ FRAPPE Y CAFÉ/ ING



* CAFÉ/ REPOSTERÍA, CAFÉ/ SMOOTHIE Y CAFÉ/ TÉ



* DUL/ BEBIDAS



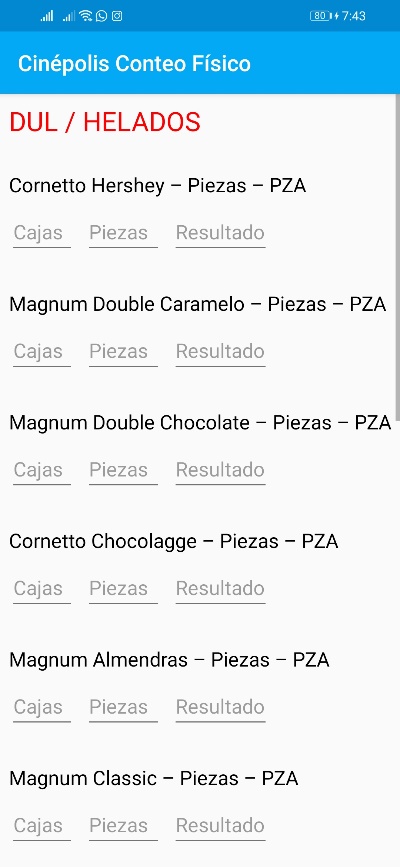
* DUL/ GRANEL



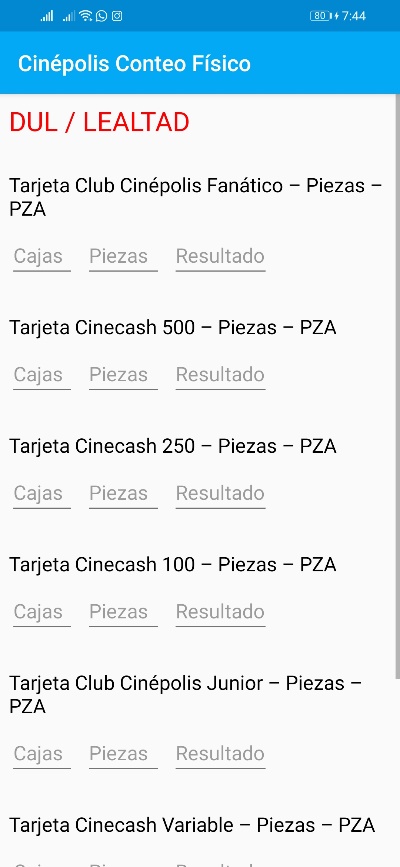
* DUL/ HD



* DUL/ HELADO



* DUL/ LEALTAD

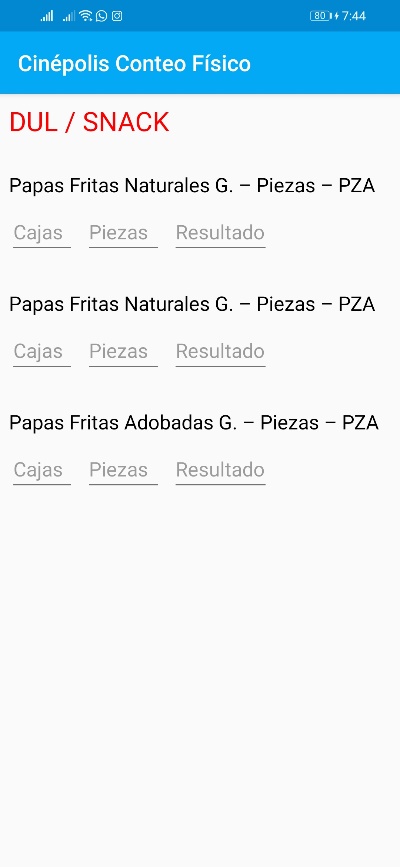


* DUL/ NACHOS



* DUL/ PAL

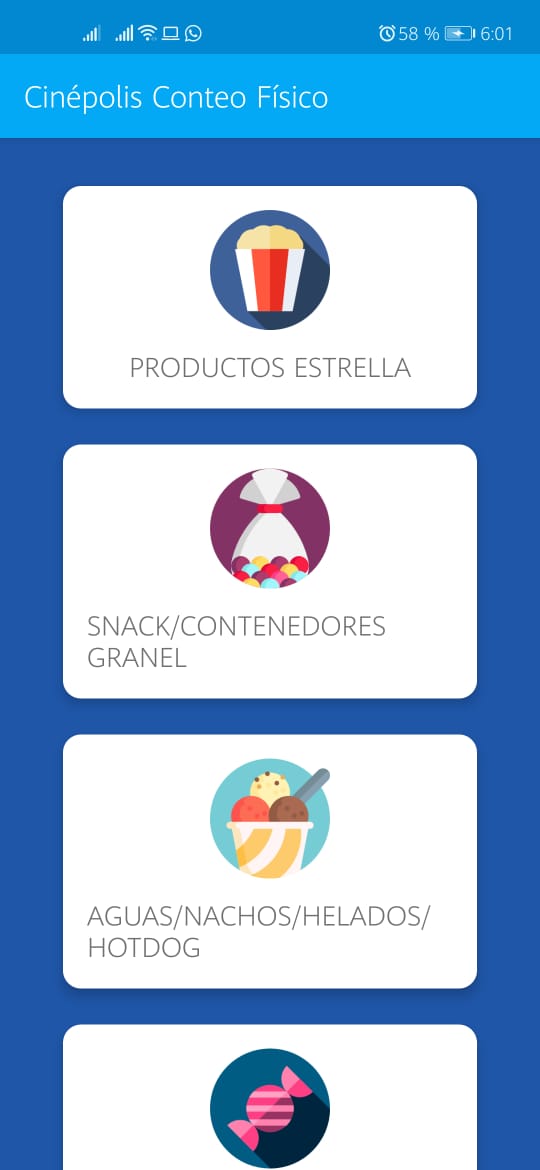
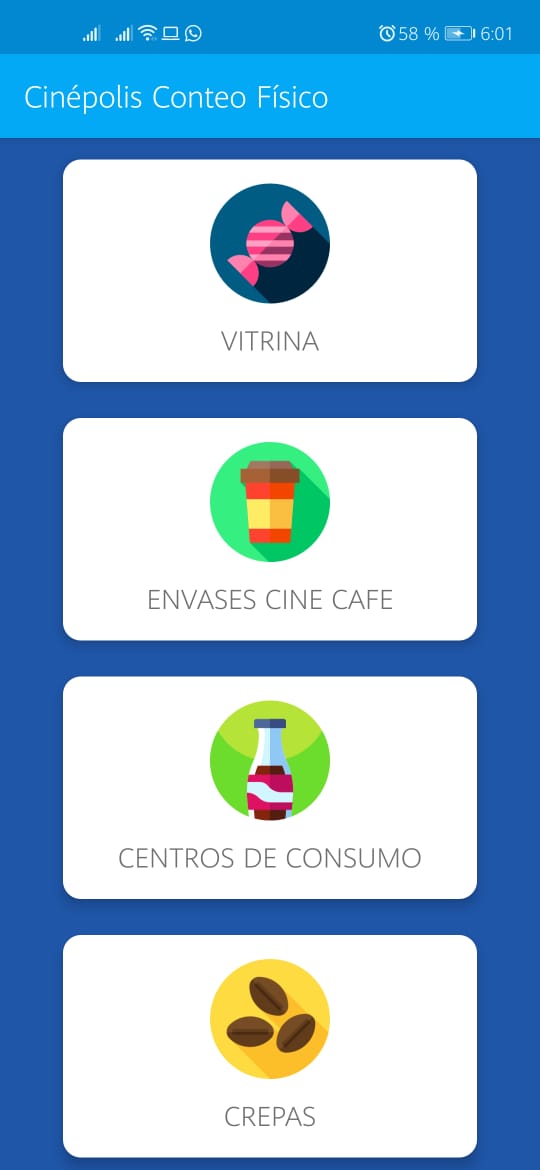
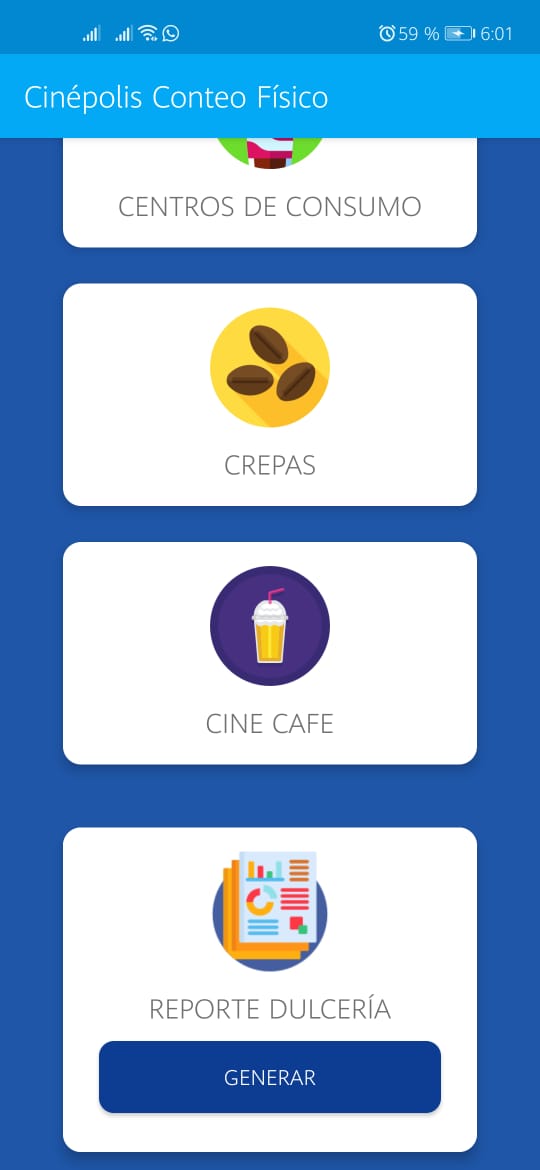


