



**Universidad
Andrés Bello**

Facultad de ingeniería
Ingeniería Civil informática
Ingeniería en software

“Proyecto APP móvil Joyería Fevida”

Hito 3

Grupo 8

Jefe proyecto: Vicente Tapia Vera - V.tapiavera@gmail.com

Proyecto en trello: <https://trello.com/b/3VgAeJ52/proyecto-fevida>

Proyecto en git: https://github.com/Ampersandx/ICF232_202010_Grupo8

GoogleDrive:

https://drive.google.com/drive/folders/1w1tluKl_wA0nJlmSSDNAvcFmvrj5cHqV?usp=sharing
[Fmvrj5cHqV?usp=sharing](https://drive.google.com/drive/folders/1w1tluKl_wA0nJlmSSDNAvcFmvrj5cHqV?usp=sharing)

Integrantes: Matias Cerda González.

Rafael Comnte Bahamondes.

Victor Martinez Parada.

Vicente Tapia Vera.

Profesores: Pablo Schwarzenberg.

Gustavo Gatica.

Fecha: 05-12-2020

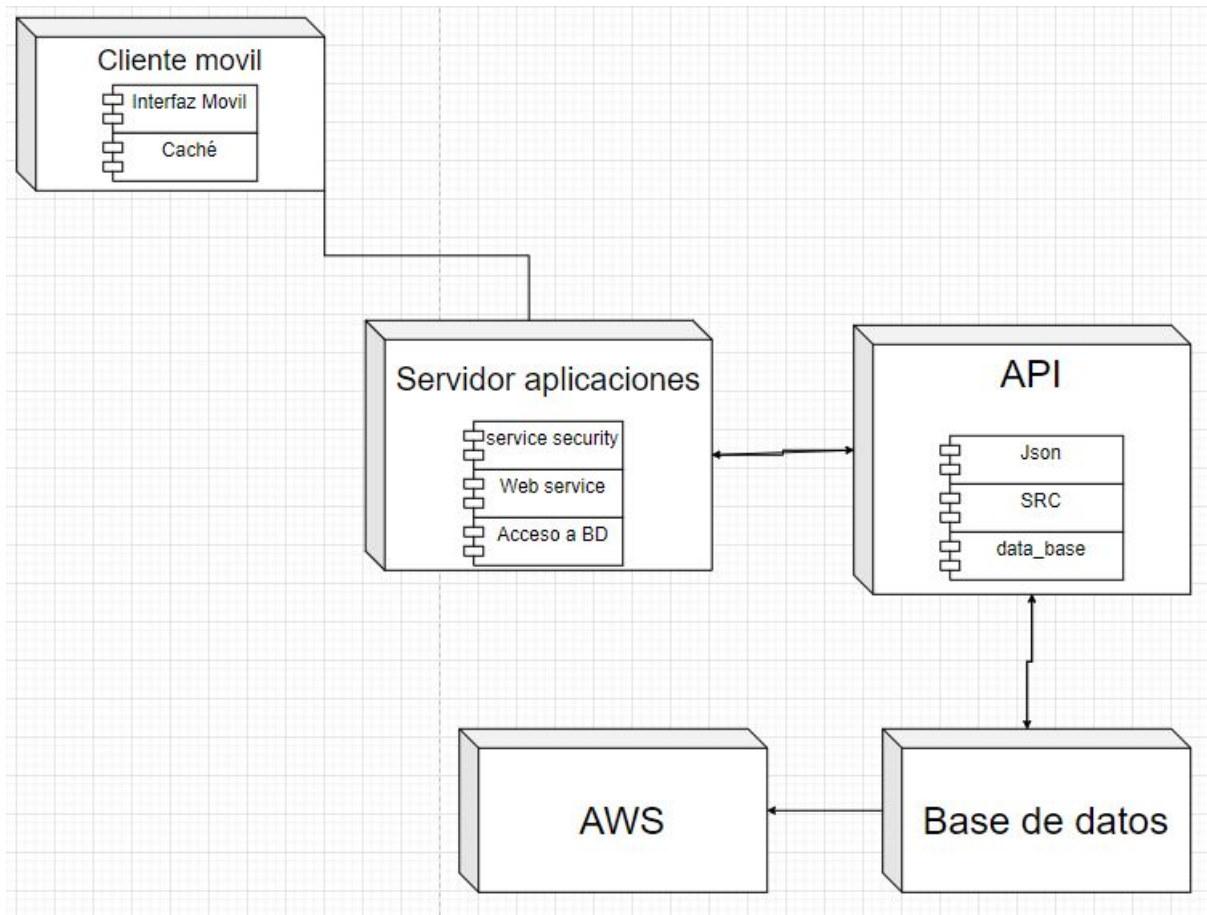
Índice:

Introducción	2.
Diagrama de despliegue	3.
Diagrama de flujo	4.
Diagrama de estados	5.
Diagrama de clases	6.
Diagrama de componentes	7.
Dependencias	8.
Backlog	10.
Burndown Chart	13.
Conclusión	14.

