**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**ESCUELA PROFESIONAL ACADÉMICA DE INGENIERÍA DE SOFTWARE**

****

**SISTEMA DE PUNTO DE VENTA PARA LICORERIA - eLiquor**

**DOCUMENTO DE ARQUITECTURA DE SOFTWARE**

**Docente: LENIS ROSSI WONG PORTILLO**

**Curso: Gestión de la configuración de software**

**Sección: 1**

**Grupo: 2**

**Integrantes:**

* **Tejeda Echegaray, Yosmar Aldair 20200298**
* **Figueroa Garay, Jhoan Joseph 20200261**
* **Paz Anchayhua, Gerardo Ruben 20200080**
* **Alvarez Huarsaya, Richard Saul 20200047**
* **Ortiz Urbai, Sebastian 20200056**
* **Chavez Campos, Jean Pier 20200071**
* **Aznaran Cabrera, Gerson Eduardo 20200247**
* **Tuyo Acero, Gustavo Alonso 20200302**

**Semestre 2022 - 2**

**HISTORIAL DE VERSIONES**

| **Versión** | **Autores** | **Descripción** | **Fecha** |
| --- | --- | --- | --- |
| 1.0 | * Gustavo Alonso Tuyo Acero * Chavez Campos, Jean Pier * Figueroa Garay, Jhoan Joseph * Aznaran Cabrera, Gerson Eduardo * Ortiz Urbai, Sebastian | Se terminó la introducción, los objetivos de negocio, la visión de la solución, el alcance, el contexto del sistema, los drivers y las consideraciones generales de la arquitectura. | 23/10/2022 |

1. **Introducción**
   1. **Propósito**

El presente documento tiene como intención presentar la arquitectura que hemos definido para el Sistema de Punto de Venta eLiquor. Para ello, se describirán aspectos como las tecnologías a usar, las capas arquitectónicas del sistema, las entidades diseñadas que se usarán en cada módulo y las interfaces de usuario relacionadas a cada uno de ellas.

* 1. **Alcance**
* Definir una visión general de la arquitectura de nuestro sistema.
* Señalar las tecnologías a utilizar para el desarrollo del sistema.
* Definir los módulos del sistema y las entidades relacionadas a cada uno de ellos.
  1. **Objetivos**
* Establecer las tecnologías principales que los desarrolladores del equipo usarán y servirán como herramientas de apoyo para construir el sistema.
* Formar la documentación base para que en posteriores revisiones que los analistas y arquitectos de software realicen puedan usar como referencia.

1. **Objetivos de Negocio**

| **ID** | **Descripción del objetivo del Negocio** |
| --- | --- |
| **ON-1** | Incrementar en 50% las ventas de licores mediante la implantación de un sistema de punto de venta. |
| **ON-2** | Colectar datos con el fin de mejorar continuamente el servicio 4 meses después de su puesta en marcha. |
| **ON-3** | Liberar el sistema antes del 31 de diciembre de 2022 |

1. **VISIÓN DE LA SOLUCIÓN**

**3.1 Fase de visión**

eLiquor será un sistema moderno de venta de licores. Permitirá realizar ventas, consultas de estados de productos, estados de ventas, etc.