* DUL/ SNACK

* DUL/ VITRINA

* GENERAR EL REPORTE DEL ÁREA DE ALMACÉN GENERAL

### **5.3.6 Menú dulcería**

En este menú podremos elegir entre una de las siguientes categorías que existen en el área de dulcería para realizar el conteo físico y finalmente nos da la opción para generar el reporte de dicha área:

Parte del código asociado:

//Método para mandar a llamar el activity del menu de dulceria de productos estrella

public void Irdulceriaproductosestrella (View view){

Intent irdulceriaproductosestrella = new Intent(this, Dulceria\_Productos\_Estrella.class);

startActivity(irdulceriaproductosestrella);

}

//Método para mandar a llamar el activity del menu de dulceria de snack y contenedores

public void Irdulceriasnackcontenedores (View view){

Intent irdulceriasnackcontenedores = new Intent(this, Dulceria\_Snack\_Contenedores.class);

startActivity(irdulceriasnackcontenedores);

}

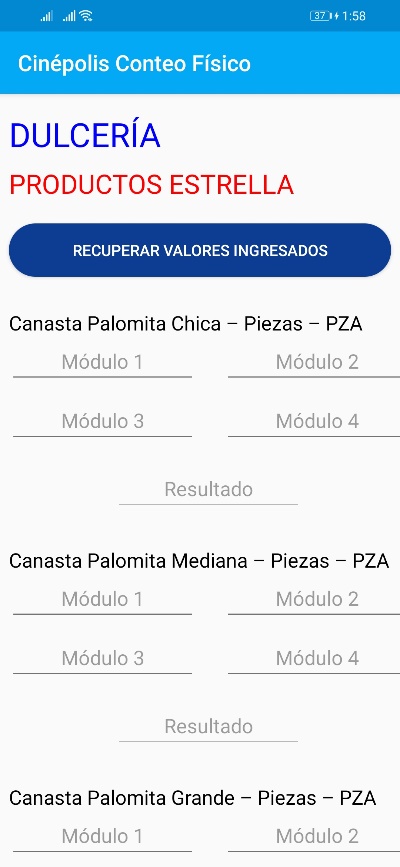
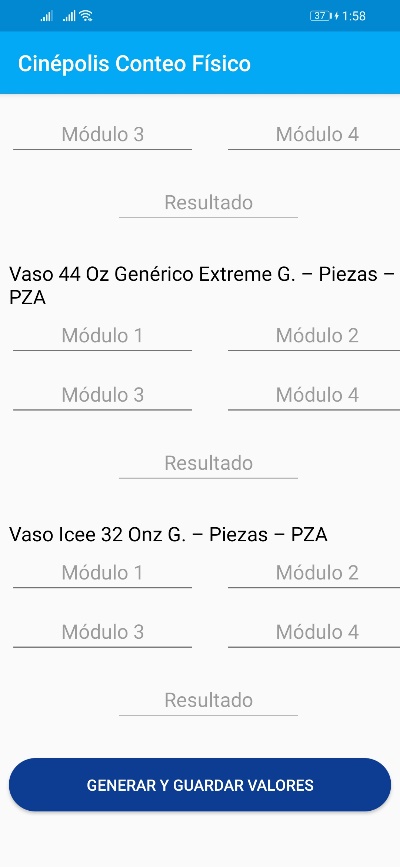
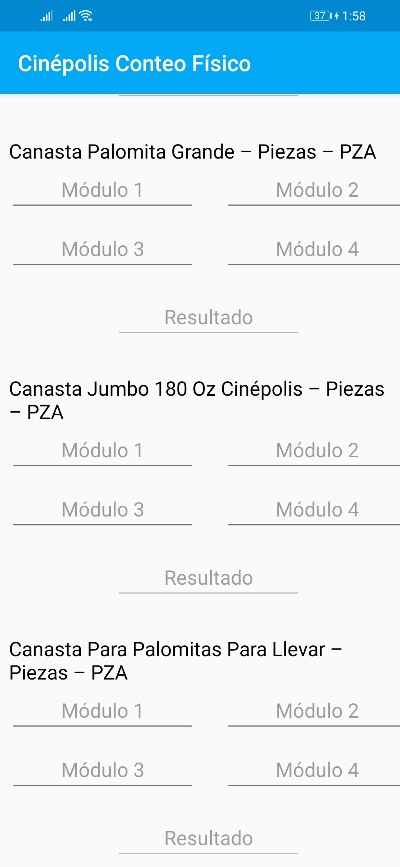
//Método para mandar a llamar el activity del menu de dulceria de aguas, nachos, helados y hotdog

public void Irdulceriasanhh (View view){

Intent irdulceriaanhh = new Intent(this, Dulceria\_Agua\_Nachos\_Helado\_Hotdog.class);

startActivity(irdulceriaanhh);

