**Universidad de La Punta**

**U.L.P**

**Laboratorio 3**

**[VENTAS CON DELIVERY]**

**[AGUERO, Miguel Alfredo]**

**[Legajo]**

**Fecha: [10/05/2021]**

**Docente/Tutor: [Luis MERCADO, Mariano LUZZA]**

Tabla de contenido

[Introducción: 3](#_Toc71304709)

[Justificación 3](#_Toc71304710)

[Objetivo general del proyecto 3](#_Toc71304711)

[Objetivos específicos del proyecto 4](#_Toc71304712)

[Objetivo general del sistema 4](#_Toc71304713)

[Límite 4](#_Toc71304714)

[Alcance 4](#_Toc71304715)

[No Contempla 4](#_Toc71304716)

[T.I.C. (Tecnología de la Información y Comunicación): 5](#_Toc71304717)

[Competencia: 6](#_Toc71304718)

[Listado de requerimientos funcionales: 7](#_Toc71304719)

[Listado de requerimientos no funcionales: 7](#_Toc71304720)

[Desarrollo del Prototipo: 8](#_Toc71304721)

[Análisis y Diseño: 8](#_Toc71304722)

[Diagrama de Casos de Uso: 9](#_Toc71304723)

[Diagrama Entidad Relación: 12](#_Toc71304724)

[Interfaz Gráfica: 12](#_Toc71304725)

[BIBLIOGRAFIA: 15](#_Toc71304726)

[ANEXO I: 16](#_Toc71304727)

[Formularios en Soporte papel 16](#_Toc71304728)

# Introducción:

Este proyecto propone una solución que reúne a vendedores y clientes a través de una aplicación y un servicio de delivery.

Además, pretende abarcar un nicho de mercado desatendido por empresas como “Pedidos Ya” y “mercado libre”.

# Justificación

La localidad pequeña en la que vivo no tiene el servicio de “Pedidos Ya”, que es el modelo que inspira este proyecto. Esta situación se ha evidenciado en el contexto de pandemia que nos toca.

Han surgido muchos emprendimientos que venden sus productos y los entregan por medio de servicio de delivery, debido a las restricciones vigentes y al aislamiento/distanciamiento social.

En este marco de situación, este proyecto pretender ser una herramienta que aporte organización al servicio de delivery generando una fuente de trabajo mas ordenada, mejore las ventas de los comercios y preste un servicio al cliente.

# Objetivo general del proyecto

Diseñar, construir e implementar un sistema de información transaccional para la venta de productos, que pueda ser utilizada por vendedores, clientes y delivery’s.

# Objetivos específicos del proyecto

* Diseñar e implementar una base de datos relacional que de solución a los requerimientos del proyecto.
* Diseñar e implementar una solución backend que provea los recursos para ser accedidos desde una aplicación web y desde una aplicación móvil.
* Diseñar e implementar una aplicación web para que los vendedores publiquen sus productos.
* Diseñar e implementar una aplicación móvil para que los clientes puedan buscar y comprar los productos, pagar la compra y calificar al vendedor y al delivery.
* Diseñar e implementar una aplicación móvil para que los delivery’s se notifiquen de los pedidos a entregar y puedan calificar al cliente.

# Objetivo general del sistema

Gestionar la publicación, la compra y la entrega de productos entre vendedores, clientes y delivery’s respectivamente.

# Límite

Los límites del sistema están enmarcados desde el registro de un vendedor y sus publicaciones, pasando por la compra por parte del cliente, hasta el registro de entrega del pedido a cargo del delivery. La calificación de los participantes de una transacción (vendedor, delivery, cliente) es el último paso del proceso.

# Alcance

Los procesos que van a ser soportados dentro del sistema son:

1. Registro de un vendedor.
2. Registro de un delivery.
3. Registro de la publicación de un producto.
4. Actualización de una publicación.
5. **Búsqueda de un producto por nombre y/o rango de precios y/o proveedor.**
6. **Registro de un cliente.**
7. **Registrar pedido de uno o varios productos.**
8. **Listar pedidos.**
9. Pago de un pedido.
10. Asignar un pedido a un delivery
11. Registro de la entrega de un pedido.
12. Calificar a un vendedor, a un cliente y a un delivery.