Introducción:

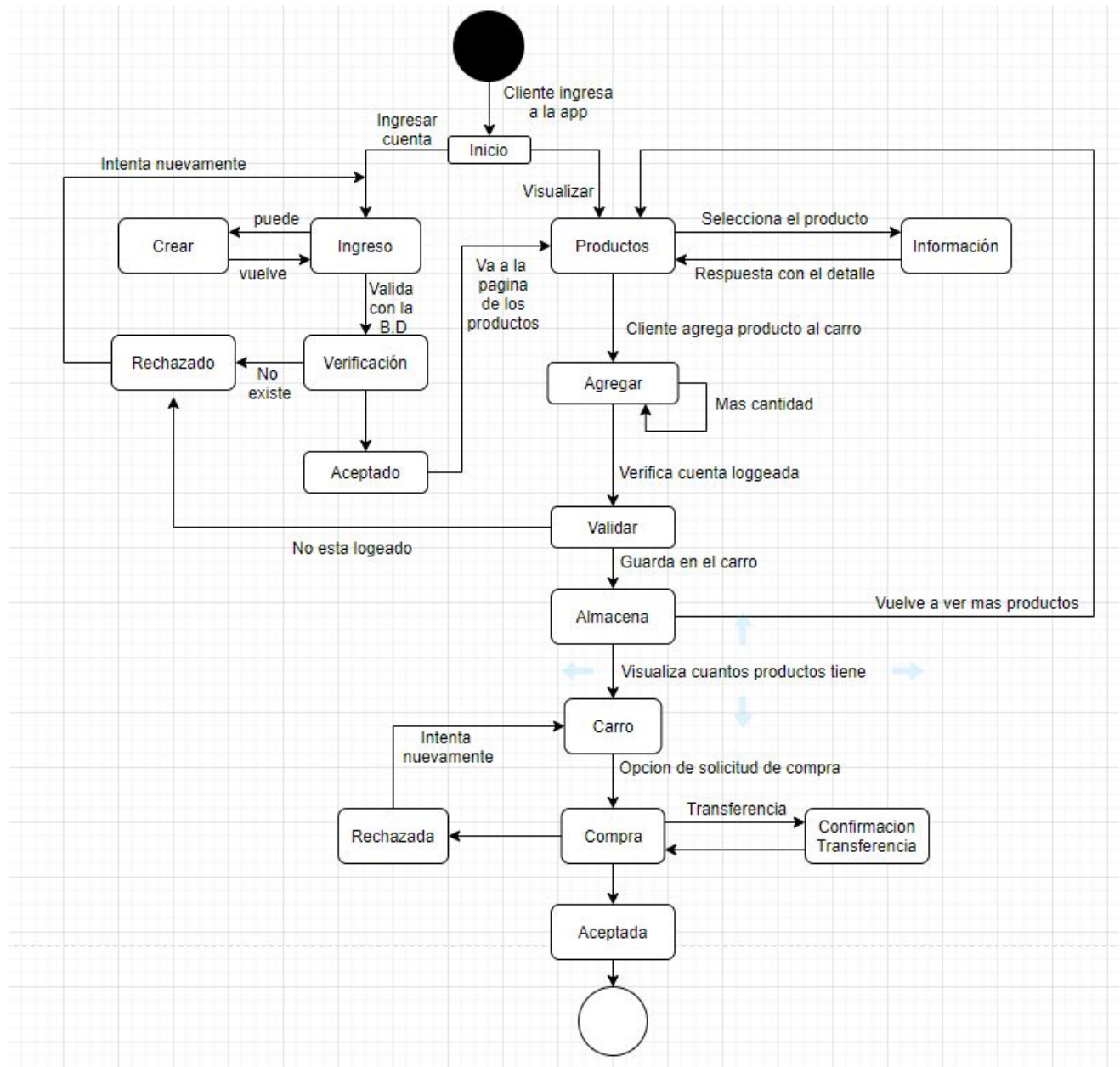
En el presente documento se dará a conocer las principales actividades que se realizaron para el Hito 3, además de explicar funcionalidad y desarrollo de la aplicación, mostraremos las dificultades que tuvimos al momento de programar y ejecutar nuestro proyecto esto se dará a conocer a base de diagramas y el mismo backlog de las actividades y concluimos con una reflexión del proyecto.