}

* PRODUCTOS ESTRELLA

Parte del código asociado:

//Realizamos las operaciones correspondientes

float existenciafisica = Float.parseFloat(modulo1)+Float.parseFloat(modulo2)+Float.parseFloat(modulo3)+Float.parseFloat(modulo4);

//Mandamos el resultado de la operación a una caja de texto

TextVasoiceefisico\_dul.setText(String.valueOf(existenciafisica));

//Mandamos a llamar la función actualizar datos

actualizardatos("DUL-BEBIDAS",431, existenciafisica);

//Mensaje de pantalla

Toast.makeText(this, "Datos Generados y Guardados Correctamente", Toast.LENGTH\_LONG).show();

private void actualizardatos(String categoria, int id\_producto, float existenciafisica){

//Recibimos el id que tiene el usuario actual

String id = firebaseAuth.getCurrentUser().getUid();

//Creamos el mapa de los valores que vamos a actualizar

Map<String, Object> map = new HashMap<>();

map.put("Id-Usuario", id);

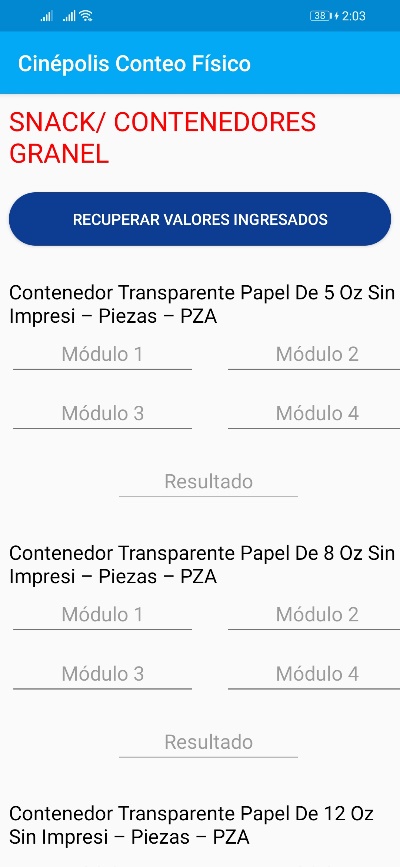
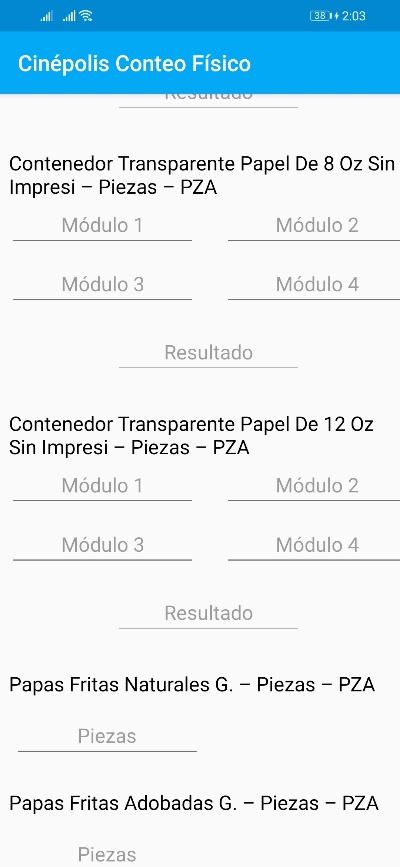
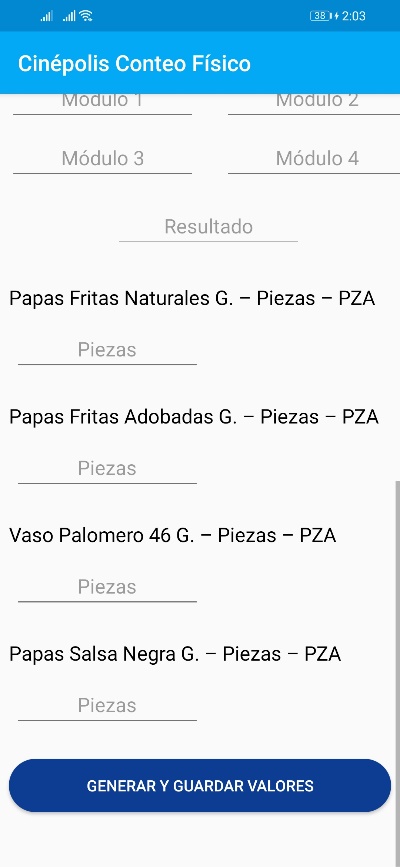
map.put("Existencia-Fisica",existenciafisica);

//Actualizamos los datos con esta función

databaseReference.child("Productos").child("Dulceria").child(categoria).child(""+id\_producto).updateChildren(map);

}

* SNACK/CONTENEDORES GRANEL



Parte del código asociado:

//Realizamos las operaciones correspondientes

float existenciafisica = Float.parseFloat(modulo1)+Float.parseFloat(modulo2)+Float.parseFloat(modulo3)+Float.parseFloat(modulo4);

//Mandamos el resultado de la operación a una caja de texto

Textcontenedor12onzfisico\_dul.setText(String.valueOf(existenciafisica));

//Mandamos a llamar la función actualizar datos

actualizardatos("DUL-GRANEL",11719, existenciafisica);

private void actualizardatos(String categoria, int id\_producto, float existenciafisica){

//Recibimos el id que tiene el usuario actual

String id = firebaseAuth.getCurrentUser().getUid();

//Creamos el mapa de los valores que vamos a actualizar

Map<String, Object> map = new HashMap<>();

map.put("Id-Usuario", id);

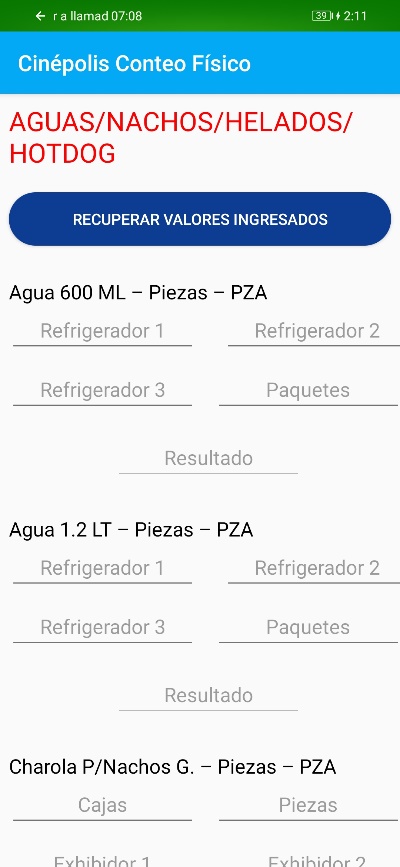
map.put("Existencia-Fisica",existenciafisica);

