**ITERACIÓN 4 – MODELAJE DE UNA APLICACIÓN TRANSACCIONAL**

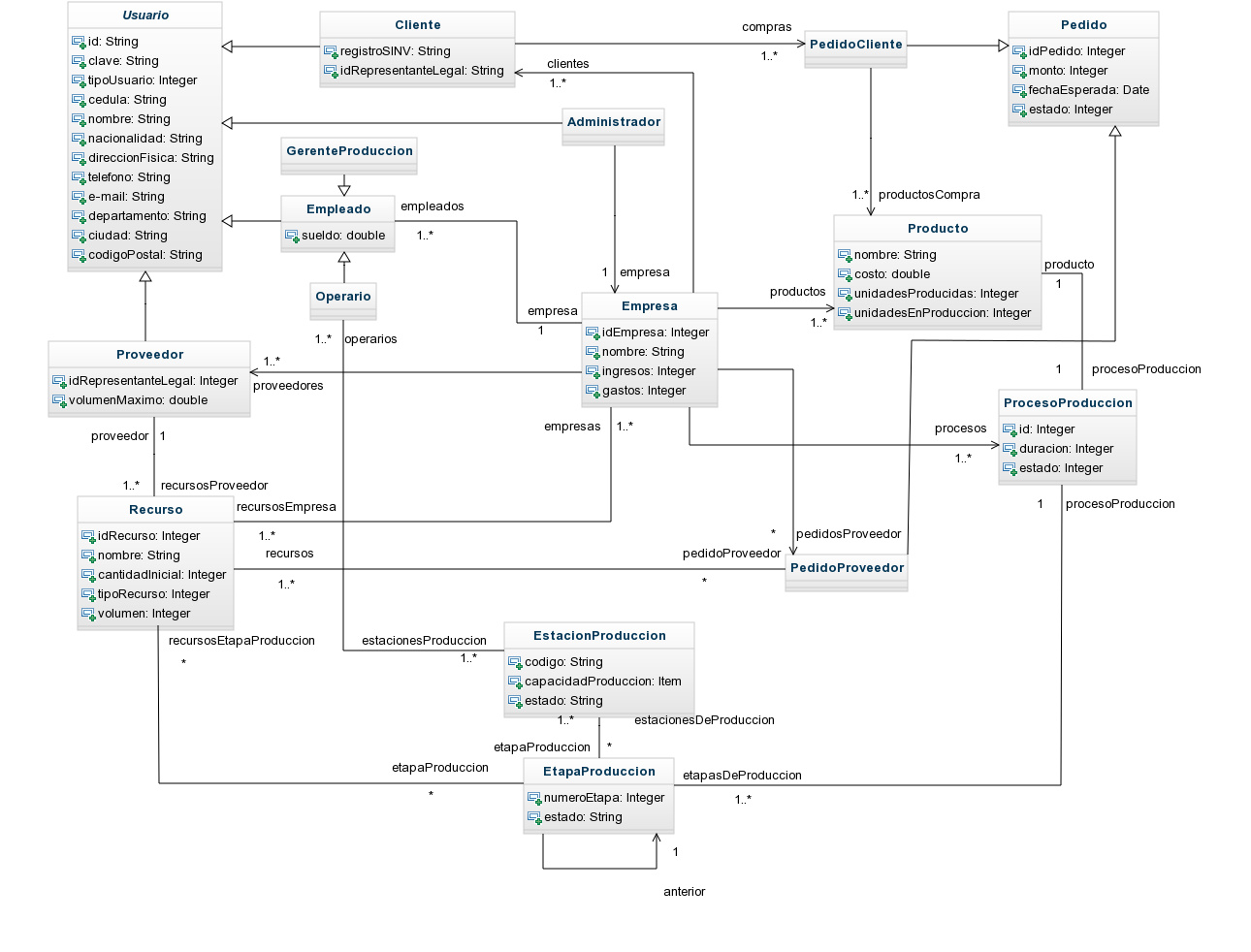
**Grupo 3**

**Carlos Felipe Agudelo Ospina 201328150**

**Sergio Yodeb Velásquez Yepes 201315851**

1. **Análisis**

Con respecto a la entrega pasada no se realizó ningún cambio al modelo conceptual.