Diagrama de despliegue:



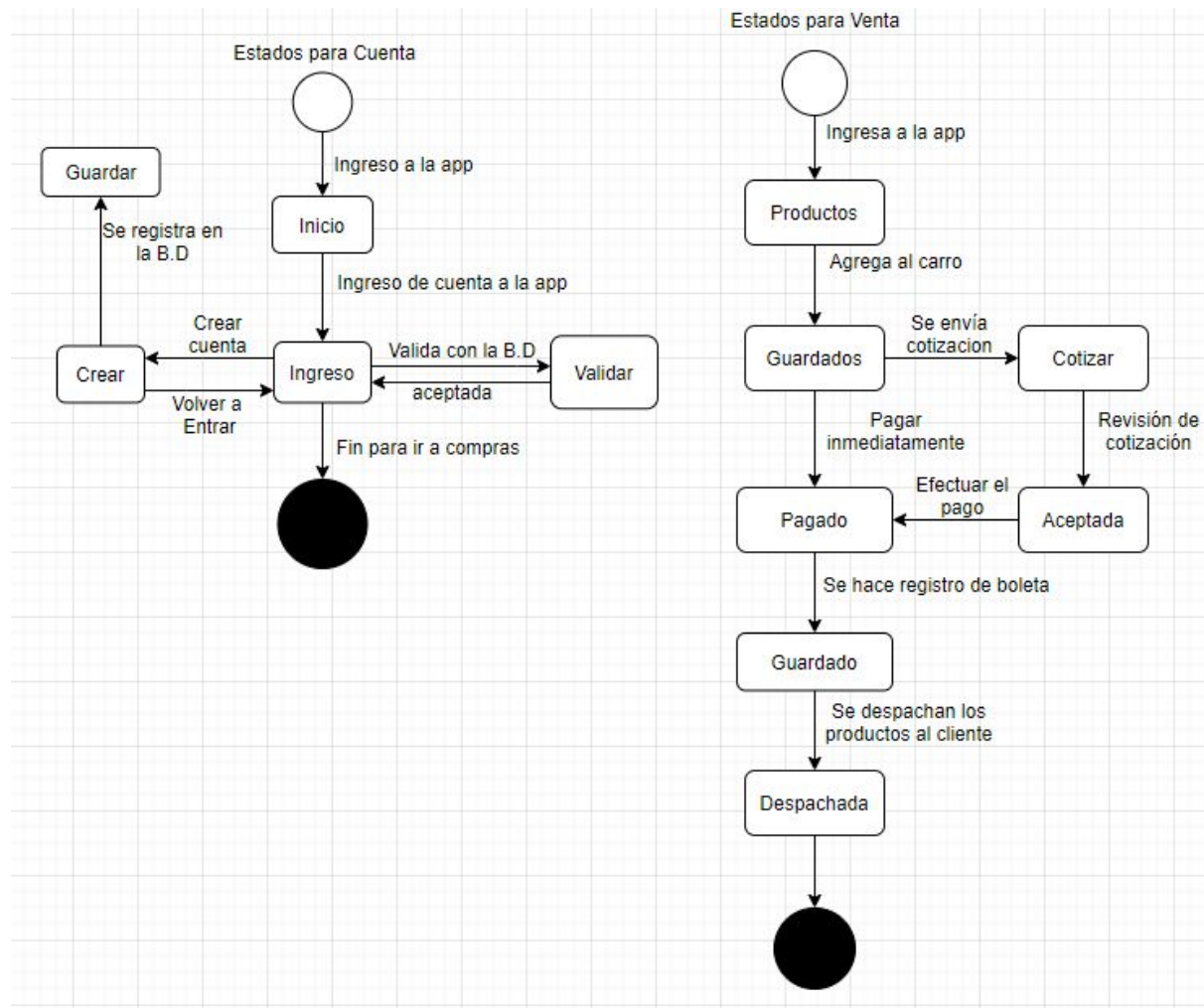
En el diagrama de despliegue se puede apreciar la vista física que tiene el proyecto, se muestra que la aplicación con la base de datos está conectada a través de una API (Application Programming Interface) la cual da el host necesario para poder ejecutar la app, además trabajamos con amazon web service para conectar nuestra base de datos y transformarla a una forma global y de acceso libre para los desarrolladores.

Diagrama de Flujo:



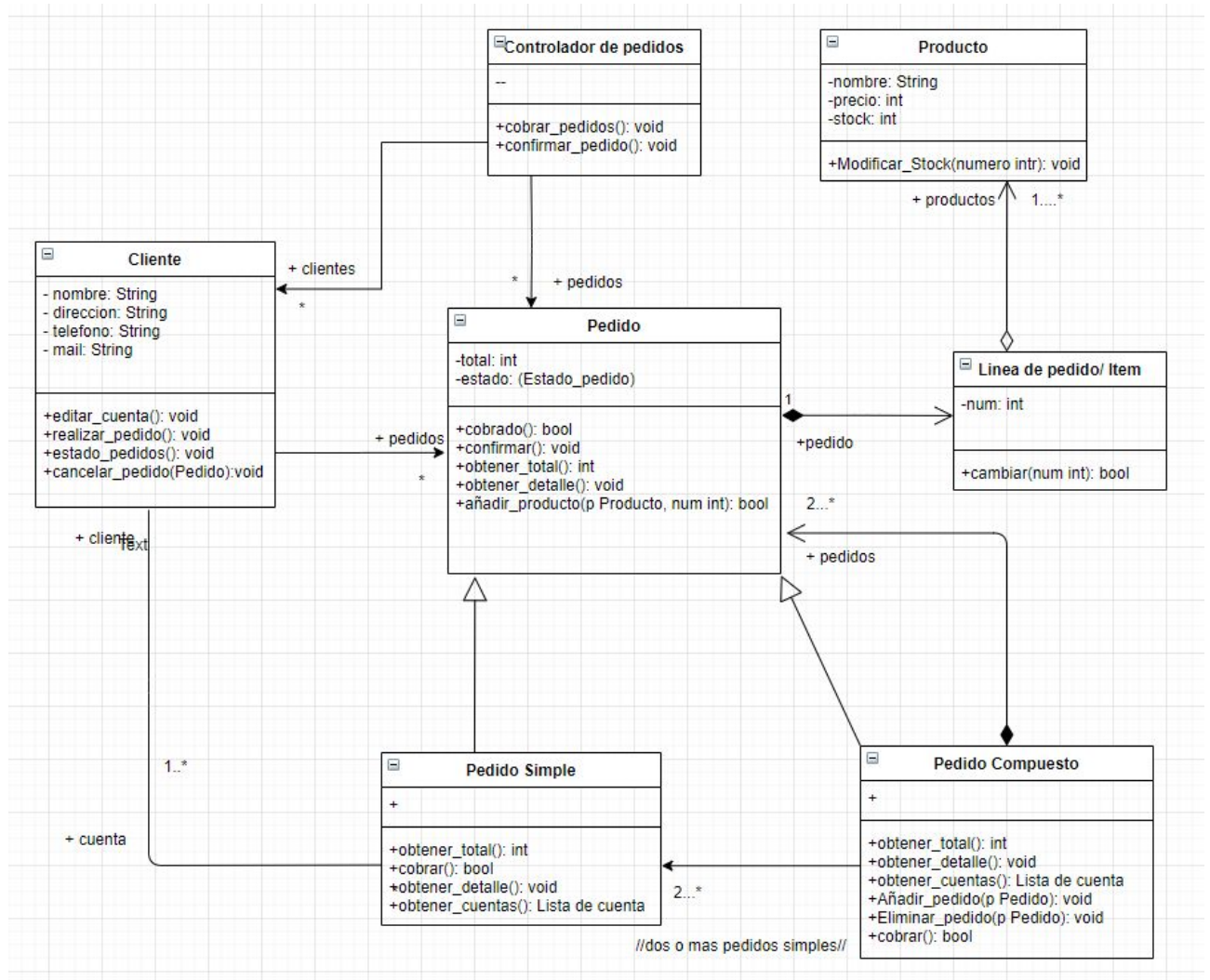
En el diagrama de flujo la unica actualizacion que se le ha dado es el cambio del método de pago que ahora se requiere una confirmación para pasar a aceptar la compra, esta se hace por el método de transferencia la cual te llegarán los datos de la cuenta al correo o “whatsapp”, en lo demás el flujo de las actividades sigue igual.

Diagrama de estados:



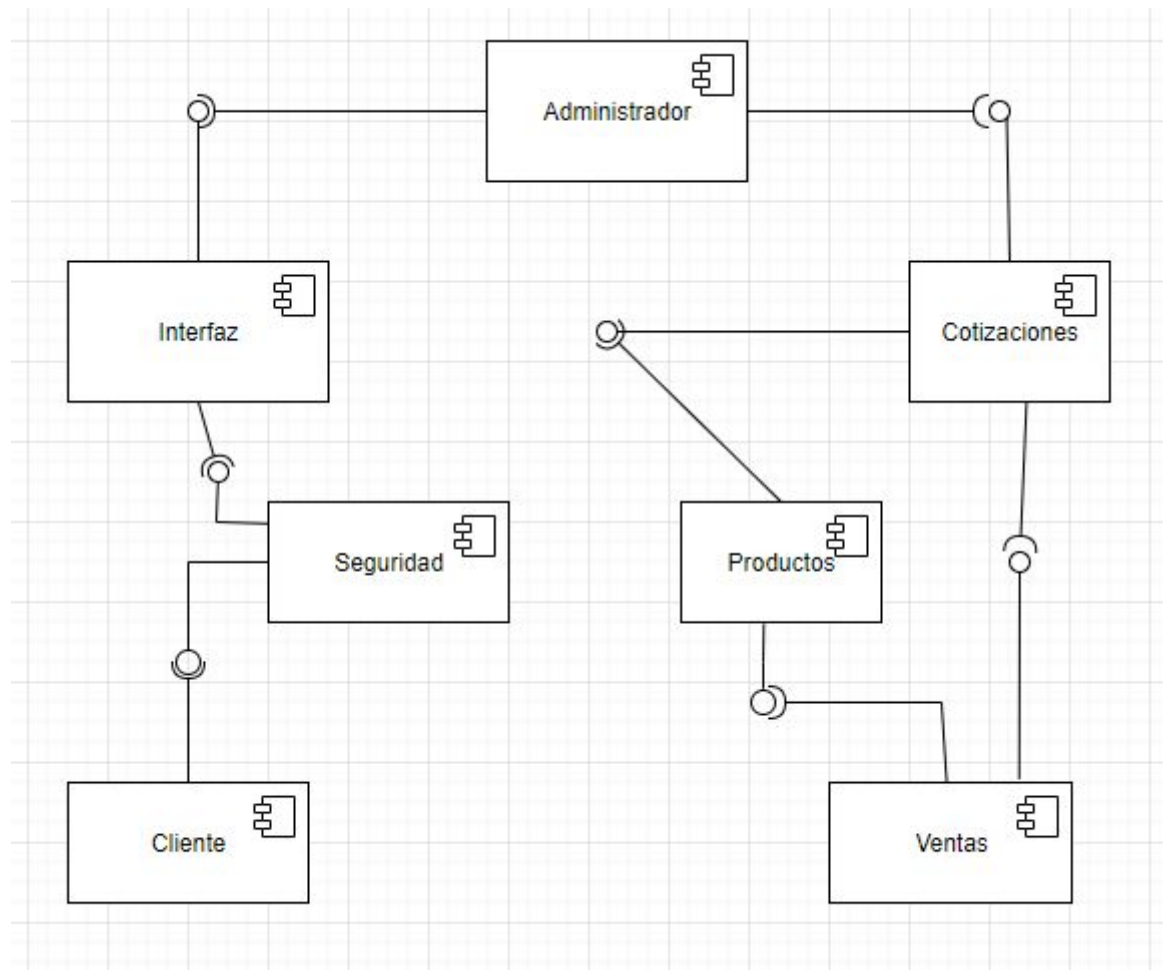
En el diagrama de estados las modificaciones que se hicieron a través de todos los hitos son que la boleta se envía directamente al correo del comprador y al correo de la empresa para tener el registro y no a la BD.

Diagrama de clases:



En el diagrama de clases se eliminó el cuadro de estado del producto, ya que esto se confirma hablando directamente con el encargado de la tienda, además ya no aplicamos códigos de descuento en las compras ya que al comprar por mayor el precio del producto cambia solo y no se aplica un descuento aparte según políticas de la empresa. Dentro tenemos los pedidos simples que son comprar menores de productos unitarios y además tenemos pedidos compuestos que son compras al por mayor y de productos variados.

Diagrama de componentes:



En el diagrama de componentes cambiamos algunas dependencias no había al principio, por la rama de las ventas ya no hay una dependencia con la seguridad de la app sino que un administrador revisa y genera las rutas para ver los productos y cotizaciones de los clientes, además este administra la interfaz de la app que esta maneja lo que es la seguridad y datos que tiene el clientes además genera las vistas que tiene la app.

Nota: El diagrama de casos de uso está en forma de documento word dentro del drive que está en la portada del informe. Los diagramas subidos aquí son los que han sufrido modificaciones a lo largo del proyecto.

Las **dependencias** usadas en este trabajo fueron:

```
"dependencies": {
  "@angular/animations": "^10.2.3",
  "@angular/cdk": "^10.2.7",
  "@angular/common": "~10.0.0",
  "@angular/core": "^10.2.3",
  "@angular/forms": "~10.0.0",
  "@angular/http": "^7.2.16",
  "@angular/material": "^10.2.7",
  "@angular/platform-browser": "^10.0.14",
  "@angular/platform-browser-dynamic": "~10.0.0",
  "@angular/router": "~10.0.0",
  "@capacitor/android": "^2.4.4",
  "@capacitor/core": "2.4.4",
  "@ionic-native/core": "^5.0.0",
  "@ionic-native/splash-screen": "^5.0.0",
  "@ionic-native/status-bar": "^5.0.0",
  "@ionic/angular": "^5.0.0",
  "@ionic/storage": "^2.3.1",
  "concurrently": "^5.3.0",
  "mysql": "^2.18.1",
  "rxjs": "^6.6.3",
  "rxjs-compat": "^6.6.3",
  "tslib": "^2.0.0",
  "zone.js": "~0.10.3"
},
```

```
"devDependencies": {
  "@angular-devkit/build-angular": "^0.1100.0",
  "@angular/cli": "10.0.8",
  "@angular/compiler": "~10.0.0",
  "@angular/compiler-cli": "~10.0.0",
  "@angular/language-service": "~10.0.0",
  "@capacitor/cli": "2.4.4",
  "@ionic/angular-toolkit": "^2.3.0",
  "@types/jasmine": "~3.5.0",
  "@types/jasminewd2": "~2.0.3",
  "@types/node": "^12.11.1",
  "codelyzer": "^6.0.0",
  "cordova-sqlite-storage": "^5.1.0",
  "express": "^4.17.1",
  "jasmine-core": "~3.5.0",
  "jasmine-spec-reporter": "~5.0.0",
  "karma": "~5.0.0",
  "karma-chrome-launcher": "~3.1.0",
  "karma-coverage-istanbul-reporter": "~3.0.2",
  "karma-jasmine": "~3.3.0",
  "karma-jasmine-html-reporter": "^1.5.0",
  "nodemon": "^2.0.6",
  "protractor": "~7.0.0",
  "ts-node": "~8.3.0",
  "tslint": "~6.1.0",
  "typescript": "~3.9.5"
},
```