//Actualizamos los datos con esta función

databaseReference.child("Productos").child("Dulceria").child(categoria).child(""+id\_producto).updateChildren(map);

}

* AGUAS/NACHOS/HELADOS/HOT DOG



Parte del código asociado:

//Realizamos las operaciones correspondientes

float existenciafisica = Float.parseFloat(modulo1)+Float.parseFloat(modulo2)+Float.parseFloat(modulo3);

//Mandamos el resultado de la operación a una caja de texto

Textmichamordiskofisico\_dul.setText(String.valueOf(existenciafisica));

//Mandamos a llamar la función actualizar datos

actualizardatos("DUL-HELADO",9692, existenciafisica);

private void actualizardatos(String categoria, int id\_producto, float existenciafisica){

//Recibimos el id que tiene el usuario actual

String id = firebaseAuth.getCurrentUser().getUid();

//Creamos el mapa de los valores que vamos a actualizar

Map<String, Object> map = new HashMap<>();

map.put("Id-Usuario", id);

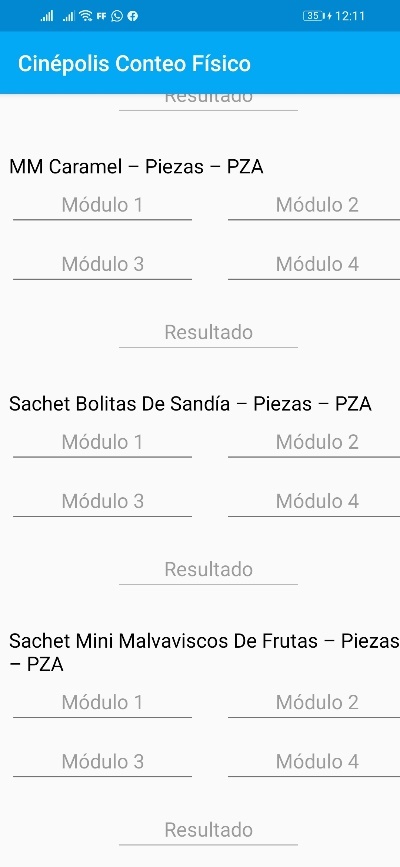
map.put("Existencia-Fisica",existenciafisica);

//Actualizamos los datos con esta función

databaseReference.child("Productos").child("Dulceria").child(categoria).child(""+id\_producto).updateChildren(map);

}

* VITRINA



Parte del código asociado:

//Realizamos las operaciones correspondientes

float existenciafisica = Float.parseFloat(modulo1)+Float.parseFloat(modulo2)+Float.parseFloat(modulo3)+Float.parseFloat(modulo4);

//Mandamos el resultado de la operación a una caja de texto

Textkitkat4fisico\_dul.setText(String.valueOf(existenciafisica));

//Mandamos a llamar la función actualizar datos

actualizardatos("DUL-VITRINA",11345, existenciafisica);

private void actualizardatos(String categoria, int id\_producto, float existenciafisica){

//Recibimos el id que tiene el usuario actual

String id = firebaseAuth.getCurrentUser().getUid();

//Creamos el mapa de los valores que vamos a actualizar

Map<String, Object> map = new HashMap<>();

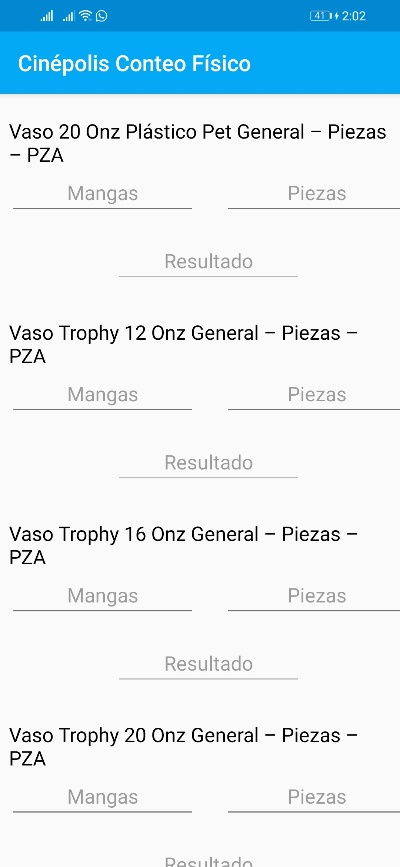
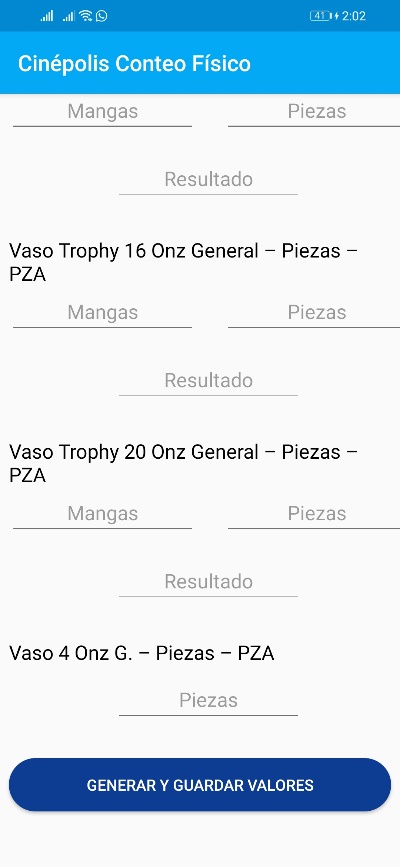
map.put("Id-Usuario", id);

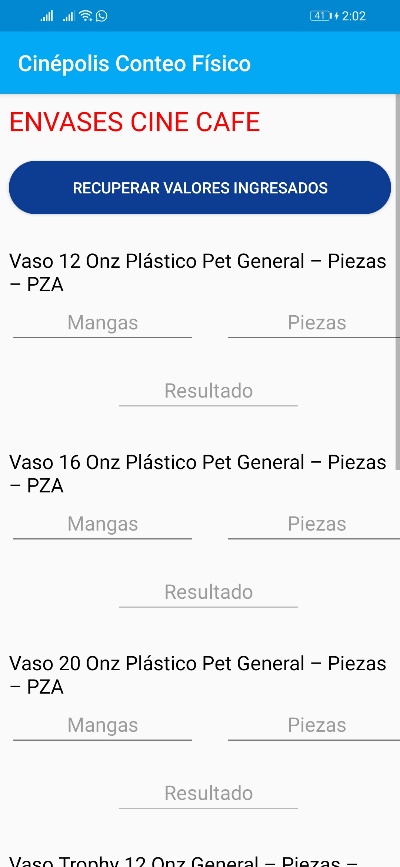
map.put("Existencia-Fisica",existenciafisica);

//Actualizamos los datos con esta función

databaseReference.child("Productos").child("Dulceria").child(categoria).child(""+id\_producto).updateChildren(map);

}

* ENVASES CINE CAFÉ



Parte del código asociado:

//Realizamos las operaciones correspondientes

float existenciafisica = (Float.parseFloat(modulo1)\*25)+Float.parseFloat(modulo2);

//Mandamos el resultado de la operación a una caja de texto

Textvaso12onzfisico\_dul.setText(String.valueOf(existenciafisica));