Con respecto al modelo conceptual se remplazó la tabla operan con la tabla ejecutaron. Para que se pudiera tener la relación de ejecución de etapas con respecto a una estación de producción, porque en un principio no se había analizado bien esta realción.

|  |  |  |
| --- | --- | --- |
| **Operan** |  |  |
| idOperario | idEtapaProduccion | fechaEjecucion |
| PK1 | PK2, FK(EtapasProduccion.idEtapa) | NN |
| duracion | idPedido |  |
| NN | PK3(Pedido.idPedido) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ejecutaron** | |  |  |  |
| idEtacionProduccion | | idEtapaProduccion | idPedido |
| PK1, FK(EstacionesProduccion.idEstacionProduccion) | | PK2, FK(EtapasProduccion.idEtapa) | PK3,FK(Pedido.idPedido) |
| TiempoEjecucion | | idOperario | fechaEjecucion |  |
| NN | | PK4, FK(Operarios.idOperario) | NN |  |
|  |

Por esta misma razón se agregó la columna IDPEDIDO a la tabla de ejecutan para que hubiera una relación de cada ejecución con un pedido específico.

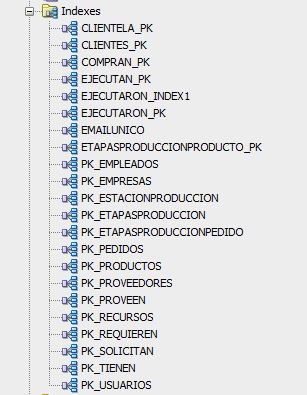
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ejecutan** |  |  |  |  |
| IdEstacionProduccion | IdEtapaProduccion | IdPedido |  |
| PK1, FK(EstacionesProduccion.idEs) | PK2, FK(EtapasProduccion.idEtapa) | PK3, FK(Pedidos.idPedido) |  |

1. **Diseño físico**
   1. **Documentación del diseño físico**

Los índices que manejamos son principalmente los primary key de todas las tablas que son generados por Oracle al nosotros asignarles el valor de primary key. Debido a la diversidad de las consultas crear índices para hacer más eficiente cada una de estas terminaría siendo crear índices para todas las columnas de la base de datos. Por lo cual nosotros miramos que columnas eran las más usadas en todos los requerimientos en general, encontramos que las que más se usaban en nuestro modelo relacional eran la FECHAPEDIDO, la FECHALLEGADA de la tabla Pedidos, el COSTO de la tabla recursos y de la tabla productos, . Puesto que la mayoría de requerimientos tienen como parámetro de búsqueda estas columnas. Y específicamente para esta iteración se usaron índices en la FECHAEJECUCION de la tabla ejecutaron, en TIPORECURSO en la tabla recursos.

Pero además hicimos 2 métodos en la clase DAO que nos permiten crear y eliminar índices cuando sea necesario.

Los índices creados de forma automática por Oracle son:



Los índices creados son las primary key de las tablas y ayudan al rendimiento en la medida de que una búsqueda sobre las tuplas de la tabla se hace buscando por las columnas que la diferencian, que justamente son las primary key.

* 1. **Documentación del análisis**
     1. **Documentación del escenario de pruebas**
     2. **Análisis de eficiencia**
* **RFC8. Consultar ejecución de etapas de producción 1**
  + **Escenarios con diferentes selectividades**

La selectividad de este requerimiento se puede dividir en la selectividad de las fechas y la selectividad del criterio de búsqueda asociado. Por lo tanto los escenarios serían:

Hay una dispersión muy baja en las fechas, todas son de un mismo periodo de tiempo, esto implicaría que la selectividad de las fechas es

* + **Diseño de un plan de ejecución**
  + **Comparar con el de Oracle**
* **RFC9. Consultar ejecución de etapas de producción 2**
* **RFC10. Consultar pedidos 2**
* **RFC11. Consultar material 2**

1. **Construcción de la aplicación y análisis de resultados**
   1. **Documentación del proceso de la carga de datos**
   2. **Análisis del proceso de optimización y el modelo de ejecución de consultas**

La diferencia principal es que el manejador de bases de datos tiene desarrollada una serie de algoritmos que le permiten encontrar de forma muy eficiente los datos, así como una forma de organizar físicamente la información para encontrarla de forma más efectiva, reduciendo los tiempos de la consulta a tiempos muy cortos. Habitualmente tienen los datos organizados en arboles b+ o tablas hashing expandibles, lo que les permite hacer operaciones de selección de una manera muy rápida. Además normalmente cuando uno hace la consulta el manejador no carga todos los datos inmediatamente sino que a medida que uno va pidiendo más datos él los va buscando y trayendo a la memoria principal.

Mientras que si uno quisiera hacer esto en la aplicación le tomaría mucho tiempo, puesto que si quisiera primero organizar los datos para hacer búsquedas sobre ellos tendría que sumarle al tiempo de búsqueda el tiempo de organizar los datos y para cada búsqueda con datos distintos tendría que volver a hacer el mismo proceso. O si intentara hacer la búsqueda sin ordenar el tiempo sería que para cada operación tendría que recorrer todos los datos que tiene.