**3.2 Características del sistema**

| ID | Título | Descripción | Prioridad | Objetivo del Negocio Asociado |
| --- | --- | --- | --- | --- |
| CAR-01 | Acceso de la aplicación cajero o administrador | Como **usuario general**, quiero poder acceder a una interfaz que me permita el ingreso de ciertas credenciales (usuario y contraseña) que validen mi autorización, para poder acceder al sistema. | Alta | ON-1 |
| CAR-02 | Registro y control de usuarios | Como **usuario administrador** quiero poder gestionar la actividad y privilegios de los usuarios de rol empleado, para poder llevar un buen seguimiento de las labores del personal. | Alta | ON-1 |
| CAR-03 | Registro de ventas y devoluciones | Como **usuario general** quiero guardar todos los datos en el sistema, que conllevan el momento de realizar una venta y poder generar recibo rápidamente, para poder tener un seguimiento del flujo de dinero e inventario del negocio. | Alta | ON-1 |
| CAR-04 | Registro de clientes | Como **usuario general** quiero poder registrar los datos de nuevos clientes para poder posteriormente ofrecerles más beneficios. | Alta | ON-1 |
| CAR-05 | Gestión de inventarios | Como **usuario general** quiero controlar (agregar, eliminar, etc) el stock de los productos que se ofrece en el negocio como también poder categorizarlos y agregarles toda la información respectiva con el fin de tener un mejor control del inventario. | Alta | ON-1 |
| CAR-06 | Gestión de proveedores | Como **usuario general** quiero poder consultar y administrar todos los proveedores con los que trabajaremos para poder tener un mejor seguimiento de los pedidos que se solicitan al proveedor. | Alta | ON-1 |
| CAR-07 | Generación de reportes | Como **usuario administrador** quiero generar el reporte de las ventas del día por empleado, además generar el reporte de mi inventario, para poder facilitar el análisis del nivel de éxito del negocio. | Alta | ON-02 |
| CAR-08 | Botón de ayuda de visualización del programa | Como **usuario general** quiero poder tener una guía en la propia aplicación. | Media | ON-1 |

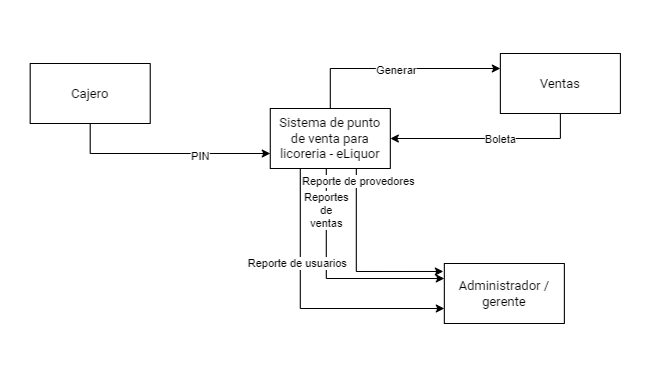
1. **ALCANCE**

| **Número de entrega** | **Tema principal** | **ID de características a incluir** |
| --- | --- | --- |
| 1.0 | Funcionalidad Básica | CAR-01, CAR-02 |
| 2.0 | Gestión de inventarios, registro de ventas y generación de reportes | CAR-03, CAR-07, CAR-05, |
| 3.0 | Registro de clientes, gestión de proveedores y botón de ayuda. | CAR-04, CAR-06, CAR-08 |

1. **CONTEXTO DEL SISTEMA**
   1. **Interesados**

| **Nombre** | **Descripción** | **Responsabilidades** |
| --- | --- | --- |
| Juan Pérez | Director de operación. | * Aprobar la visión y el alcance del sistema. * Proporcionar acceso a instalaciones y a procedimientos existentes. * Aprobar entregas del proyecto. |
| Martha Ramírez | Representante de usuarios | * Proveer y validar requerimientos de interacción de usuarios finales con el sistema. * Validar prototipos. * Mostrar la manera en que se llevan a cabo las tareas que serán automatizadas. |
| Ramón Salas | Administrador del sistema | Proporcionar y validar requerimientos relacionados con la administración y las necesidades de recolección de datos del sistema |
| Sebastián Ortiz | Líder de proyecto | Coordinar el proyecto y representar al equipo de desarrollo. |

* 1. **Diagrama de contexto**



1. **Drivers**
   1. **Drivers funcionales**
      1. **Modelo de Requisitos Funcionales:**