//Mandamos a llamar la función actualizar datos

actualizardatos("CAFE-ENVASES",522, existenciafisica);

private void actualizardatos(String categoria, int id\_producto, float existenciafisica){

//Recibimos el id que tiene el usuario actual

String id = firebaseAuth.getCurrentUser().getUid();

//Creamos el mapa de los valores que vamos a actualizar

Map<String, Object> map = new HashMap<>();

map.put("Id-Usuario", id);

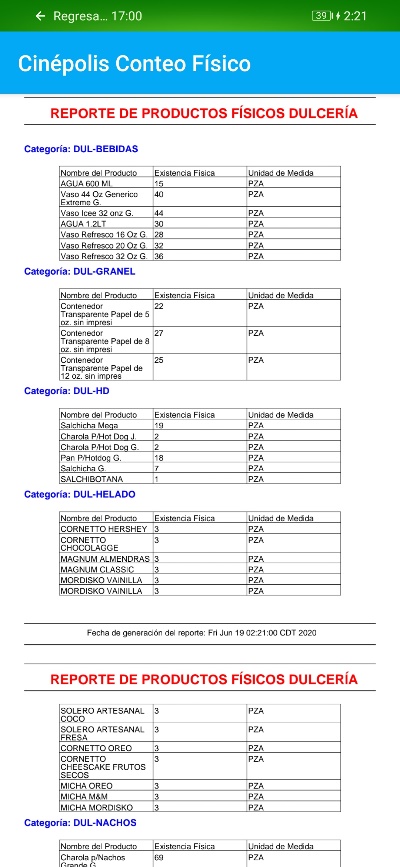
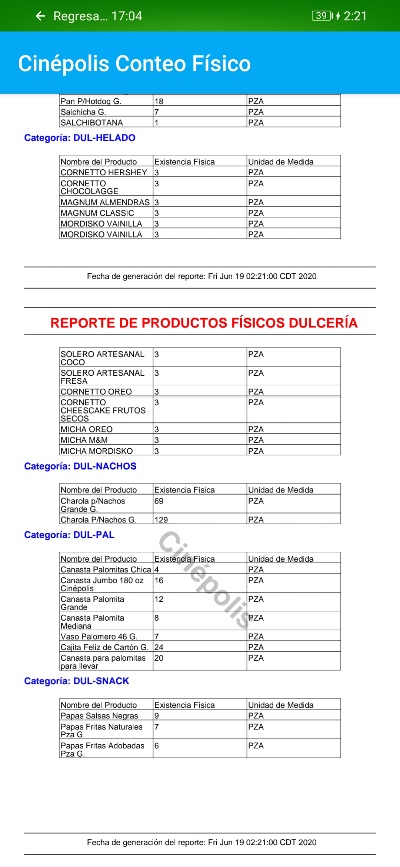
map.put("Existencia-Fisica",existenciafisica);

//Actualizamos los datos con esta función

databaseReference.child("Productos").child("Dulceria").child(categoria).child(""+id\_producto).updateChildren(map);

}

* CENTROS DE CONSUMO
* CREPAS
* CINE CAFÉ
* GENERAR EL REPORTE DEL ÁREA DE DULCERÍA



Parte del código asociado:

private void crearPDF() {

// Creamos el documento.

Document documento = new Document();

try {

//Tomo la fecha y la hora del sistema (Telefono, Tablet, SmartWatch, etc)

DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = new Date();

// Creamos el fichero con el nombre que deseemos.

archivoPDF = crearFichero(NOMBRE\_DOCUMENTO);

// Creamos el flujo de datos de salida para el fichero donde

// guardaremos el pdf.

FileOutputStream ficheroPdf = new FileOutputStream(

archivoPDF.getAbsolutePath());

// Asociamos el flujo que acabamos de crear al documento.

PdfWriter writer = PdfWriter.getInstance(documento, ficheroPdf);

// Añadimos un título con una fuente personalizada.

Font font1 = FontFactory.getFont(FontFactory.HELVETICA, 20,

Font.BOLD, Color.RED);

// Incluimos el píe de página y una cabecera

HeaderFooter cabecera = new HeaderFooter(new Phrase(

" REPORTE DE PRODUCTOS FÍSICOS DULCERÍA",font1), false);

HeaderFooter pie = new HeaderFooter(new Phrase(

" Fecha de generación del reporte: " + date), false);

documento.setHeader(cabecera);

documento.setFooter(pie);

// Abrimos el documento.

documento.open();

// Añadimos un título con la fuente por defecto.

//documento.add(new Paragraph("Título 1"));

// Añadimos un título con una fuente personalizada.

Font font2 = FontFactory.getFont(FontFactory.HELVETICA, 14,

Font.BOLD, Color.BLUE);

documento.add(new Paragraph("Categoría: DUL-BEBIDAS \n\n", font2));

// Insertamos una tabla

PdfPTable tabla1 = new PdfPTable(3);

documento.add(tabla1);

// Agregar marca de agua

font2 = FontFactory.getFont(FontFactory.HELVETICA, 42, Font.BOLD,

Color.lightGray);

ColumnText.showTextAligned(writer.getDirectContentUnder(),

Element.ALIGN\_CENTER, new Paragraph(

"Cinépolis", font2), 297.5f, 421,

writer.getPageNumber() % 2 == 1 ? 45 : -45);

Toast.makeText(contexto,"PDF creado correctamente",Toast.LENGTH\_LONG).show();

} catch (DocumentException e) {

Log.e(ETIQUETA\_ERROR, e.getMessage());

} catch (IOException e) {

Log.e(ETIQUETA\_ERROR, e.getMessage());

} finally {

// Cerramos el documento.

documento.close();

}

}

private void verPDF() {

//Primero busca el archivo

buscarPDF();

if(fPDF==1) {

//De que actividad estoy a cual quiero ir

Intent intent = new Intent(contexto, PDF\_Taquilla.class);

//Envío de información entre actividades

intent.putExtra("path", archivoPDF.getAbsolutePath());

intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

contexto.startActivity(intent);

Toast.makeText(contexto,"Si existe archivo PDF para LEER",Toast.LENGTH\_LONG).show();

}

else

if(fPDF==0){

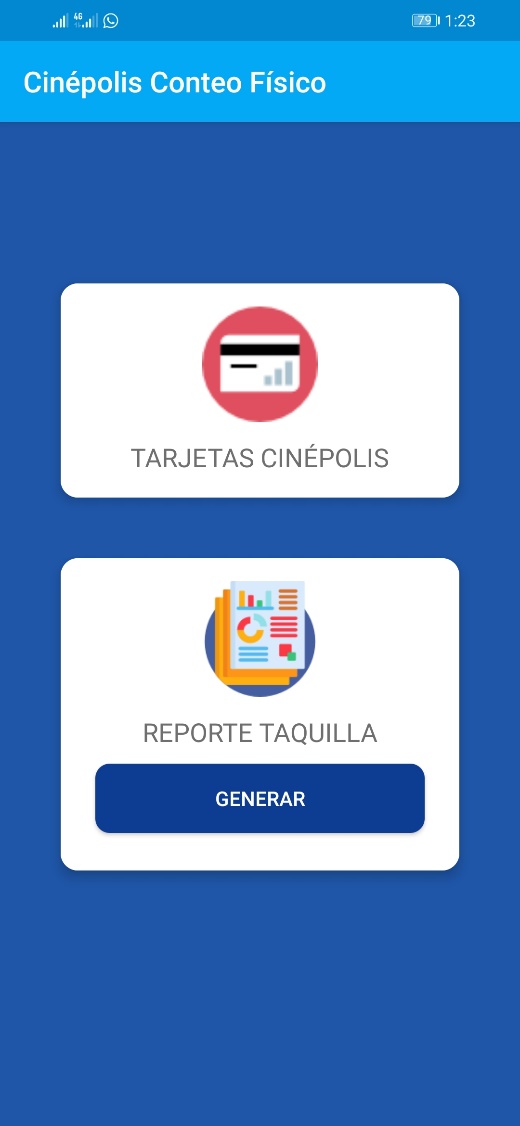
Toast.makeText(contexto,"No existe archivo PDF para LEER",Toast.LENGTH\_LONG).show();

