PC-Drome

ARES 3.0

Fecha de entrega

20 de abril de 2021

Versión 1.0



Autores

Bryan Felipe Santillán Gómez

Diana Natalia Chaparro Sanabria

Diana Stefania García Yomayuza

Edwin Fabian Vesga Escobar

Fabian Andrés Rojas Molina

Jerson Giovany Osorio Cely

Julián David Tarazona Torres

# **Historial de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| Numero de cambio | Fecha del cambio | Descripción del cambio | Persona que realizó el cambio |
| 1 | 11/04/2021 | Se agrega la información de la vista física del sistema y el historial de cambios del documento | Stefania Garcia |
| 2 | 11/04/2021 | Se agregó la arquitectura | Edwin Vesga |
| 3 | 12/04/2021 | Se agregó la vista lógica del sistema | Fabian Rojas  Julián Tarazona |
| 4 | 13/04/2021 | Se agregó la estructura del sistema | Fabian Rojas  Julián Tarazona |
| 5 | 15/04/2021 | Se agregó el diagrama de secuencia de Armar PC | Diana Chaparro |
| 6 | 15/04/2021 | Se agregó el diagrama de secuencia de Registrarse | Fabian Rojas |
| 7 | 15/04/2021 | Se agregó la introducción y la interfaz de usuario | Stefania Garcia |
| 8 | 16/04/2021 | Se agregó la vista de procesos del sistema | Julián Tarazona |
| 9 | 17/04/2021 | Se agregó la persistencia | Bryan Santillán  Edwin Vesga |

# **Tabla de Ilustraciones**

[Ilustración 1- Arquitectura PCDrome 6](#_Toc69830688)

[Ilustración 2 Diagrama de componentes 7](#_Toc69830689)

[Ilustración 3 operaciones del diagrama de componentes 8](#_Toc69830690)

[Ilustración 4 Diagrama de despliegue 9](#_Toc69830691)

[Ilustración 5 Diagrama de actividad de registro de usuario 12](#_Toc69830692)

[Ilustración 6 Diagrama de actividad de armar PC 13](#_Toc69830693)

[Ilustración 7 Diagrama de actividad de calificar un producto 13](#_Toc69830694)

[Ilustración 8 Diagrama de clases 14](#_Toc69830695)

[Ilustración 9 Diagrama de secuencia de historia de usuario Registrarse 16](#_Toc69830696)

[Ilustración 10 Diagrama de secuencia de historia de usuario Armar PC parte 1 17](#_Toc69830697)

[Ilustración 11 Diagrama de secuencia de historia de usuario Armar PC parte 2 18](#_Toc69830698)

[Ilustración 12 Árbol de colecciones de procesador 19](#_Toc69830699)

# **Tabla de Contenidos**

[**Historial de Cambios** 3](#_Toc69819043)

[**Tabla de Ilustraciones** 5](#_Toc69819044)

[**Tabla de Contenidos** 6](#_Toc69819045)

[**Introducción** 7](#_Toc69819046)

[**Arquitectura** 8](#_Toc69819047)

[**Vista lógica del sistema** 9](#_Toc69819048)

[**Vista física del sistema** 10](#_Toc69819049)

[**Vista de procesos del sistema** 14](#_Toc69819050)

[**Diseño Detallado** 16](#_Toc69819051)

[**Estructura del sistema** 16](#_Toc69819052)

[**Comportamiento del sistema** 18](#_Toc69819053)

[**Persistencia** 20](#_Toc69819054)

[**Interfaz de usuario** 21](#_Toc69819055)

[**Anexos** 23](#_Toc69819056)

[**Referencias** 24](#_Toc69819057)

# **Introducción**

El diseño de software es la actividad en la cual se realizan los requisitos para producir una descripción de la estructura interna del software. Esto se realiza por medio de distintos diagramas de diseño. Los diagramas de diseño permiten al lector conocer más sobre la arquitectura, los programas que se van a utilizar, para que clase de ordenadores está diseñada la aplicación que se va a presentar.

Para entrar un poco en contexto **ARES 3.0** desea realizar una página web que permita diseñar un computador teniendo en cuenta las marcas preferidas por el usuario, el costo que estarían dispuestos a pagar y además obtener un computador que satisfaga sus necesidades.

Para que el lector conozca y entienda la forma en que se va a realizar esta página web se realizaron diferentes diagramas de diseño. El diagrama de clases describe la estructura del sistema mostrando las clases, los atributos, los métodos y las relaciones entre los objetos.

Un diagrama de despliegue permite visualizar el hardware y el software físico del sistema. Por medio de nodos que muestran como por medio de los elementos de hardware se observan las relaciones entre los artefactos del software

Así como los mencionados anteriormente existen muchos más modelos de diseño arquitectónico. El diseño arquitectónico representa los componentes del programa, son patrones definidos que proporcionan un marco definido para interactuar con el código fuente.

A lo largo de este documento se mostrará cómo se realizará la página web por medio de una arquitectura con capas que permite separar las partes que componen un sistema.

# **Arquitectura**

Para nuestro proyecto PCDrome, el cual requiere el desarrollo de un portal web para la compra y venta de partes de computadoras, la gestión de usuarios y el filtrado inteligente de partes para armar una computadora de manera personalizada, necesitamos una arquitectura la cual permita 3 cosas fundamentales. La primera, es la adaptabilidad a nuevos componentes. Segundo, que sea entendible en cuanto a la abstracción del problema y que permita separar claramente funcionalidades. La arquitectura que más se adapta a esto para nosotros ha sido la arquitectura por capas. (Pelaez, 2009)

Esta arquitectura es entendida por todos los integrantes del grupo, por lo que es la opción más adecuada para nuestro proyecto. Ahora bien, tomamos una arquitectura por capas llamada *Clean Architecture* (Martin, 2012)como referencia para adaptarla a nuestra idea de proyecto y al final quedo de la manera que se ve en la *ilustración #.* La arquitectura Clean es bastante usada en desarrollo de aplicaciones móviles que se enlazan con Firebase, y así mismo, en aplicaciones web que tienen altas transacciones y que constantemente deben actualizar servicios y demás. Sin embargo, es una arquitectura compleja de entender, por lo que como se ha comentado más arriba, la adaptamos de manera que fuese más entendible por todos.

