Proyecto para la flota

Proyecto para empresa de transporte que desea implementar la venta de tiquetes y el manejo de líneas y demás de manera virtual.

Requerimientos Funcionales:

Para tiquetes:

Se entiende por tiquetes, cualquier tipo de venta al cliente ya sean tiquetes, encomiendas y demás entregables al cliente.

|  |  |
| --- | --- |
| REQT1 | Permitir la venta de tiquetes a cualquier línea |
| REQT2 | Permitir la anulación de tiquetes en líneas actuales o ya despachadas. |
| REQT3 | Permitir la creación de pasajeros dado el documento |
| REQT4 | Permitir la edición de tiquetes |
| REQT5 | Permitir la reimpresión de tiquetes |
| REQT6 | Permitir la creación de encomiendas. |
| REQT7 | Ver los tiquetes actuales por línea. |
| REQT8 | Ver los tiquetes totales. |
| REQT9 | Ver un tiquete especifico. |
| REQT10 | Permitir la creación de clientes nuevos. |

Para líneas:

Se entiende por líneas los rodamientos de los buses y los despachos desde cualquier sucursal a cualquier sucursal, el impreso es el despacho o planilla, en esta se ven los tiquetes asociados e información perteneciente al conductor y al vehículo, además a esta planilla se pueden adjuntar vehículos y encomiendas.

|  |  |
| --- | --- |
| REQL1 | Permitir el despacho de líneas |
| REQL2 | Permitir la anulación de despacho de líneas |
| REQL3 | Permitir la creación de líneas de cualquier sucursal a cualquier sucursal |
| REQL4 | Permitir la consulta de líneas desde cualquier sucursal |
| REQL5 | Permitir la consulta de la lista de todos los conductores actuales |
| REQL6 | Añadir viáticos a las líneas que lo necesiten. |
| REQL7 | Permitir la edición de líneas en cualquier estado |

Otros requerimientos:

Para otros requerimientos se tienen en cuenta algunos como el ingreso de usuarios, cierre de caja, consultas internas y demás procesos que no son pertinentes para el cliente o el funcionamiento durante el proceso de ventas.

|  |  |
| --- | --- |
| REQO1 | Permitir el ingreso de sesión de cada taquillero |
| REQO2 | Permitir el ingreso de traslado de fondos de cada taquillero |
| REQO3 | Permitir el cierre de caja de cada taquillero |
| REQO4 | Permitir la consulta de cierres de caja de cada taquillero |
| REQO5 | Permitir la consulta de las finanzas de cada empresa entre cualesquiera fechas. |
| REQO6 | Obtener un consolidado de ventas con una frecuencia dada por el usuario (Ej: Mensual) de cada empresa |
| REQO7 | Obtener un balance de ventas con una frecuencia dada por el usuario (Ej: Mensual)de cada taquillero |
| REQO8 | Permitir la creación de nuevos taquilleros, carros, rutas y demás. |

Diagramas:

Diagrama de casos de uso:

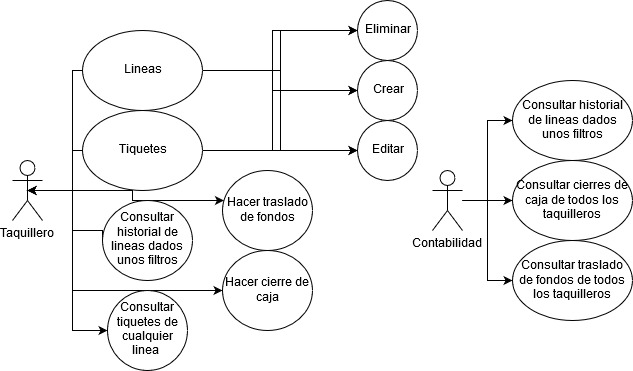


Diagrama de base de datos:

Diagrama de despliegue:

Se usará una arquitectura PENV:

* Base de Datos:
  + Para la base de datos se usara una base de datos relacional, mas específicamente POSTGRESQL, la cual permite de manera eficiente almacenar, editar, consultar y eliminar todos los datos generados en el transcurso de las jornadas de trabajo.
  + Se necesita que esta base este disponible durante todo el trascurso de la jornada de trabajo (3:30AM a 8:00PM), puesto que no está permitido ingresar a las plataformas fuera de los horarios laborables.
  + Esta base cuenta con todas las facilidades para cumplir con las transacciones que se vayan a ejecutar cumpliendo con el esquema ACID.
  + La base de datos tendrá una capacidad para guardar todos los datos generados durante un mes fiscal, pasado este mes estos datos pasaran a ser archivados para eliminarlos de la base de datos y agilizar consultas y acciones sobre la base de datos. Así mismo, los datos archivados van a poder ser consultados en cualquier momento por cualquier usuario que lo requiera.
* Back-End:
  + Para la arquitectura en el back-end se usará el lenguaje NodeJS, puesto que es un lenguaje que se acopla bien a las necesidades del negocio y provee facilidad para interactuar con todos los demás componentes de la arquitectura.
  + Se usará Express como framwork para NodeJS puesto que este es el más popular ofreciendo grandes capacidades de manejo de peticiones, comunicación con los demás componentes de la arquitectura además de una gran eficiencia.
* Front-End:
  + Para front-end se usará VueJS. Este framework que es uno de los más populares, ofrece gran facilidad de integración con los demás componentes de la arquitectura; además, es útil para integrar todos los componentes del front en un solo archivo y los programadores están mas familiarizados con este framework.

**AJUSTE: Tiquetes deben tener id de quien lo vendió para poder hacer cuentas por taquillero**