}

}

### **5.3.7 Menú taquilla**

En este menú podremos elegir entre una de las siguientes categorías que existen en el área de taquilla para realizar el conteo físico y finalmente nos da la opción para generar el reporte de dicha área:



Parte del código asociado:

//Método para mandar a llamar el activity de taquilla

public void Irtaquilla (View view){

Intent irtaquilla = new Intent(this, Taquilla.class);

startActivity(irtaquilla);

}

public void onClick(View v) {

switch (v.getId()){

case R.id.generarReporteTaquilla:

//Invocamos al método

crearPDF();

verPDF();

break;

}

}

* TARJETAS CINÉPOLIS

Parte del código asociado:

//Realizamos las operaciones correspondientes

float existenciafisica = (Float.parseFloat(cajas)\*100)+

Float.parseFloat(piezas);

//Mandamos el resultado de la operación a una caja de texto

TextresultadoTCCSFTaquilla.setText(String.valueOf(existenciafisica));

//Mandamos a llamar la función actualizar datos

actualizardatos("DUL-LEALTAD",3802, existenciafisica);

//Mensaje de pantalla

Toast.makeText(this, "Datos Generados y Guardados Correctamente", Toast.LENGTH\_LONG).show();

private void actualizardatos(String categoria, int id\_producto, float existenciafisica){

//Recibimos el id que tiene el usuario actual

String id = firebaseAuth.getCurrentUser().getUid();

//Creamos el mapa de los valores que vamos a actualizar

Map<String, Object> map = new HashMap<>();

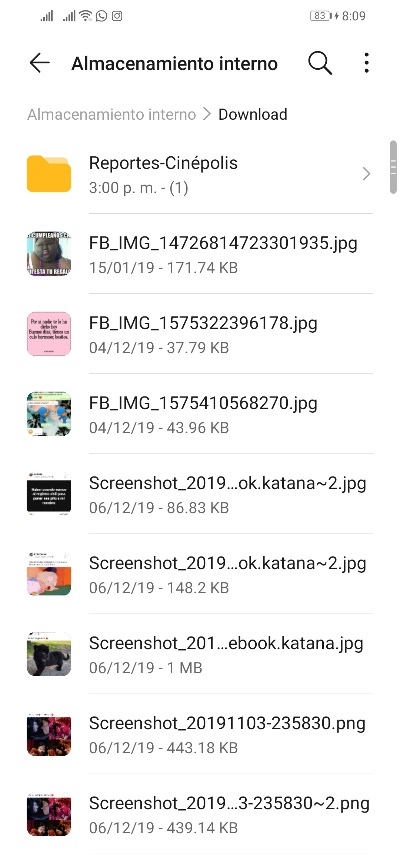
map.put("Id-Usuario", id);

map.put("Existencia-Fisica",existenciafisica);

//Actualizamos los datos con esta función databaseReference.child("Productos").child("Taquilla").child(categoria).child(""+id\_producto).updateChildren(map);

}

* GENERAR EL REPORTE DEL ÁREA DE TAQUILLA





Parte del código asociado:

private void crearPDF() {

// Creamos el documento.

Document documento = new Document();

try {

//Tomo la fecha y la hora del sistema (Telefono, Tablet, SmartWatch, etc)

DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = new Date();

// Creamos el fichero con el nombre que deseemos.

archivoPDF = crearFichero(NOMBRE\_DOCUMENTO);

// Creamos el flujo de datos de salida para el fichero donde

// guardaremos el pdf.

FileOutputStream ficheroPdf = new FileOutputStream(

archivoPDF.getAbsolutePath());

// Asociamos el flujo que acabamos de crear al documento.

PdfWriter writer = PdfWriter.getInstance(documento, ficheroPdf);

// Añadimos un título con una fuente personalizada.

Font font1 = FontFactory.getFont(FontFactory.HELVETICA, 20,

Font.BOLD, Color.RED);

// Incluimos el píe de página y una cabecera

HeaderFooter cabecera = new HeaderFooter(new Phrase(

" REPORTE DE PRODUCTOS FÍSICOS TAQUILLA",font1), false);

HeaderFooter pie = new HeaderFooter(new Phrase(

" Fecha de generación del reporte: " + date), false);

documento.setHeader(cabecera);

documento.setFooter(pie);

// Abrimos el documento.

documento.open();

// Añadimos un título con la fuente por defecto.

//documento.add(new Paragraph("Título 1"));

// Añadimos un título con una fuente personalizada.

Font font2 = FontFactory.getFont(FontFactory.HELVETICA, 14,

Font.BOLD, Color.BLUE);

documento.add(new Paragraph("Categoría: DUL-LEALTAD \n\n", font2));

// Insertamos una tabla

PdfPTable tabla = new PdfPTable(3);

documento.add(tabla);

// Agregar marca de agua

font2 = FontFactory.getFont(FontFactory.HELVETICA, 42, Font.BOLD,

Color.lightGray);

ColumnText.showTextAligned(writer.getDirectContentUnder(),

Element.ALIGN\_CENTER, new Paragraph(

"Cinépolis", font2), 297.5f, 421,

writer.getPageNumber() % 2 == 1 ? 45 : -45);

Toast.makeText(contexto,"PDF creado correctamente",Toast.LENGTH\_LONG).show();

} catch (DocumentException e) {

Log.e(ETIQUETA\_ERROR, e.getMessage());

} catch (IOException e) {

Log.e(ETIQUETA\_ERROR, e.getMessage());

} finally {

// Cerramos el documento.

documento.close();

}

}

private void verPDF() {

//Primero busca el archivo

buscarPDF();

if(fPDF==1) {

//De que actividad estoy a cual quiero ir

Intent intent = new Intent(contexto, PDF\_Taquilla.class);

//Envío de información entre actividades

intent.putExtra("path", archivoPDF.getAbsolutePath());

intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

contexto.startActivity(intent);

Toast.makeText(contexto,"Si existe archivo PDF para LEER",Toast.LENGTH\_LONG).show();

}

else

if(fPDF==0){

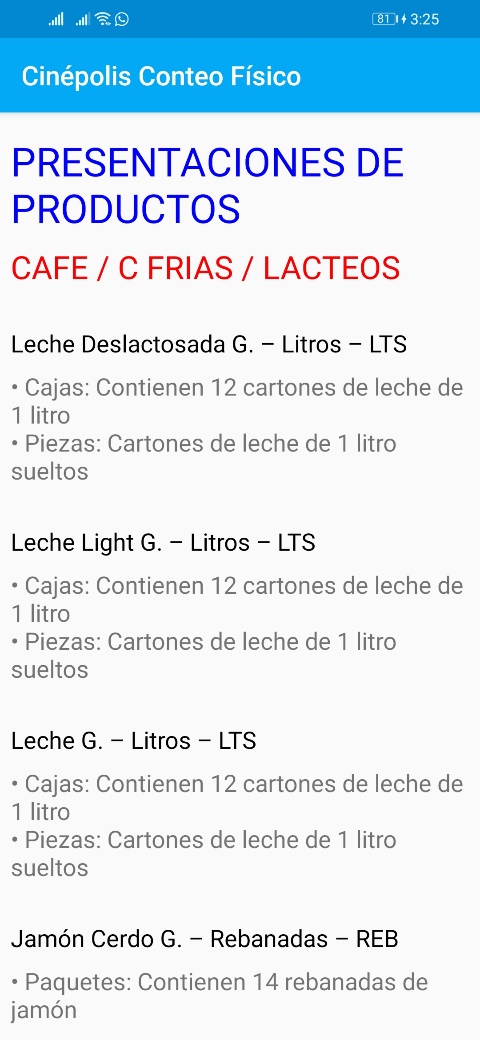
Toast.makeText(contexto,"No existe archivo PDF para LEER",Toast.LENGTH\_LONG).show();

}

}

### **5.3.8 Presentación de productos**

En esta opción del menú principal los empleados podrán visualizar las presentaciones que tienen los productos, en caso de que a estos se les olvide dicha información:



Parte del código asociado:

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:text="CAFE / C FRIAS / LACTEOS"

android:textColor="#FF0000"

android:textSize="24sp" />

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:text="\nLeche Deslactosada G. – Litros – LTS"

android:textColor="#000000"

android:textSize="18sp" />

<TextView

android:id="@+id/textView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:text="• Cajas: Contienen 12 cartones de leche de 1 litro

\n• Piezas: Cartones de leche de 1 litro sueltos"

android:textSize="18sp" />

<TextView

android:id="@+id/textView5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

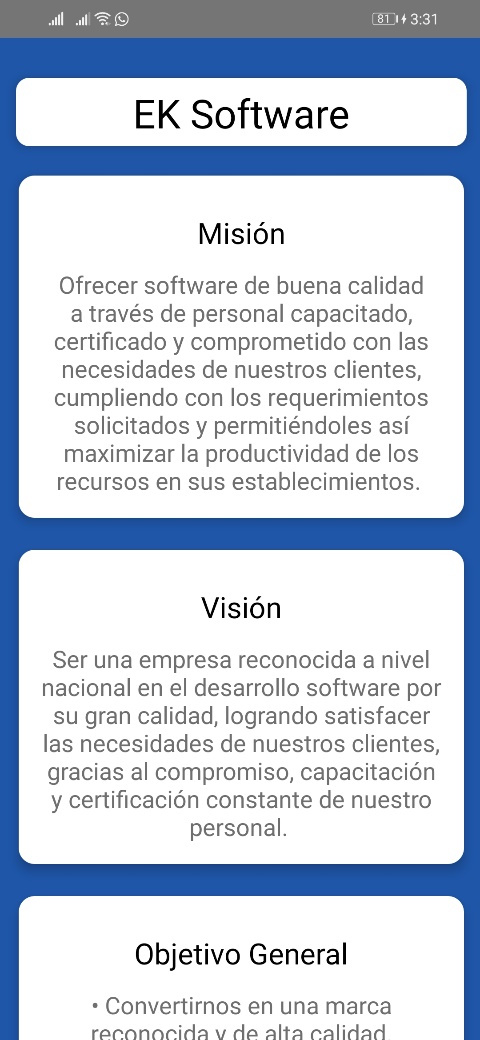
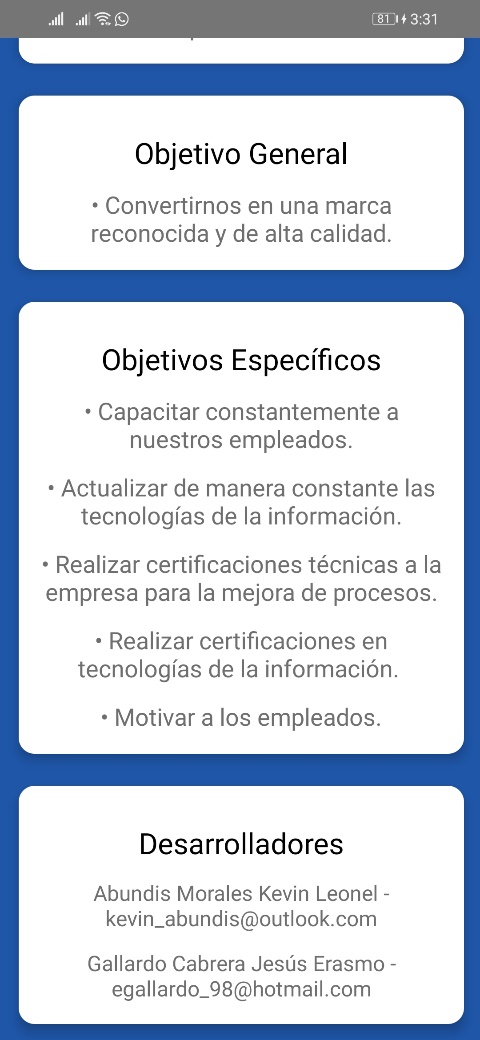
android:text="\nLeche Light G. – Litros – LTS"

android:textColor="#000000"

android:textSize="18sp" />

### **5.3.9 Acerca de nosotros**

En esta opción del menú principal los empleados podrán visualizar información sobre la empresa, los nombres y correos de contacto de los desarrolladores que realizaron la aplicación:



Parte del código asociado:

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

app:cardElevation="6dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="12dp"

android:text="Desarrolladores"

android:textColor="#000000"

android:textSize="22sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Abundis Morales Kevin Leonel - kevin\_abundis@outlook.com"

android:textColor="#6f6f6f"

android:textSize="16sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Gallardo Cabrera Jesús Erasmo - egallardo\_98@hotmail.com"

android:textColor="#6f6f6f"

android:textSize="16sp" />

</LinearLayout>

</androidx.cardview.widget.CardView>

# **Conclusión**

Habiendo realizado esté proyecto podemos concluir que el proceso de toma física de inventario de Cinépolis La Isla Acapulco, es un proceso complicado y tedioso, que necesitaba ser digitalizado para poder agilizar el proceso y ahorrar tiempos en su realización, sin embargo, este proceso se vio complicado dada la cantidad de productos que tienen y las diferentes presentaciones que puede tener un mismo producto, por ejemplo, puede haber queso crema philadelphia en presentaciones de una barra completa que pesa 1900 gramos, una porción que pesa 50 gramos o en bolsas de 10 porciones que pesan 500 gramos y cada una debe ser contabilizada y otro ejemplo que podemos mencionar son los diferentes tipos de palomitas que hay, pueden ser naturales, enchiladas, queso, entre otras, todo esto puede dar pie a errores cometidos por los encargados de tomar el inventario, si bien el error humano sigue estando presente a la hora de tomar el inventario, este se ve reducido al haber un aplicación móvil de por medio que tiene los productos ya registrados y que únicamente tienen que ser tomados de un menú y la aplicación móvil se encarga de realizar las operaciones correspondientes, las cuales anteriormente se hacían completamente a mano.