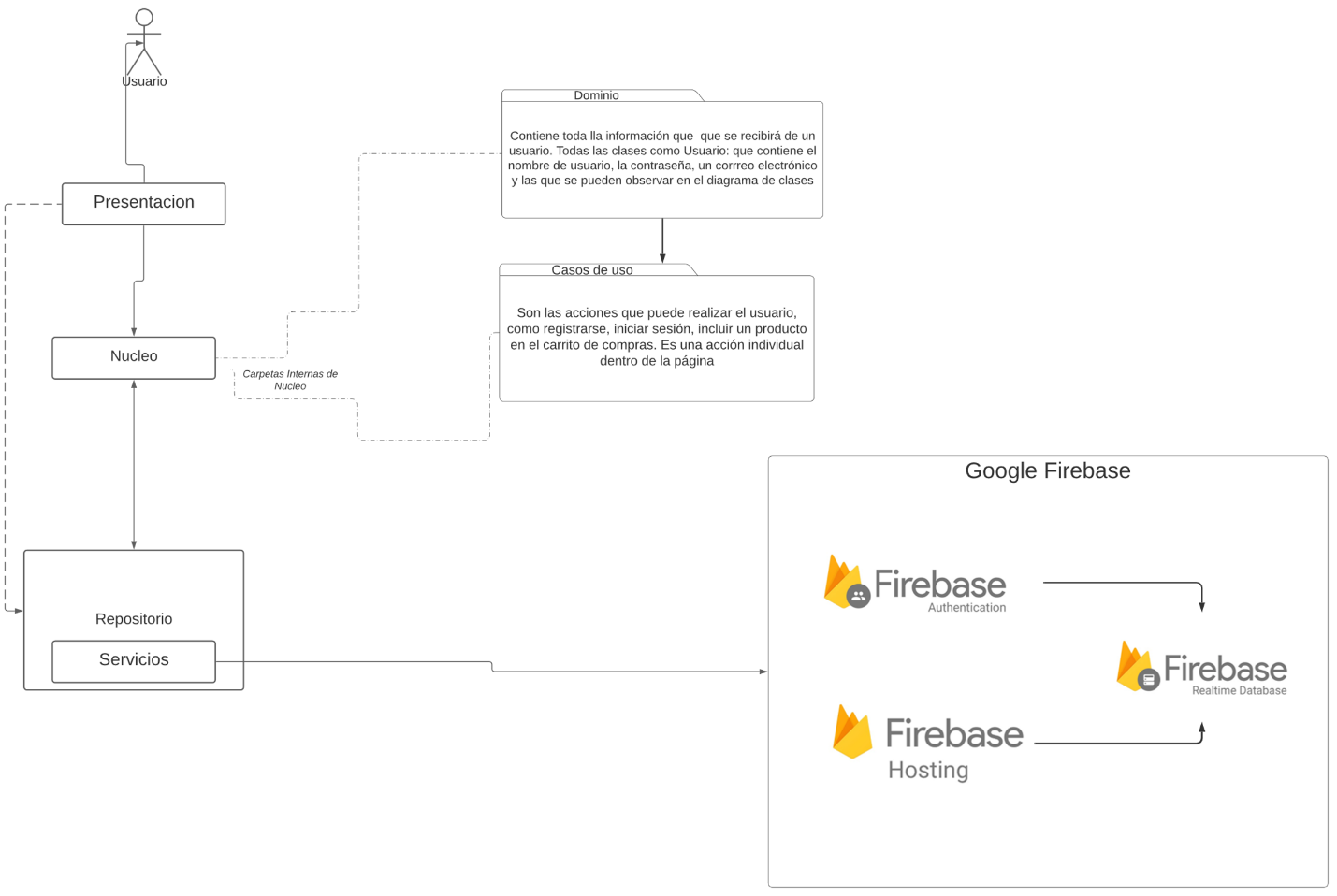


Ilustración 1- Arquitectura PCDrome

La capa superior, la capa de presentación, gestiona los componentes que interactúan directamente con el usuario. Luego, la capa llamada núcleo, no es un núcleo como tal, pero centraliza los casos de uso por cada actividad dentro de la aplicación web y además, tienen las clases que se enviaran y gestionaran luego hacia la capa de repositorio y posteriormente, a Firebase. Luego, la capa repositorio, se encarga de recibir y enviar la información a Firebase, pero, además, transforma los datos entrantes en datos visualizables o sencillos para ser gestionados fácilmente en la presentación.

## **Vista lógica del sistema**

En el diagrama de componentes se muestran los componentes diferenciados por color para indicar la capa a la cual pertenecen, el color verde es la capa de presentación, el azul es la capa de repositorio y el amarillo es la capa de núcleo.

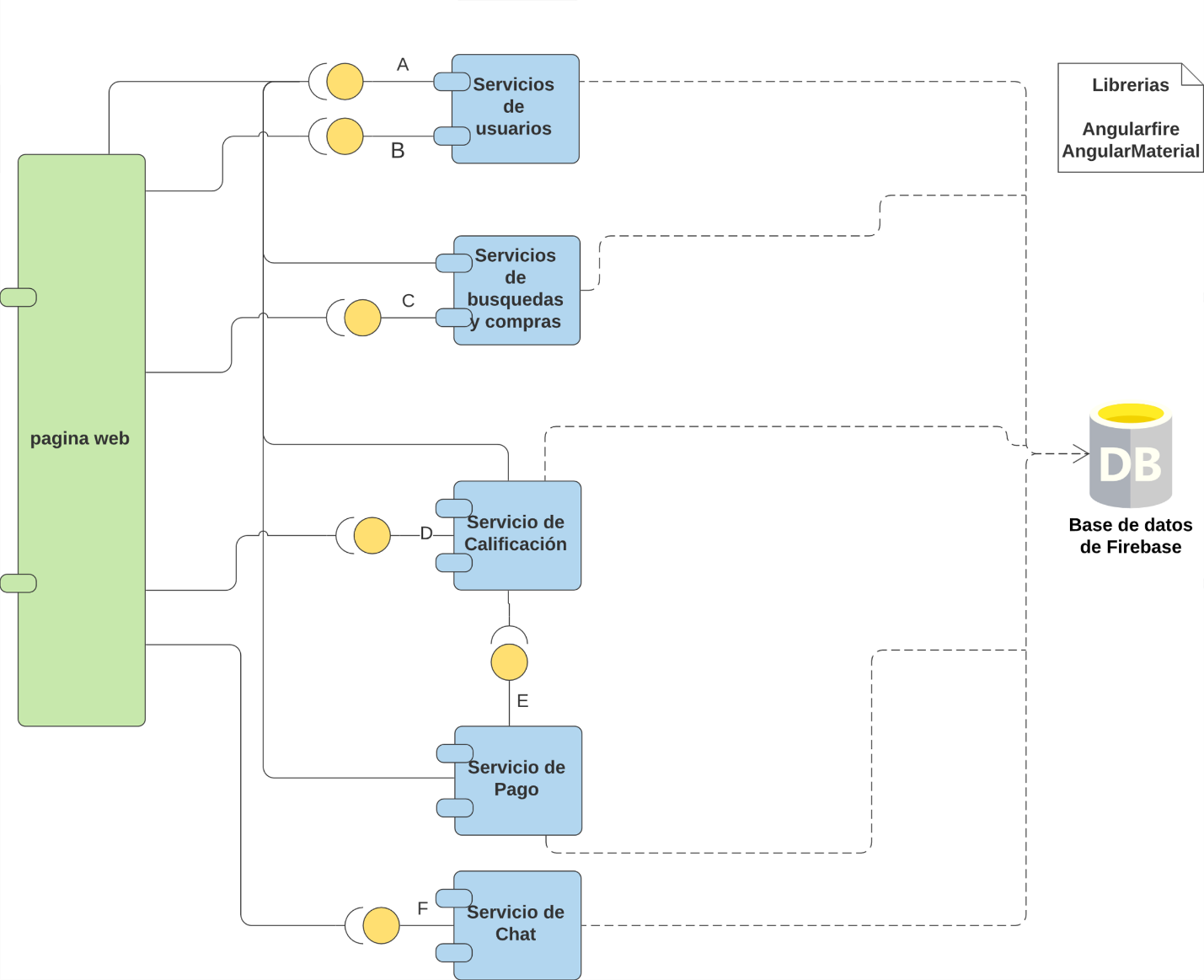


Ilustración 2 Diagrama de componentes

* **Página web:** El componente de página web representa el archivo de JavaScript que contendrá todas las funcionalidades que se le mostraran al usuario.
* **Servicios de usuarios:** contiene todas las operaciones que los usuarios pueden hacer con respecto a su información personal.
* **Servicios de búsqueda y compra:** Contiene las operaciones de la búsqueda de los productos y el cálculo del precio total por los productos seleccionados.
* **Servicio de calificación:** Contiene las operaciones para comentar y calificar un producto que el cliente haya comprado.
* **Servicio de pago:** Contiene la funcionalidad de pagar el o los productos almacenados en el carrito.
* **Servicio de chat:** Contiene la funcionalidad de enviar un mensaje al equipo de PC-DROME.
* **Usuarios:** Contiene las clases de cliente y administrador.
* **Productos:** Contiene las clases del carrito de compras, el pedido, el producto base y todas sus clases hijas que serían los componentes de una computadora.
* **Calificación:** Contiene las clases de comentario y calificación de un producto.
* **Medios de pago:** Contiene la clase del medio de pago que usara el cliente para pagar los productos.

Las operaciones que poseen las interfaces son las siguientes:

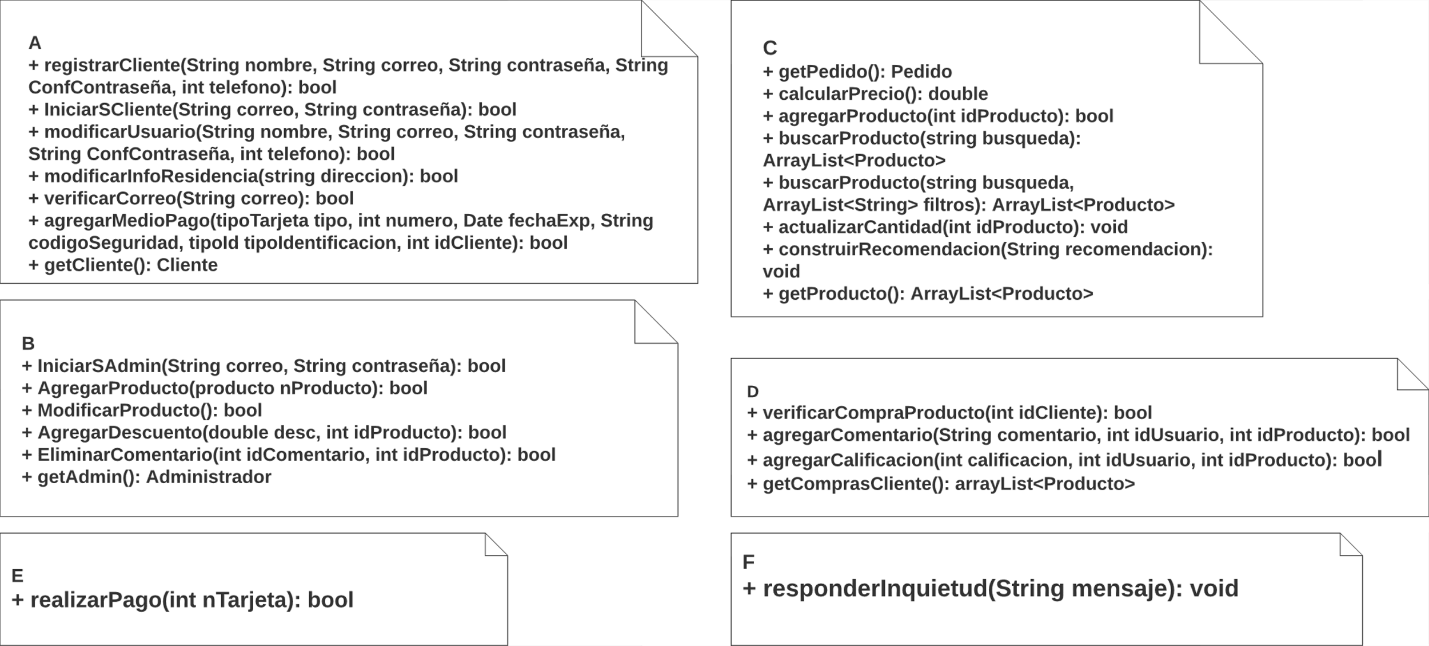


Ilustración 3 operaciones del diagrama de componentes

## **Vista física del sistema**

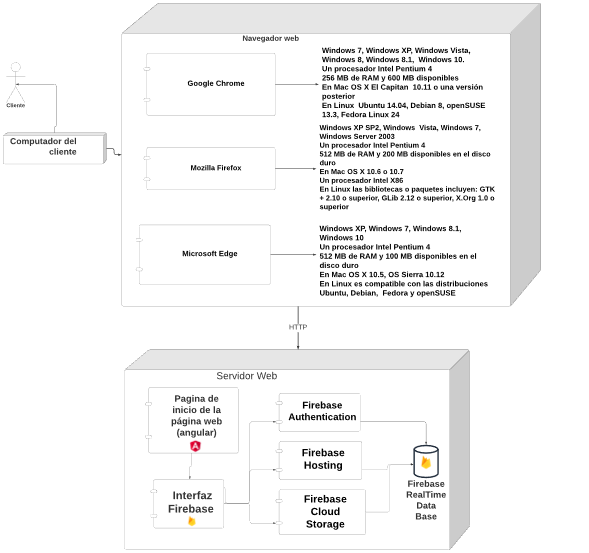


Ilustración 4 Diagrama de despliegue

En un diagrama de despliegue se utilizan dos tipos de conexiones: Nodos y conexiones.

Los nodos son aquellos que representan los elementos físicos que interactúan con el sistema. En el diagrama de despliegue presentado se tiene nodos distintos:

1. Computador del cliente: Permite al cliente conectarse a la aplicación por medio de un navegador web

Un nodo puede contener otros elementos, como componentes. En estos nodos se encuentran:

1. Navegador web: Por medio del navegador web el cliente puede observar la información que contiene una página web. El navegador interpreta el código en que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido
   1. Los navegadores para los que está diseñada la página web son Google Chrome, Mozilla Firefox y Microsoft Edge
      * Para Google Chrome se requiere que el computador del cliente requiera versiones de Windows como: Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10.

En cuanto a Linux se requiere de: Versiones de Ubuntu 14.04, Debian 8, openSUSE 13.3, Fedora Linux 24

En Mac se requieren versiones como: OS X El Capitán 10.11 o versiones posteriores 10.11.2, 10.11.3, entre otras

El computador del cliente debe tener un procesador Intel Pentium 4 con una memoria RAM de 256 MB y 600 MB disponibles en el disco duro.

* + - Para Mozilla Firefox se requiere que el computador del cliente requiera versiones de Windows como: Windows XP SP2, Windows Vista, Windows 7, Windows Server 2003.

En cuanto a Linux se requiere de las siguientes bibliotecas o paquetes: GTK + 2.10 o superior, GLib 2.12 o superior, X.Org 1.0 o superior

En Mac se requieren versiones como: OS X 10.6 0 10.7 con un procesador Intel X86

El computador del cliente debe tener un procesador Intel Pentium 4 con una memoria RAM de 512 MB y 200 MB disponibles en el disco duro.

* + - Para Microsoft Edge se requiere que el computador del cliente requiera versiones de Windows como: Windows XP, Windows 7, Windows 8.1, Windows 10.

En cuanto a Linux es compatible con las distribuciones Ubuntu, Debian, Fedora y openSUSE

En Mac se requieren versiones como: OS X 10.5, OS Sierra 10.12

El computador del cliente debe tener un procesador Intel Pentium 4 con una memoria RAM de 512 MB y 100 MB disponibles en el disco duro.