Las dependencias sirven para el correcto funcionamiento del programa. En el cual su instalación se debe hacer por consola en la carpeta del proyecto. Un ejemplo de como hacerlo es el siguiente:

```
PS C:\xampp\htdocs\ICF232_202010_Grupo8\Codigo> npm i "@angular/core"
```

en el cual yo tengo mi programa en la carpeta Código y desde la consola ingreso a la carpeta y se ejecuta el comando npm i "@angular/core".

Así con todas las dependencias.

(Observación: Hay dependencias que contienen otras dependencias, en el caso de instalar dependencias duplicadas aparecerán Warning o en el peor de los casos un error.)

Backlog:

ID	Requerimientos	Prioridad	Estimacion de esfuerzo (SP)	Sprint (Semanal)	Prueba
	Esquematzar la App para tener una idea de como empezar a desarrollarla	1	4		1 Tener todo planeado con anterioridad
	Programar la interfaz en HTML y CSS	2	2		2 Poder entrar a la App sin saltar errores y poder ver el banner
	Crear la seccion de Crear Cuenta en HTML y CSS	3	2		2 Poder saltar desde la interfaz hasta "Crear cuenta" al presionar el boton del mismo nombre
	Crear la seccion de Login en HTML y CSS	4	2		2 Poder saltar desde la interfaz hasta "Login" al presionar el boton del mismo nombre
	Crear una base de datos para almacenar a los usuarios y Articulos	5	4		3 Poder hacer consultas de los usuarios y articulos de prueba
	Crear la seccion Productos	6	3		3 Poder saltar desde la interfaz hasta "Tienda" ya sea por el boton "Tienda" o por el buscador
	Crear la seccion preguntas frecuentes	7	2		4 Poder saltar desde la interfaz hasta "Preguntas frecuentes" y visualizar la informacion pertinente
	Conectar la base de datos con la App	8	4		4 Si agregas los datos al crear cuenta, estos se guardaran en una base de datos
	Agregar las restricciones pertinentes para Crear Cuenta	9	3		5 no se puede ingresar correos o usuarios ya registrados, los correos sin @ no funcionan, las dos contraseñas ingresadas deben coincidir
	Agregar Las restricciones pertinentes para Login	10	3		5 Los datos ingresados deben estar almacenadas en la base de datos, alineados con el mismo ID
	App capaz de Crear y logear cuentas correctamente	11	3		6 Si no existe tal usuario se puede crear desde la app en "Crear Cuenta" si ya existe en la base de datos se logeara con "Login"
	Importar los productos de la base de datos a la App	12	3		6 Poder visualizar los productos de la base de datos en la App
	Estetizar la pagina de Productos	13	2		6 la seccion de productos se ve bien y es intuitiva
	Crear la lista de articulo en carrito	14	4		7 Al presionar "al carrito" la Id del articulo se almacenara en una lista
	Agregar un contador para el numero de articulos en la lista del carrito	15	1		7 Al presionar "al carrito" el contador aumentara en 1 su valor actual
	Crear el apartado de Cotizaciones	16	2		7 habra una opcion para consultar las cotizaciones
	Importar correctamente los datos de la lista a la seccion de vista del carrito	17	4		8 los datos ya se pueden llamar correctamente a la seccion "carrito"
	Agregar todas las funciones necesarias a la seccion "carrito" (quitar del carro, ir a pagar, visualizar precios individuales y finales)	18	4		8 Llegar hasta la seccion de elegir opcion de pago sin contratiempos, poder visualizar los precios y quitar productos de la lista
	Modificar cuenta	19	4		9 Un usuario podra cambiar sus datos de registros, tras iniciar session
	Se necesita estar logeado	20	3		9 No se puede agregar al carrito o comprar sin antes logearse

				habra un espacio para escribir palabras clave (ej: Anillo) que se utilizaran en forma de query para ubicar un producto por medio de su nombre
	Programar el buscador	21	4	9
	Actualizar el contador del carrito, en la vista.	22	1	10
	Actualizar el contador del carrito, post-compra	23	1	10
AD21	La app debe disponer de algun metodo de pago	24	3	10
	<u>Agregar Transbank</u>	25	2	10
	Agregar un boton para chatear por What's App con el admin	26	3	11
AD22	Generar una boleta para las compras	27	2	11
	Modulo de historial de compras	28	3	11
M11	Hacer publicidad personalizada para los clientes	29	4	12
M12	Implementar una API	30	6	12
V22	Hacer descuentos al comprar al por mayor	31	2	12

El Backlog contiene 31 Tareas, las cuales se separan en 3 grupos, dependiendo de las tareas que se esperan completar para la entrega de cada Hito (Hito 1 => Verde, Hito 2 => Amarillo, Hito 3 => Naranja).

de las 31 tareas completamos 19 tareas, 6 quedaron en proceso y 6 no se alcanzaron a comenzar.

de las 19 tareas completadas tenemos:

- Esquematizar la App. (Hito 1)
- Crear la interfaz. (Hito 1)
- Crear la sección de crear cuenta. (Hito 1)
- Crear la sección de login. (Hito 1)
- Crear la base de datos, para Usuarios y Productos. (Hito 1)
- Crear la sección de productos. (Hito 1)
- Conectar la base de datos con la App. (Hito 1)
- Agregar las restricciones de Crear Cuenta. (Hito 2)
- Agregar las restricciones del Login. (Hito 2)
- Agregar la funcionalidad para Crear cuenta y loguearse. (Hito 2)
- Estetizar la App, hacer que parezca App móvil en vez de una página web. (Hito 2)
- Crear lista de artículos. (Hito 2)
- Agregar contador de artículos en carrito. (Hito 2)
- Crear apartado de cotizaciones. (Hito 2)
- Importar datos de la lista a la vista del carrito. (Hito 2)
- Estar logueado es requisito para comprar. (Hito 3)
- Actualizar el contador del carrito. (Hito 3)
- Crear módulo de Historial de compra. (Hito 3)
- Implementar una API. (Hito 3)

Las 6 tareas que quedaron en proceso pero que no se concretaron fueron:

- Crear la sección de preguntas frecuentes.
- Importar la información de los artículos a la vista del mismo.
- Programar Buscador.
- Modificar Cuenta.
- Agregar las funcionalidades al carrito de compra (sección precompra)
- La app debe disponer de algún medio de pago. (redbank)

La primera tarea de las 5 no se concretó, porque a medida que pasaba el tiempo y se iba aplazando dicha tarea y después no encontrábamos espacio donde colocarla, sumado al hecho de que no era algo tan esencial para la app, dicha tarea se fue dejando de lado. Aun así se avanzó lo suficiente como para tener todas las preguntas y respuestas frecuentes escritas, incluso se hizo la sección (una versión poco estética), pero debido a lo antes dicho, esta nunca se ancló, pero está prácticamente hecha.

La segunda tarea se batalló bastante para implementarse, debido a que dio muchos problemas y constantemente se cambiaba el lenguaje con el que se haría (JavaScript y PHP), nunca llegó a funcionar del todo, las queries se lograban guardar en una variable, pero no se lograba que se imprimieran adecuadamente, en la vista de la App directamente no imprimía. Al final nunca llegó a funcionar, pero se escribió y también se borró bastante código al respecto.

La tercera tarea se programó utilizando PHP pero debido a que no se concretó la importación de información de artículos, no se pudo probar si el buscador funcionaba correctamente, por lo que nunca se ancló, pero en teoría debería funcionar al menos lo que se llegó a escribir.