Para concluir, este proyecto, si bien ha resultado ser complicado, gracias al apoyo del asesor y una constante comunicación con el personal de Cinépolis La Isla Acapulco ha podido salir adelante y se espera que sea terminado con éxito.

# **Bibliografía**

ATOX SISTEMAS DE ALMACENAJE. (02 de Octubre de 2017). *Clasificación de inventarios ABC*. Recuperado el 05 de Noviembre de 2019, de http://www.atoxgrupo.com/website/noticias/clasificacion-inventarios-abc

EGASA. (2019). *PROCEDIMIENTO DE TOMA DE INVENTARIO FÍSICO DE ALMACÉN.* Recuperado el 27 de Septiembre de 2019, de EGASA: https://catedraalimentacioninstitucional.files.wordpress.com/2017/02/03-00-procedimiento-n-003-procedimiento-de-toma-de-inventario-fc3adsico-de-almacc3a9n.pdf

FONDO DE CULTURA ECONÓMICA. (23 de Julio de 2012). *Procedimiento de Toma Física de Inventario.* (G. d. Producción, Ed.) Recuperado el 28 de Septiembre de 2019, de FONDO DE CULTURA ECONÓMICA: https://www.fondodeculturaeconomica.com/subdirectorios\_site/Normateca/FCE-GPR.01.pdf

Guerrero, S. (2009). *Inventarios Manejo Y Control* (1 ed.). Bogotá: Eco. Recuperado el 28 de Septiembre de 2019

Hérnandez, R. (2007). *LIBRO DE LOGÍSTICA DE ALMACENES.* LA HABANA, CUBA. Recuperado el 28 de Septiembre de 2019

IngenioEmpresa. (30 de Abril de 2018). *Valuación o valoración de inventarios: métodos, procedimiento y ejemplo*. Recuperado el 07 de Noviembre de 2019, de https://ingenioempresa.com/metodos-valoracion-inventarios/#Metodo\_1\_Primeros\_en\_entrar\_primeros\_en\_salir\_PEPS

Koneggui. (2019). *Toma Física de Inventario*. Recuperado el 27 de Septiembre de 2019, de Koneggui: http://koneggui.com.ec/gestion-de-logistica/toma-fisica-de-inventarios

LÓPEZ, A. A. (2015). *PROPUESTA PARAELABORAR UN MANUAL DE PROCEDIMIENTOS PARA EL MANEJO Y CONTROL DE INVENTARIOS EN LA EMPRESA TRACTEC SAS.* Recuperado el 28 de Septiembre de 2019, de UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA FACULTAD CIENCIAS ECONÓMICAS Y ADMINISTRATIVASESCUELA DE ADMINISTRACIÓN DE EMPRESAS TUNJA-BOYACÁ: https://repositorio.uptc.edu.co/bitstream/001/1477/2/TGT-225.pdf

queaprendemoshoy. (27 de Octubre de 2013). *¿Qué nos dice la Ley de Pareto?* Recuperado el 05 de Noviembre de 2019, de http://queaprendemoshoy.com/que-nos-dice-la-ley-de-pareto/

Sánchez, L. V. (2012). *Logística de Inventarios en empresas comercializadoras de productos de marroquinería.* Bogotá. Recuperado el 28 de Septiembre de 2019

Wikipedia. (29 de Octubre de 2019). *Android Studio*. Recuperado el 08 de Noviembre de 2019, de https://es.wikipedia.org/wiki/Android\_Studio

Wikipedia. (2019). *Cinépolis*. Recuperado el 14 de Septiembre de 2019, de Wikipedia: https://es.wikipedia.org/wiki/Cin%C3%A9polis

Wikipedia. (2020, 30 marzo). visual modeling and design tool based on the OMG UML. Recuperado 31 marzo, 2020, de <https://en.wikipedia.org/wiki/Enterprise_Architect_(software)>

Wikipedia. (2020, 30 marzo). Firebase. Recuperado 05 mayo, 2020, de <https://es.wikipedia.org/wiki/Firebase#Firebase_Auth[13]%E2%80%8B>

# **Glosario de términos**

**Conteo Físico:** Es el procedimiento de control para verificar la existencia de algún producto.

**Existencia Física:** Se considera que un producto está en existencia física cuando se encuentra en algún almacén físicamente esté o no esté registrado en el inventario del sistema.

**Existencia en Sistema:** Es el valor del producto que indica el sistema que debe de haber físicamente.

**Toma Física de Inventario:** La toma física de inventario consiste en llevar a cabo un proceso ordenado de verificación física de los bienes, materia prima o producto terminado a una fecha determinada, con el fin de asegurar su existencia real.

**Unidad de Negocio:** Es el identificador que tiene cada sucursal de Cinépolis.

1. Cuadrar: Es que los productos que re reflejen en sistema coincidan con los productos que hay físicamente. [↑](#footnote-ref-1)
2. Productos estrella del cine: Son productos que siempre debe de haber en el almacén, por ejemplo, canastas de palomitas y vasos de refresco. [↑](#footnote-ref-2)
3. Coffee Tree: Negocio propio de Cinépolis donde se ofrecen productos como frappés, crepas, cafés, papas fritas, entre otros productos. [↑](#footnote-ref-3)
4. Frappé: Bebida refrescante que se prepara con hielo picado y jugo de fruta, café u otras bebidas o ingredientes, como jarabes o esencias. [↑](#footnote-ref-4)
5. Vale: Es un documento que permite conocer las medidas y cálculos de un objeto y puede construir miles de cosas comerciales, ya sea, para pagar un [producto](https://es.wikipedia.org/wiki/Producto_(marketing)), o [bien](https://es.wikipedia.org/wiki/Bien_econ%C3%B3mico) un [servicio](https://es.wikipedia.org/wiki/Servicio_(econom%C3%ADa)). Puede representar el pago total o parcial ([descuento](https://es.wikipedia.org/wiki/Descuento_(mercadotecnia)) inmediato en el momento de la compra por la cantidad que aparece en el vale). [↑](#footnote-ref-5)
6. Base Rich: Concentrado de crema para batir, la cual por ser una base se requiere diluir en proporción de 2 partes de crema por una de otro líquido: leche, crema láctea, pulpa de fruta o agua purificada. [↑](#footnote-ref-6)
7. Fondo: Dinero propio del cine que se asigna a los empleados de puntos de venta, para que estos, puedan empezar a vender. A los empleados de los puntos de venta de taquilla, se les asigna un fondo de $1,500 pesos y a los empleados de los puntos de venta de dulcería, se les asigna un fondo de $600. [↑](#footnote-ref-7)
8. Stick de almacén: Persona que cubre los descansos y/o vacaciones del encargado principal de almacén y del auxiliar de almacén. [↑](#footnote-ref-8)
9. **Costo de venta:** Es el costo en que se incurre para comercializar un bien o para prestar un servicio. Es el valor en que se ha incurrido para producir o comprar un bien que se vende. [↑](#footnote-ref-9)