**(Modelado)**

| REQUERIMIENTOS FUNCIONALES | | |
| --- | --- | --- |
| ID | NOMBRE | DESCRIPCIÓN |
| RF\_01 | Acceder a la aplicación | Permite al usuario acceder al sistema usando credenciales |
| RF\_02 | Registrar usuarios | Permite al usuario crear nuevos usuarios y seleccionar los permisos de cada uno. |
| RF\_03 | Gestionar inventario | Permite al usuario crear, modificar o eliminar productos. |
| RF\_04 | Registrar ventas | Permite al usuario registrar dato de una venta como productos comprados, dinero ingresado, etc. |
| RF\_05 | Generar reportes | Permite al usuario acceder a la información de venta generada y exportarla en los formatos que este desee. |
| RF\_06 | Registrar clientes | Permite al usuario agregar, modificar o eliminar clientes. |
| RF\_07 | Gestionar proveedores | Permite al usuario agregar, modificar o eliminar proveedores. |
| RF\_08 | Visualizar ayuda de programa | Permite al usuario visualizar un tutorial para que aprenda a utilizar el sistema. |

* + 1. **Elección de Casos de Usos Primarios**

| **ID** | **Criterio de elección** |
| --- | --- |
| RF-04 | La realización de ventas es la razón primaria de este sistema (ON-01). |
| RF-02 | Es necesario registrar los usuarios quienes son los que realizarán la venta. |
| RF-06 | Es necesario registrar los clientes a quienes se les realiza la venta. |

* 1. **Drivers de atributos de Calidad**

| **ID** | **Categoría** | **Escenario** |
| --- | --- | --- |
| **EAC-01** | Seguridad | * El sistema es una aplicación de escritorio, por lo cual es más seguro. * Solo se podrá acceder a la aplicación los miembros del personal y el administrador que cuentan con un usuario y contraseña únicos otorgados por el administrador |
| **EAC-02** | Disponibilidad | * La aplicación va a estar disponible cada vez que se ejecute ya que es una aplicación de escritorio |
| **EAC-03** | Usabilidad | * La aplicación debe ser de fácil aprendizaje para el usuario. * La aplicación tiene un botón de ayuda en el cual se puede observar todas las operaciones que puede realizar la aplicación. |
| **EAC-04** | Eficiencia | * El tiempo de respuesta de la aplicación debe ser imperceptible para el usuario |
| **EAC-05** | Mantenibilidad | * La aplicación es capaz de evolucionar, para poder realizar cambios de manera eficiente y efectiva. * La documentación realizada del programa hace accesible a la mantenibilidad. |

* 1. **Drivers de Restricciones**

| Tipo de restricción | Descripción |
| --- | --- |
| Del Cliente | * El primer sprint debe realizarse hasta el 31/10/2022 * El segundo sprint debe realizarse hasta el 30/11/2022 |
| De la organización de desarrollo | Los ingenieros de los que se dispone están familiarizados con los lenguajes de java y sql. |

1. **Primera iteración: estructuración general del sistema**

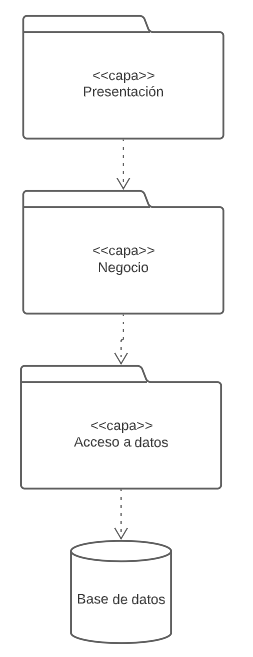
| **Resumen primera iteración** | | |
| --- | --- | --- |
| Elemento a descomponer | El sistema (iteración inicial). | |
| Drivers elegidos para la iteración | Para la segunda iteración los drives elegidos son: RF\_01 y RF\_02. Estos drivers están relacionados al módulo de login y el de registro y control de usuarios descritos en el documento de especificación de requisitos. | |
| **Conceptos de diseño** | **Concepto** | **Justificación** |
|  | Estilo o patrón arquitectónico de capas | Las capas permiten aislar de forma lógica distintas responsabilidades del sistema: los aspectos relacionados con la interacción con el usuario (capa de presentación), el manejo de la lógica de negocio (capa de negocio) y la persistencia de los datos (capa de acceso a datos). |