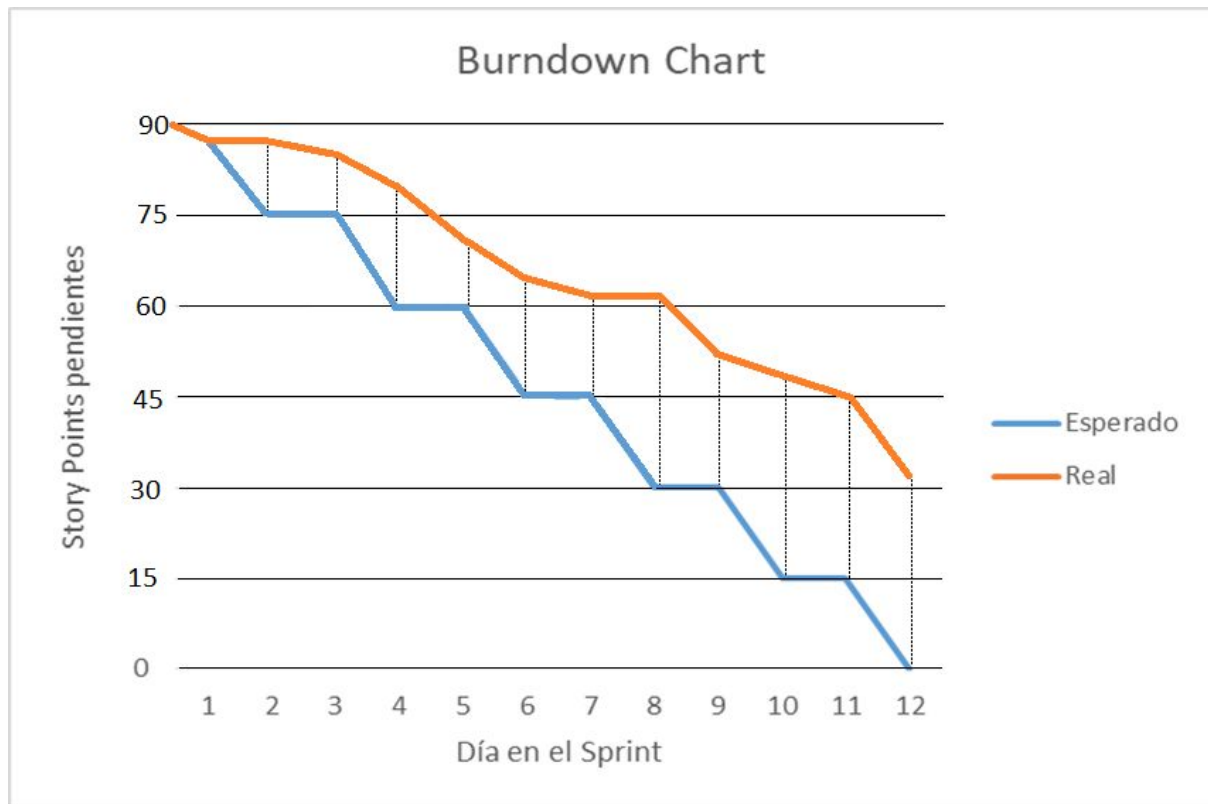
Al igual que con la segunda tarea, la cuarta tarea no se pudo concretar por problemas al interactuar con la base de datos, en la segunda siendo las consultas con la base de datos y esta vez no se logró actualizar los datos de la base de datos, al modificar información del perfil de usuario.

La quinta tarea nunca se logró concretar, se tardó mucho tiempo en la implementación de la API, la cual dio muchos problemas, pero se logró implementar. el carrito sin embargo no se le dedicó el suficiente tiempo y era una sección compleja en cuanto a funcionalidad

Y respecto a la Sexta y última tarea, no se pudo completar porque tenía una implementación más compleja y transbank daba más trabas de lo esperado al punto que podía tardar un mes o más en ver si validaba nuestra petición y para entonces sería muy tarde

Sumado a todo lo anterior dicho del Backlog adjunto la imagen del Burndown Chart donde se visualiza la expectativa del avance del trabajo, frente a la realidad, en una gráfica de semana laboral vs puntos de estimación de esfuerzo restantes y en la cual podemos ver que se completaron casi $\frac{2}{3}$ del trabajo. Y de haberse concretado las 6 tareas que quedaron a medias, hubiésemos tenido 76 de 90 Pts en vez de 56 de 90.

Burndown Chart:



En un comienzo partimos bien, pero a medida que pasaban los sprint se nos fueron acumulando tareas incompletas de sprints anteriores por lo cual siempre íbamos atrasados, en la semana 5 tratamos de completar todas las tareas pendientes y las que corresponden a esa semana, tuvimos el problema de conectar la app con una base de datos para guardar los datos de usuario, lo que hicimos fue crear una conexión local a una BD, y realizar pruebas, cosa que completamos en la semana 6 . Luego se nos fue de las manos el concretar tareas y atrasamos bastante hasta el sprint 12.

Conclusión:

Desde el inicio estuvimos perdidos debido a la falta de conocimiento sobre aplicaciones móviles, desde que frameworks utilizar, conexiones de la app con la base de datos, integraciones externas como redbank, generar un alcance tan amplio desconociendo la realidad de aplicar dichas tareas, intentamos hacer lo mejor posible en cuanto a la interfaz de la aplicación para generar una experiencia de usuario agradable.

Como grupo hicimos un “mea culpa” en el cual nos dimos cuenta que intentamos hacer todo de manera meticulosa y eso nos jugó en contra, ya que no logramos focalizar nuestro trabajo en una sola tarea, debido a eso fue poco lo que logramos concretar.

Tuvimos varios aprendizajes en este proyecto que valoramos mucho, por ejemplo el utilizar la herramienta de git para trabajar en conjunto, conocer nuevos frameworks, aprender nuevos lenguajes, también que debemos partir con tareas que sean funcionales, para luego comenzar a pulir detalles como mejoras de seguridad, interfaz, y mejoras en la base de datos.