1. Servidor web: Devuelve información cuando recibe peticiones por parte de los usuarios. Permite a los usuarios ver una página web en su navegador
   1. Para mostrar el inicio de la página web se usará Angular. Angular es una aplicación donde la interacción y navegación se hacen en una sola página, lo que evita que se recargue la página varias veces. Permite construir el Front-end, hará las peticiones para que el Back-end le devuelva información
   2. En cuanto a la interfaz se va a usar Firebase. Firebase permite almacenar todo en la nube, tener un control de rendimiento por medio de métricas analíticas, gestionar fácilmente nuevos usuarios. En Firebase se usará:
      1. Firebase Authentication: Permite obtener credenciales de autenticación del usuario como el correo electrónico y la contraseña del usuario. Se verificarán esas credenciales y mostrarán una respuesta al cliente
      2. Firebase Hosting: Ofrece la infraestructura, las funciones y las herramientas orientadas a la implementación y administración de sitios web. Incluye SSL para que el contenido se publique siempre de forma segura
   3. Firebase Realtime Database: Es una de las grandes ventajas de Firebase, es una base de datos conectada a la nube donde los datos se almacenan y se sincronizan en tiempo real con cada cliente conectado, sirve para millones de usuarios y no afecta la capacidad de rápida respuesta

Las conexiones representan una asociación entre dos nodos o componentes en un nodo, donde se puede transmitir información entre ellos en forma de mensajes o señales y se agrega una etiqueta que represente de que medio se realiza la conexión.

Las conexiones presentadas en el diagrama de despliegue empiezan con la comunicación entre el cliente y el computador del cliente. Luego, por medio del computador del cliente se puede realizar una conexión desde un navegador web como Google Chrome, Mozilla Firefox o Microsoft Edge teniendo en cuenta las características específicas que debe tener el computador por cada navegador. Por medio de un protocolo HTTP que permite las transferencias de información a través de archivos en la World Wide Web, se conecta a un servidor donde se observa la página web por medio de Angular. Angular se conecta con la interfaz de Firebase, Firebase pasa por distintos servicios como Firebase Authentication, Firebase Hosting y esos servicios llevan información a la base de datos de Firebase.

## **Vista de procesos del sistema**

Los diagramas de actividad describen de forma simplificada el comportamiento del sistema que se está modelando. A continuación, se presentan los diagramas de actividad de algunas de las principales interacciones que tiene el cliente con el sistema.

**Registrarse:**

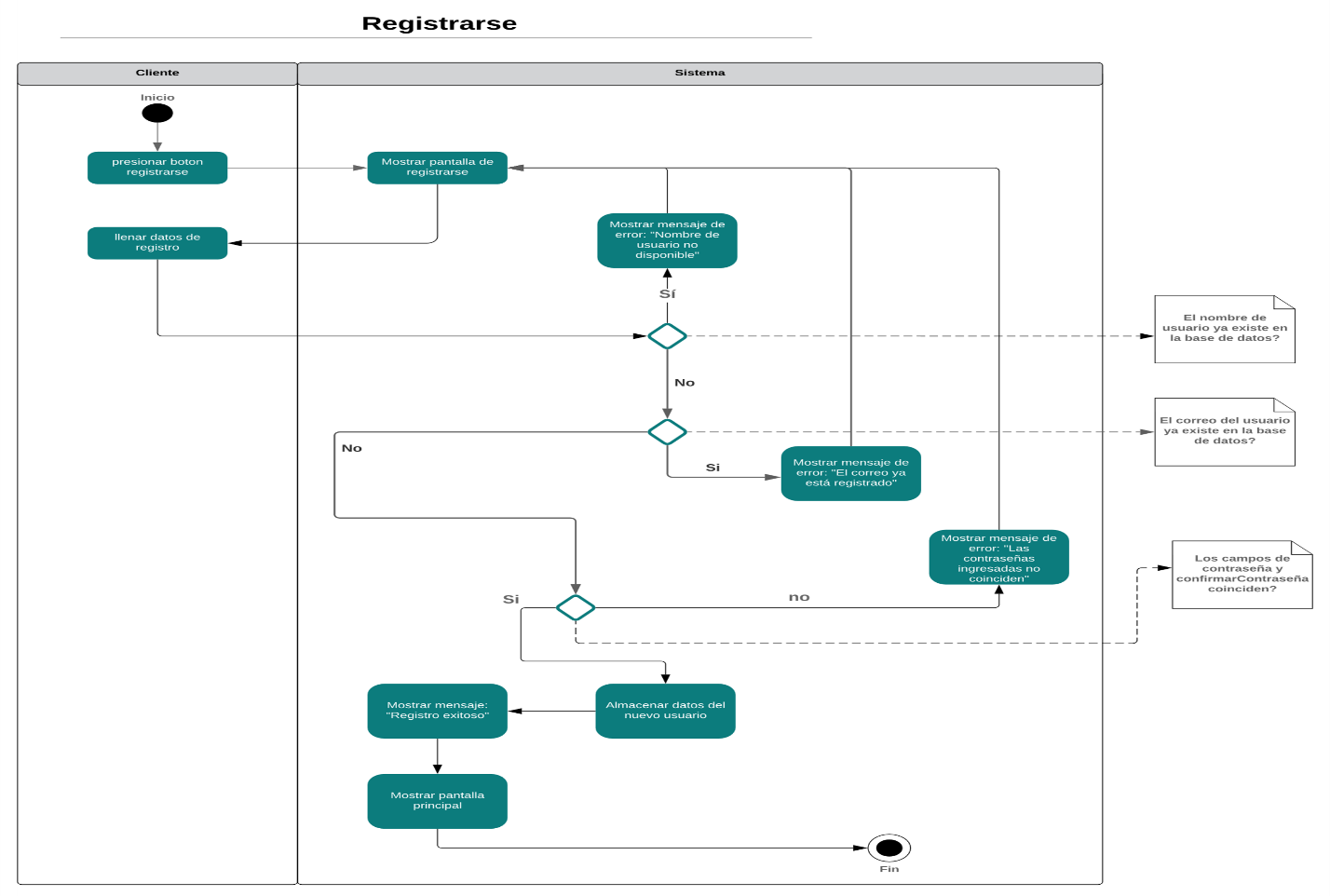


Ilustración 5 Diagrama de actividad de registro de usuario

En este diagrama se muestra el procedimiento que un usuario debe realizar para crearse una cuenta en PC-DROME, los datos que serán verificados al momento de crear la cuenta serán el nombre de usuario y la contraseña. En el caso del nombre de usuario se verificará no esté registrado y la contraseña se verificará que coincida con un campo que el usuario debe llenar con la misma contraseña.

**Armar PC:**

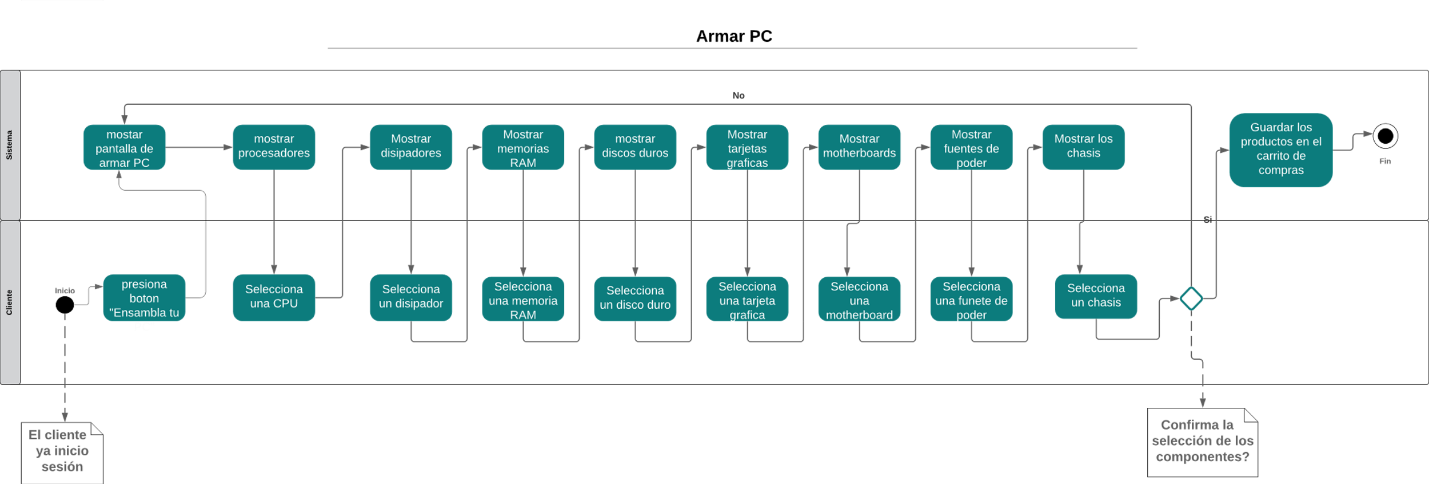


Ilustración 6 Diagrama de actividad de armar PC

En este diagrama se muestra como el usuario va seleccionando los componentes de su computador a medida que el sistema va cambiando los productos que aparecen en pantalla. Cuando el usuario ya terminó de escoger todas las piezas este decide si confirmar la selección realizada, en caso de negarla se le devolverá a la pantalla del principio y se le mostrarán nuevamente los procesadores, y en caso de confirmar los productos, estos se enviarán al carrito de compras.

**Calificar producto:**

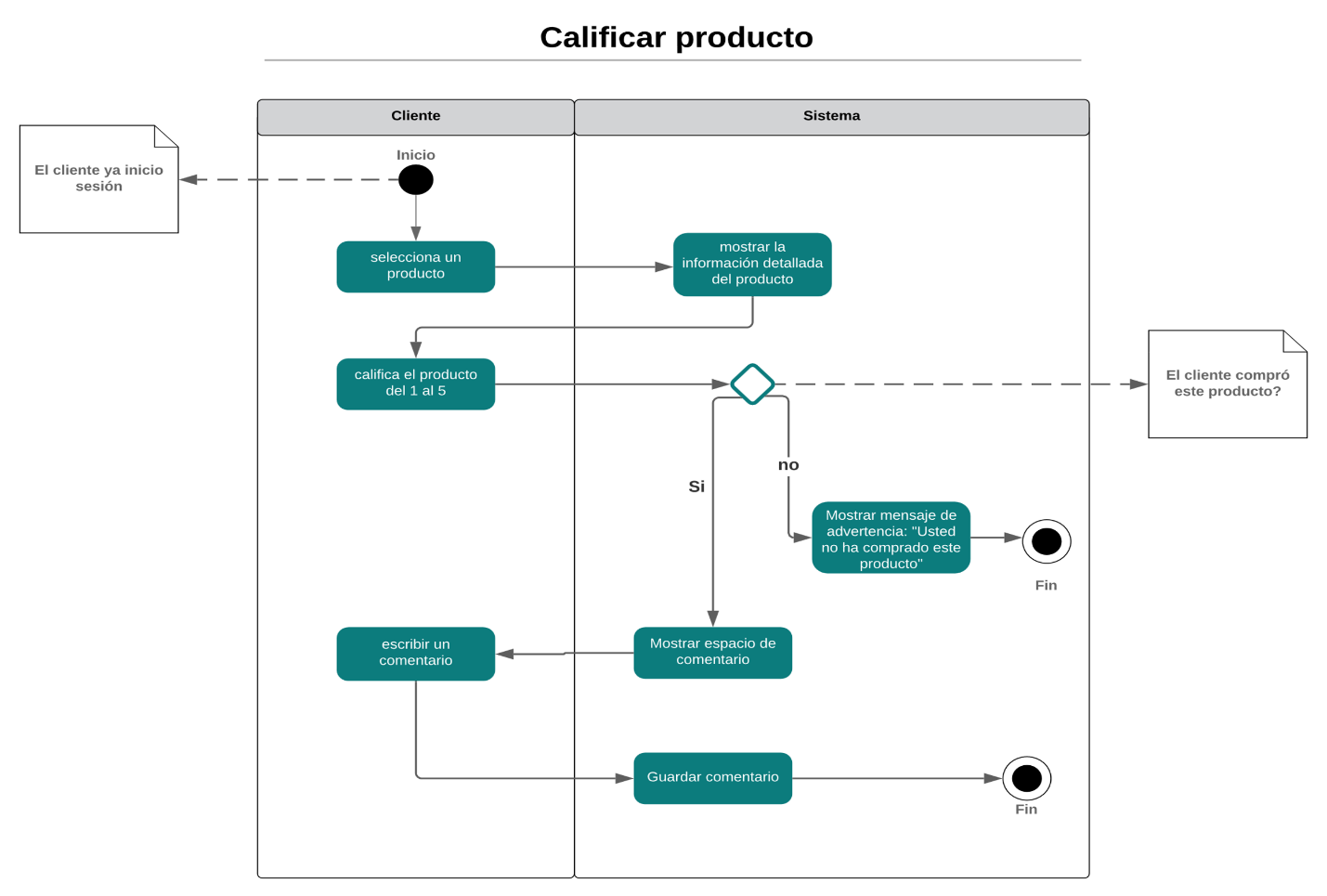


Ilustración 7 Diagrama de actividad de calificar un producto

En este diagrama se muestra como un cliente califica y comenta un producto. Se verifica que el cliente haya comprado el producto para que su calificación sea válida, si este no lo ha comprado entonces se cancela la calificación y se notifica al usuario, en caso de que la haya comprado el cliente puede continuar y escribir un comentario.

# **Diseño Detallado**

## **Estructura del sistema**

Debido a que la arquitectura de PC-DROME está constituida por capas, entonces el diagrama de clases se presenta en tres capas diferentes, la capa de presentación, capa de repositorio y la capa de núcleo.

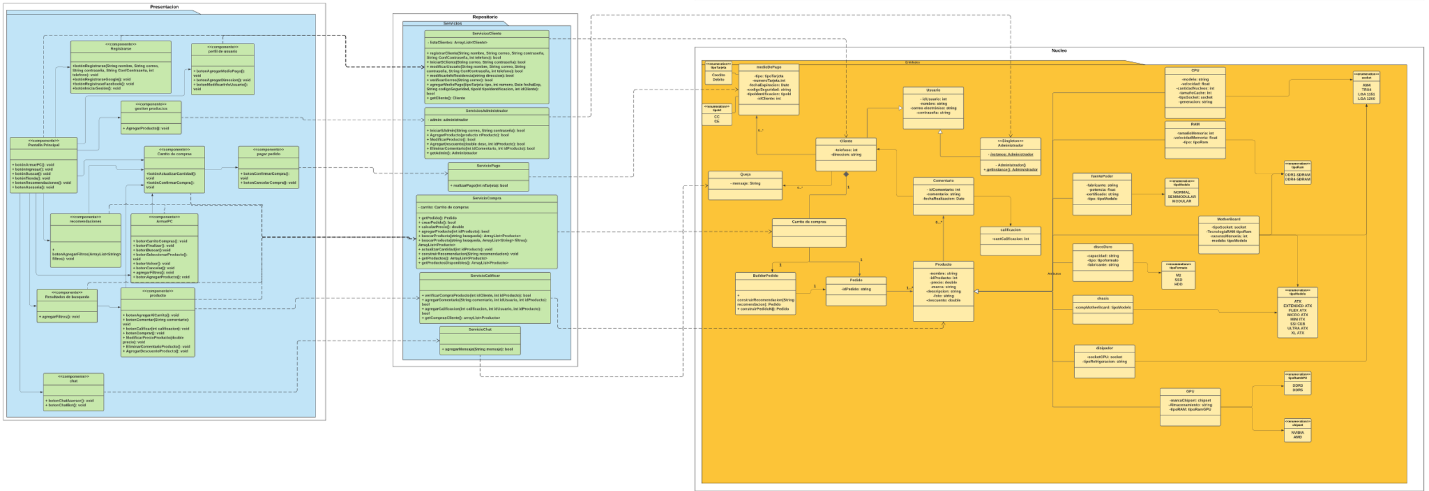


Ilustración 8 Diagrama de clases

Para ver la imagen completa del diagrama de clases, ver anexo #1.

**Capa de presentación:** En esta capa se presentan todos los componentes que conforman la página web. Cada componente posee métodos que ejecutaran una acción que será mostrada en el propio componente que este en pantalla, es decir, no habrá métodos que nos lleven a otros componentes.

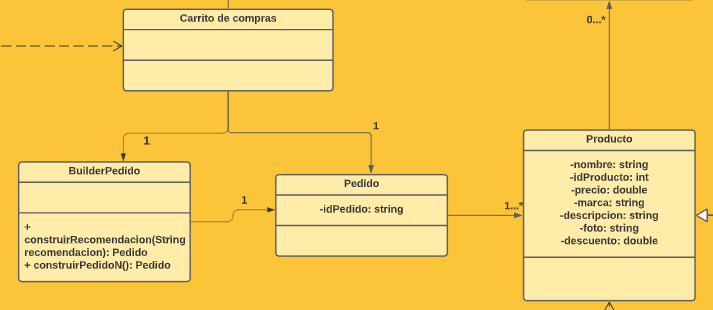
**Capa de repositorio:** En la capa de repositorio se encuentran todos los servicios que harán la conexión entre las capas de presentación y núcleo ejecutando las historias de usuario indicadas en el documento anterior. Algunos de los servicios presentados poseen métodos get cuya función es la de conectarse directamente con la base de datos de FireBase para extraer los datos que sean necesitados para poder ejecutar el servicio.

**Capa de núcleo:** En la capa de núcleo se encuentran las clases que se utilizaran para representar la lógica de negocio de PC-DROME son las siguientes:

1. **Usuario:** Clase padre de los usuarios que van a usar la página web, este usuario va a estar compuesto por un id, un nombre, un correo electrónico y una contraseña.
2. **Cliente:** Clase que hereda de usuario. Los clientes serán los que utilizarán la página con el fin de comprar, este añade los atributos de teléfono y dirección.
3. **Administrador:** Clase que hereda de usuario. El administrador puede navegar por la página web y además tendrá la función de añadir y modificar productos y también eliminar comentarios.
4. **MedioDePago:** Esta clase representa la tarjeta que utilizará el cliente para pagar los productos que comprará. Los atributos son el tipo de la tarjeta, número de tarjeta, fecha de expiración, código de seguridad, tipo de identificación y el número de identificación.
5. **CarritoDeCompras:** Esta clase tiene como función almacenar el pedido que hará el cliente.
6. **Pedido:** Esta clase almacena los productos que el cliente ira seleccionando. Este pedido puede ser identificado por un número id.
7. **BuilderPedido:** Posee el mismo propósito que Pedido pero este Builder se usara únicamente cuando el cliente seleccione las piezas recomendadas por PCDROME o seleccione un computador fabricado por los filtros creando un pedido con los productos recomendados.
8. **Producto:** Es la clase padre de todos los componentes que conforman una computadora y otros accesorios disponibles en PC-DROME. Está compuesto por nombre, un id, el precio, la marca, una descripción, una foto y un descuento.
9. **Comentario:** Los comentarios hacen parte de los productos, estos son las opiniones que dejan los clientes sobre el producto que hayan comprado. Este posee una id, el comentario y una fecha de realización.
10. **Calificación:** Las calificaciones hacen parte de los productos, estos son las valoraciones que los clientes le dan al producto que hayan comprado

Las clases de la capa de núcleo no poseen métodos ya que estos van a ser utilizados únicamente para almacenar la información proveniente de la base de datos.

**BuilderPedido:**



Como patrón GOF hemos diseñado un builder de pedido, el cual se encargara de construir un pedido (computadora completa) que con anterioridad le hemos recomendado al cliente, así que si alguna de esas recomendaciones le llama la atención al cliente y selecciona para adquirirla, se hará uso de esta builder para que construya por completo el PC y quede agregado en el carrito de compras y solo tenga que pagarlo.

## **Comportamiento del sistema**

**Diagrama de secuencia de historia de usuario Registrarse:**

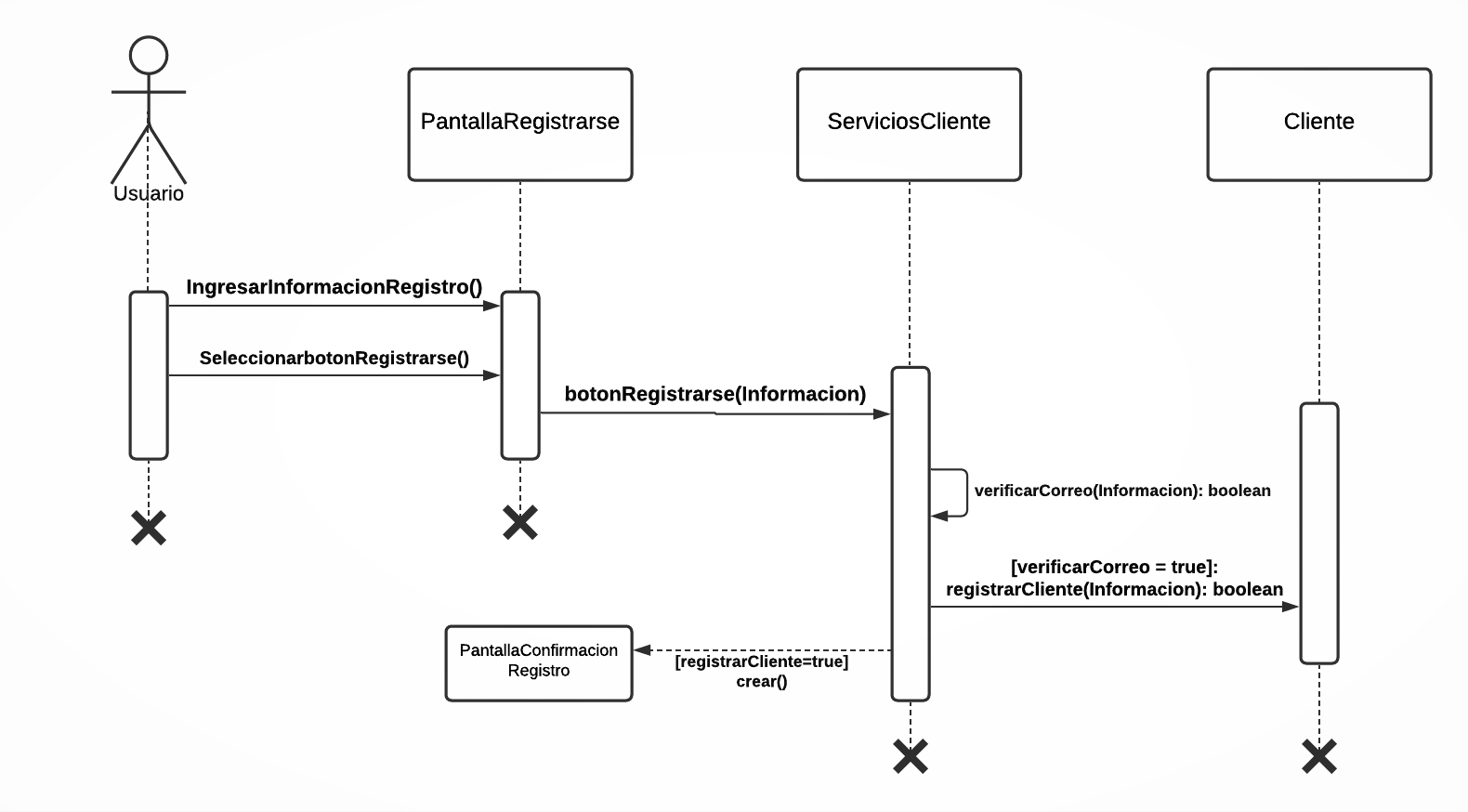


Ilustración 9 Diagrama de secuencia de historia de usuario Registrarse

**Diagrama de secuencia de historia de usuario Armar PC:**

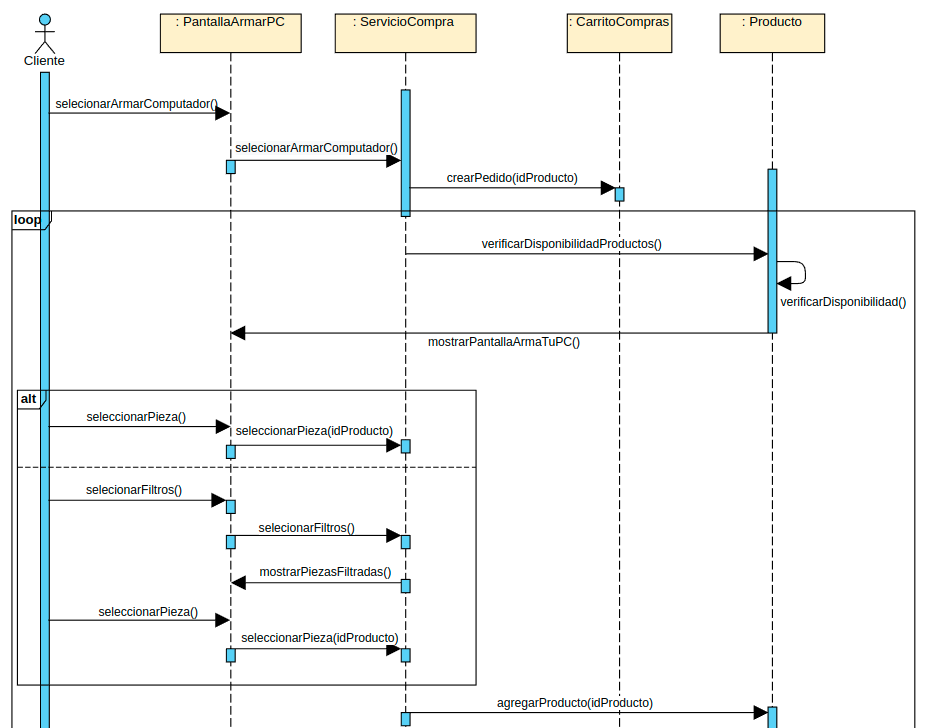


Ilustración 10 Diagrama de secuencia de historia de usuario Armar PC parte 1

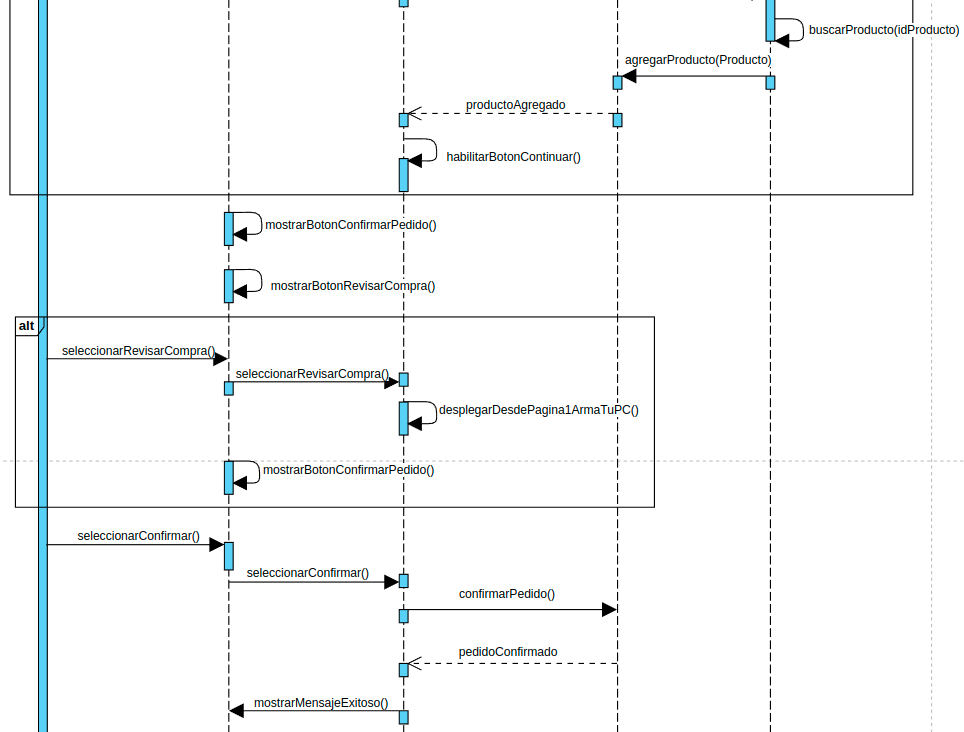


Ilustración 11 Diagrama de secuencia de historia de usuario Armar PC parte 2

## **Persistencia**

Para el almacenamiento de los productos utilizaremos archivos JSON para crear las colecciones, estas colecciones serán las entidades de nuestro sistema, como Cliente o producto. Estas colecciones estarán compuestas por documentos que serán los productos que se ofrecerán en la página, además estos documentos estarán constituidos por sus atributos.

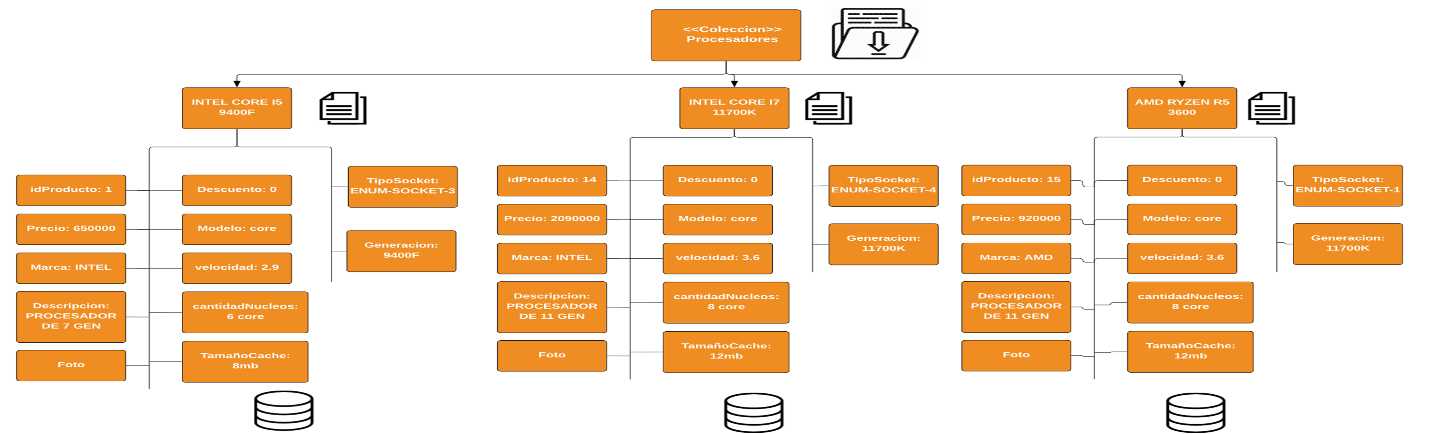
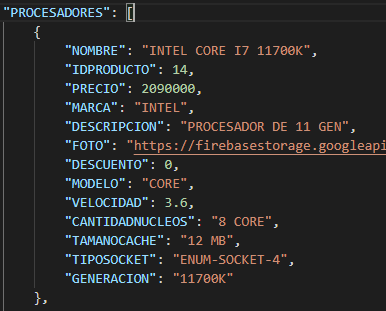


Ilustración 12 Árbol de colecciones de procesador

En la ilustración 12 se muestra cómo va a estar diseñado los objetos de la colección de procesadores. A continuación, se presentan los productos en formato JSON donde cabe destacar que no se puede representar la herencia mostrada en el diagrama de clases por lo que desde el atributo nombre hasta descuento son parte de la entidad producto, los atributos siguientes corresponderán a componentes específicos.



Atributos:

Nombre: Nombre del producto.

idProducto: Número de identificación del producto.

Precio: El valor del producto.

Marca: Fabricante del producto.

Descripción: Información básica sobre el producto.

Foto: Url que lleva a la foto del producto.

Descuento: El descuento aplicado sobre el producto.

Modelo: Modelo del producto.

Velocidad: Cantidad de operaciones por segundo que hace el procesador.

CantidadNucleos: Numero de núcleos del procesador.

TamañoCache: Almacenamiento cache del procesador.

TipoSocket: tipo de ranura que posee el procesador.

Generación: Numero de generación del procesador.

Para ver la descripción de los demás componentes ver el anexo #2.

## 

## **Interfaz de usuario**

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

# **Reporte Gerencial.**

Para esta entrega, se planteó que el día 20 de abril se realizará la reunión de Sprint Review. Se esperaba cumplir satisfactoriamente con 122 puntos de historia, equivalentes a las historias de registro, ingreso y armar pc. Sin embargo, la ejecución de este Sprint fue, entre otras cosas, el resultado de una mala comunicación, poca gestión del tiempo y la falta de capacitación en ciertas tecnologías por parte de todo el equipo. Al final, llegamos a esta parte con puntos de historia realizados en cero.

Ahora bien, planteamos una solución que permita llevar a cabo oportunamente el proyecto para la entrega final. En primera instancia, todo el equipo se reunirá para replantear los compromisos, recopilar las inquietudes y mayores sugerencias que tengamos, y, además, definir los puntos de historia para los siguientes Sprints. Porque si, tendremos que realizar dos Sprints.

# **Anexos**

1. [Diagrama](https://github.com/FundIngSoft202110/PC-Drome/blob/main/SegundaEntrega/Diagrama%20de%20clases%20PCDROME.svg) de clases.svg.
2. [Descripcion](https://github.com/FundIngSoft202110/PC-Drome/blob/main/SegundaEntrega/descripcion%20de%20atributos.xlsx) de atributos.xlxs
3. Firestore\_database\_[PCDROME](https://github.com/FundIngSoft202110/PC-Drome/blob/main/SegundaEntrega/firestore_database_PCDROME.json).json

# **Referencias**

*Diagrama de despliegue. (2018, agosto 24). Recuperado el 18 de abril de 2021, obtenido de:* [*https://diagramasuml.com/despliegue/*](https://diagramasuml.com/despliegue/)

*¿Que es Angular y para que sirve? - Victor Robles. (2017, agosto 5). Recuperado el 18 de abril de 2021, obtenido de:* [*https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/*](https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/)

*Cardona, M. P. (2016). Firebase, qué es y para qué sirve la plataforma de Google. Thinking for Innovation. Recuperado de* [*https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarroladores-google-seo-sem/*](https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarroladores-google-seo-sem/)

*Firebase Realtime Database. Recuperado el 18 de abril de 2021, obtenido de:* [*https://firebase.google.com/docs/database?hl=es*](https://firebase.google.com/docs/database?hl=es)

*Requisitos del sistema del navegador Chrome. Recuperado el 18 de abril de 2021, obtenido de:* [*https://support.google.com/chrome/a/answer/7100626?hl=es*](https://support.google.com/chrome/a/answer/7100626?hl=es)

*Requisitos del sistema Mozilla Firefox. Recuperado el 18 de abril de 2021, obtenido de:* [*https://es.brandingandwebsites.com/mozilla-firefox-system-requirements*](https://es.brandingandwebsites.com/mozilla-firefox-system-requirements)

*Sabán, A. (2021). Microsoft Edge llega a Linux: así puedes descargarlo e instalarlo. Genbeta.com. Recuperado18 de abril de 2021, obtenido de:* [*https://www.genbeta.com/navegadores/microsoft-edge-llega-a-linux-asi-puedes-descargarlo-e-instalarlo*](https://www.genbeta.com/navegadores/microsoft-edge-llega-a-linux-asi-puedes-descargarlo-e-instalarlo)*.*

*Edge, R. (2021). Requisitos técnicos de Edge para PC. Vandal. Recuperado el 18 de Abril de 2021, obtenido de:* [*https://vandal.elespanol.com/requisitos/pc/edge/14819*](https://vandal.elespanol.com/requisitos/pc/edge/14819)*.*

*Sistemas operativos admitidos con Microsoft Edge. Docs.microsoft.com. (2021). Recuperado el 18 de Abril de 2021, obtenido de:* [*https://docs.microsoft.com/es-es/deployedge/microsoft-edge-supported-operating-systems*](https://docs.microsoft.com/es-es/deployedge/microsoft-edge-supported-operating-systems)*.*

*Chrome, R. (2021). Requisitos técnicos de Chrome para PC. Vandal. Recuperado el 18 de Abril de 2021, obtenido de:* [*https://vandal.elespanol.com/requisitos/pc/chrome/2790#p-13*](https://vandal.elespanol.com/requisitos/pc/chrome/2790#p-13)*.*

*Firebase Authentication. Firebase. (2021). Recuperado el 18 de Abril de 2021, obtenido de:* [*https://firebase.google.com/docs/auth?hl=es*](https://firebase.google.com/docs/auth?hl=es)*.*

*Firebase Hosting. Firebase. (2021). Recuperado el 18 de Abril de 2021, obtenido de:* [*https://firebase.google.com/docs/hosting?hl=es*](https://firebase.google.com/docs/hosting?hl=es)*.*

*Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue. Sparxsystems.com.ar. (2021). Recuperado el 20 de Abril de 2021, obtenido de:* [*http://www.sparxsystems.com.ar/resources/tutorial/uml2\_deploymentdiagram.php*](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.php)