# No Contempla

El sistema no contempla la facturación de las ventas. No se incluye la publicación de la solución en un hosting.

## T.I.C. (Tecnología de la Información y Comunicación):

El sistema a desarrollar utilizará la arquitectura cliente-servidor. Los lenguajes para el desarrollo serán Java para la aplicación móvil y c# para la aplicación web, Ambas aplicaciones tendrán acceso al repositorio de datos a través de una aplicación api que contendrá la lógica de negocio a tal fin.

El motor de base de datos será sql server en su versión sqllocaldb para la etapa de desarrollo.

## Competencia:

Ademas de los conocidos servicios de Mercado Libre y Pedidos Ya que cubren todas las funcionalidades planteadas para este proyecto, existen otras aplicaciones como por ejemplo PedisoFREE (<https://www.pedidosfree.com.ar/>) que brinda una solución muy parecida pero que solo lo hace en el rubro “comidas”.

La ventaja de la solución propuesta en este proyecto con respecto la competencia, está dada por un nicho de mercado desatendido para el caso de Mercado Libre y Pedidos Ya. Para el caso de aplicaciones como PedidosFREE, la diferencia en favor de este proyecto, está dada por la variedad de productos que permitiría publicar para la venta.

## Listado de requerimientos funcionales:

|  |  |
| --- | --- |
| ID | Nombre |
| RF001 | Gestión de usuarios. |
| RF002 | Gestión de publicaciones. |
| RF003 | Gestión de pedidos |
| RF004 | Gestión de pagos |
| RF005 | Gestión de calificaciones |

## Listado de requerimientos no funcionales:

|  |  |  |
| --- | --- | --- |
| ID | Nombre | Tipo |
| RNF001 | Definir perfiles de usuario (vendedor, cliente, delivery) y asignar permisos de acceso correspondiente. | Seguridad |
| RNF002 | Proporcionar una interfaz amigable e intuitiva. | Usabilidad |
| RNF003 | El sistema utilizará la arquitectura cliente-servidor | Implementación |
| RNF004 | Se trabajará con software bajo licencia open source. | Implementación |
| RNF005 | La aplicación trabajará sobre plataforma web y móvil para android. | Portabilidad |
| RNF006 | Almacenar los datos en una base de datos relacional, SQLServer. | Implementación |
| RNF007 | Operar en internet. | Implementación |

# Desarrollo del Prototipo:

## Análisis y Diseño:

El sistema a desarrollar tendrá una arquitectura cliente-servidor. El sistema servidor será instalado en un hosting de internet. La red de internet será la vía de comunicación entre el cliente y el servidor. El sistema cliente tendrá dos versiones, una aplicación web y una aplicación móvil.

Del lado del servidor habrá una base de datos relacional sobre el motor de base de datos de sql server. En la capa de negocio, también del lado del servidor, funcionará como interface, una aplicación de tipo api rest, programada siguiendo el patrón MVC. La tecnología a utilizar para el desarrollo de la api rest será la plataforma de .Net core en su versión 5. El lenguaje a utilizar será c#.

Del dado del cliente, habrá una aplicación móvil par que consuma los servicios de la api rest y que será utilizada por los usuarios/cliente y los usuario/delivery del sistema. También habrá una aplicación web que también consumirá los servicios de la api rest y estará destinada a los usuarios/vendedores del sistema.

