# ANEXO EXAMEN DE CERTIFICACIÓN

|  |  |
| --- | --- |
| Plan de estudio | Desarrollo de aplicaciones Fullstack JavaScript Trainee V2.0 |
| Anexo | E-commerce Bicicletas (Caso Dos) |

# Caso “E-commerce Bicicletas”

“E-commerce Bicicletas” es una organización que se dedica a la venta de bicicletas en una tienda física en diversas ubicaciones. En este negocio se generan registros con respecto a las ventas mediante documentos, planilla de Excel, entro otros. Se registran datos relacionados a órdenes de compra, direcciones de despacho, tiendas, stock de productos, categorías de estos y clientes.

Para el caso, lo que se necesita del sistema es, en primera instancia, almacenar la información a relacionada con los procesos de venta del negocio en cuestión en una base de datos PostgreSQL, para lo cual se le entregará un modelo relacional para poder comprender el como se implementará la misma y pueda construir un sitio web que pueda proporcionar a trabajadores de esta empresa una interfaz amigable para consultar los datos de órdenes de compra, productos en stock, tiendas existentes, entre otros procesos.

Para este sistema se contará con usted como desarrollador FullStack JavaScript que participará de un equipo multidisciplinario conformado por un UX/UI, desarrollador FrontEnd, Scrum Master, QA.

Como se menciona previamente el sistema busca almacenar y administrar el registro de procesos de venta que se realizan en cada tienda de la organización. A continuación, se listan los requisitos funcionales de alto nivel:

* El sistema debe permitir consultar las tiendas existentes, los empleados asociados a cada una de estas
* Mantener registro y entregar reportes sobre órdenes de compra.
* Almacenar información sobre stock de productos y permitir la consulta de esta para mantener informado al personal correspondiente

# Modelo de Datos

A continuación, se presenta el modelo de datos diseñado por el arquitecto en conjunto con un analista.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Como se puede observar, se define un modelo relacional para poder almacenar los datos relacionados a procesos de venta de la organización, en donde, en la tabla SalesOrderHeader se especifican los datos generales de cada orden de compra, desde aquí se define una tabla SalesOrderDetails que contiene el detalle de cada orden, con respecto a que productos, cantidad de productos, precio unitario, entre otros.

Los productos se encuentran calificados por categorías y subcategorías.

Para cada orden de copra también se asocian territorios de venta, clientes asociados a cada orden, tienda, datos de tarjeta con la que se efectúa el pago.

Para cargar la Base de datos se presentan dos posibilidades:

* mediante un restore desde el cliente de base de datos que se esté utilizando o desde la Shell de PostgreSQL.
  + El backup para este proceso se encuentra en la carpeta /recursosBD/backUpSales
* Mediante la ejecución de un script SQL para crear las tablas en la base de datos y luego cargar los datos desde archivos csv desde la Shell de PostgreSQL:
  + Para este segundo caso se presentan los recursos dentro de la carpeta /recursosBD/scriptComandosSales
  + Dentro encontrarás el scriptSales.sql, que te permitirá crear las tablas en PGAdmin una vez creada la base de datos
  + Luego un archivo comandoImportShell.txt, ahí encontrarás los comandos necesarios para la carga de datos de cada tabla utilizando los archivos CSV adjuntos en la misma carpeta (NO OLVIDAR REEMPLAZAR EL PATH CORRESPONDIENTE AL TUYO PERSONAL)

# Requerimientos a desarrollar

El Scrum Master, que es el encargado de definir las tareas dentro del equipo, le ha solicitado a usted que realice las siguientes tareas:

1. Realizar consultas a la base de datos
2. Crear monitor de departamentos

A continuación, se especifica con mayor detalle cada uno de los requerimientos:

1. Realizar consultas a la base de datos

Uno de los gerentes de la organización ha solicitado algunos reportes de la base de datos mientras se desarrolla el sistema, para poder tomar algunas decisiones con cierto grado de urgencia. **Al terminar el script que contendrá todas las consultas de este hito, agregarlas a una carpeta dentro del proyecto a desarrollar unos pasos más adelante para posteriormente comprimirlas junto a todos los códigos fuente.**

* Se requiere listar los productos que tienen definido a cuál subcategoría pertenecen, omitir del listado aquellos que no cuentan con subcategoría, se necesita mostrar el productid, name, productnumber y productsubcategoryid, la salida debiese mostrar un resultado similar al que se presenta a continuación:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

* Generar un listado con las cantidades de ordenes emitidas por cada año, considerar la fecha de emisión (orderdate), la salida debiese verse similar a la siguiente imagen:

Tabla

Descripción generada automáticamente

* Se necesita un reporte que muestre el listado de órdenes que cuentan con forma de pago con tarjeta vista, tener en consideración tablas creditcard y salesorderheader, se requieren los campos salesorderid, orderdate, duedate, shipdate, cardtype, cardnumber, la salida debe ser similar a la siguiente.

Tabla

Descripción generada automáticamente

1. Construcción de sistema de Gestión de Productos

Se requiere construir una página web dinámica (por el lado del cliente) que permita mostrar el listado de productos y sus subcategorías permitiendo filtrar por criterios que se definen más adelante, tal como se detalla en la siguiente imagen mock-up

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Se pide:

* En el momento que se cargue la página de inicio se deben cargar la totalidad de los productos con los campos que se indican dentro del mock-up, tener en consideración ASOCIACIÓN con productsubcategory para mostrar toda la información necesaria
* Al momento de presionar el botón Filtrar, se debe realizar un proceso de solicitud al backend para filtrar con respecto a los parámetros ingresados por el usuario de la página (ojo que por código se producto se refiere a productnumber)
* Se debe tener en consideración que se debe construir un filtro dinámico, dado que, el usuario puede ingresar uno, dos, tres o cuatro parámetros de filtro
* Validar los procesos necesarios y control de excepción interno para controlar los posibles sucesos que puedan pasar en tiempo de ejecución
* Aplicar seguridad para el sistema mediante JWT a los endpoints correspondientes (no es necesario implementar una tabla para este proceso, basta con validar credenciales de acceso mediante datos hardcoreados en el código JS)

Para realizar el requerimiento, el Scrum Master menciona lo siguiente:

* Utilizar Bootstrap para enriquecer visualmente la página HTML
* Utilizar NodeJS para poder trabajar con JS fuera del Navegador
* Utilizar NPM para la administración de paquetes en su proyecto
* Utilizar ExpressJS para construir y levantar un servicio rest
* Utilizar las librerías necesarias para conectar a base de datos
* Tener en cuenta que el sitio debe ser responsivo al momento de implementar la vista
* Para la construcción del sistema se debe plantear la construcción de una API Rest con la consideración de la implementación de recursos estáticos que permitan consumir los EndPoints de la API y de esta forma construir el frontend como se plantea en el mockup con los requerimientos funcionales que se definen previamente.