1. **Consideraciones generales**
   1. **Definición de arquitectura**

* Para que el desarrollo del Sistema de Punto de Venta eLiquor se optó por trabajar en 3 capas (presentación, lógica de negocio y acceso datos).
  1. **Tecnologías utilizadas**

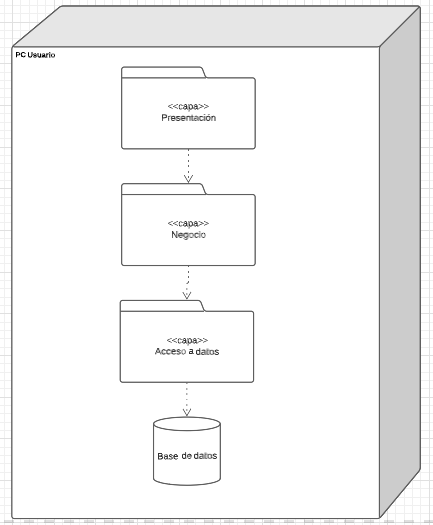
| Tecnología | Descripción | Logo |
| --- | --- | --- |
| Apache NetBeans IDE 15 | Entorno de desarrollo integrado libre que proporciona soporte para lenguajes de programación como Java, PHP, C/C++, entre otros. Nos servirá para la codificación y el desarrollo de la lógica e interfaces gráficas del aplicativo, debido a que nos otorga dichas funcionalidades |  |
| Java 17 | Java es un lenguaje de programación que aplica el paradigma de orientación a objetos, que nos será útil para desarrollar la lógica de nuestro sistema. Además, incluye la librería Swing que nos ayudará con la elaboración de interfaces gráficas. |  |
| MySQL Workbench 8.0 | Es una herramienta de diseño y administración de bases de datos, que nos será útil para la creación y verificación de la base de datos utilizada por el sistema |  |

* 1. **Capas arquitectónicas**
     1. **Perspectiva Lógica**



| Elemento | Responsabilidad | Propiedades |
| --- | --- | --- |
| Capa de presentación | Engloba todos los componentes encargados de recibir interacción por parte del usuario y de mostrarle resultados | Lenguaje: Java |
| Capa de Negocio | Engloba componentes de coordinación que se encargan de llevar a cabo la ejecución de los casos de uso del sistema. | Lenguaje: Java |
| Capa de Acceso a Datos | Engloba componentes encargados de almacenar objetos en una base de datos, ocultando del resto de la aplicación el hecho de que se trata de una base de datos relacional. | Lenguaje: Java y SQL |

7. Perspectiva física: (simbología: UML)



| Elemento | Responsabilidad | Propiedades |
| --- | --- | --- |
| PC Usuario | Computadora personal mediante la cual acceden los usuarios y administradores del sistema | - |

**DISEÑO DE ARQUITECTURA**

**3.1 Primera iteración: estructuración general del sistema**

| **Resumen primera iteración** | | |
| --- | --- | --- |
| Elemento a descomponer | El sistema (iteración inicial). | |
| Drivers elegidos para la iteración | Dado que es la iteración inicial, el arquitecto debe proponer una estructuración general del sistema. Para hacerlo toma en cuenta el conjunto de casos de uso primarios y escenarios de atributos de calidad, así como las siguientes restricciones (véase sección 5.3 del documento de visión): | |
| **Conceptos de diseño** | **Concepto** | **Justificación** |
|  | Estilo o patrón arquitectónico de capas | Las capas permiten aislar de forma lógica distintas responsabilidades del sistema: los aspectos relacionados con la interacción con el usuario (capa de presentación), el manejo de la lógica de negocio (capa de negocio) y la persistencia de los datos (capa de acceso a datos). |

3.1.2 Perspectiva lógica: (simbología: UML)