## Diagrama de Casos de Uso:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nro. UC | Rq. Fc. Asoc. | Nombre | Prioridad | Complejidad |
| UC001 | RF001 | Registrar Usuario | Alta | Alta |
| UC002 | RF001 | Iniciar sesión | Alta | Baja |
| UC003 | RF002 | Registrar publicación | Alta | Alta |
| UC004 | RF002 | Buscar publicación | Alta | Alta |
| UC005 | RF003 | Generar pedido | Alta | Media |
| UC006 | RF004 | Generar Pago del pedido | Alta | Alta |
| UC007 | RF003 | Listar pedidos | Alta | Baja |
| UC008 | RF003 | Asignar/Tomar pedido | Alta | Baja |
| UC009 | RF003 | Entregar pedido | Alta | Baja |
| UC010 | RF005 | Calificar al cliente | Media | Media |
| UC011 | RF005 | Calificar al vendedor y delivery | Media | Media |

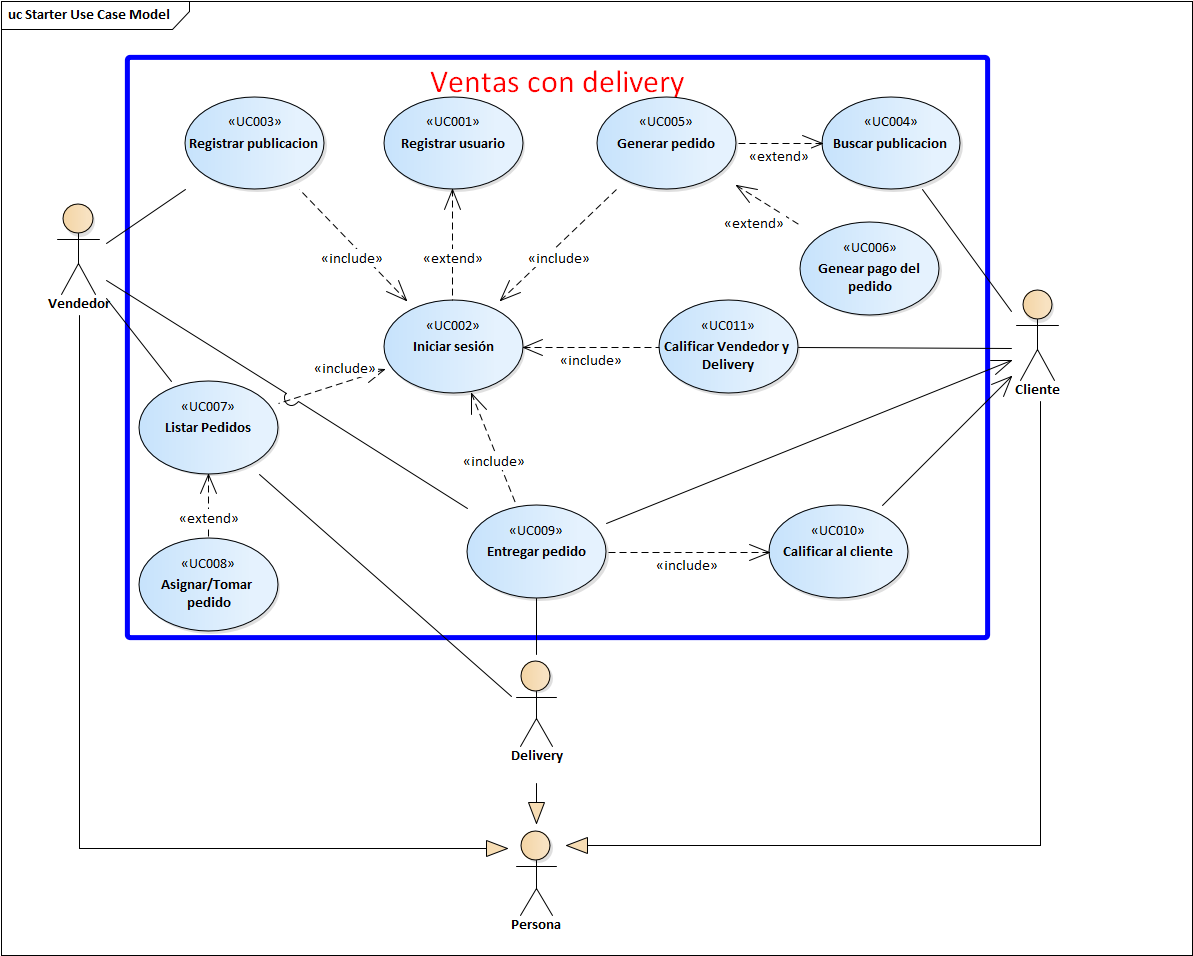
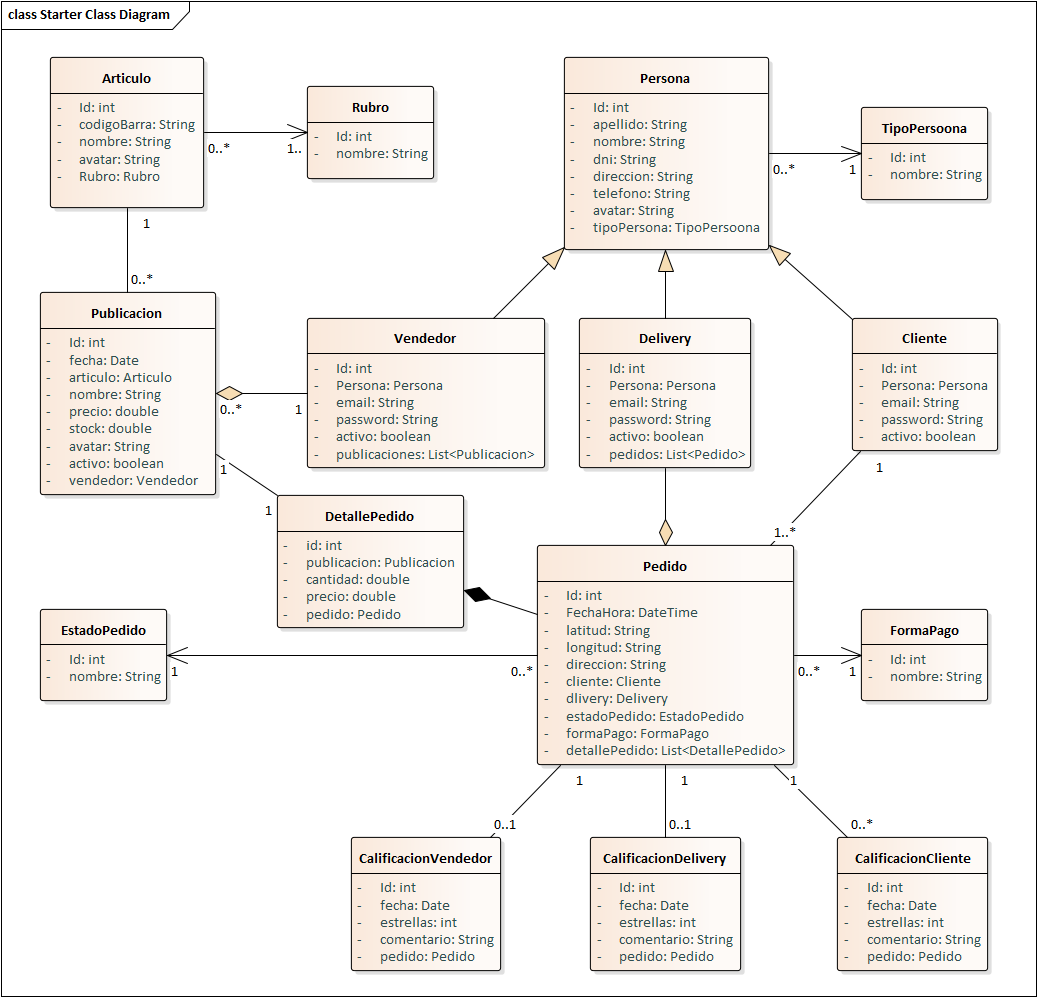


Diagrama de Clases:



## Diagrama Entidad Relación:

****

## Interfaz Gráfica: