**Prueba Java Junior y Senior**

La prueba está dividida en dos partes: **Prueba Junior** y **Prueba Senior**. El postulante deberá completar y anexar toda la información solicitada según su caso.

**Consideraciones:**

* **Para Junior**:
* El postulante **Junior** debe completar únicamente la sección **Prueba Junior**, que consta de **4 actividades**.
* Si el postulante considera que tiene la capacidad de desarrollar la **Prueba Senior** total o parcialmente: se evaluará con un puntaje positivo adicional.
* **Para Senior**:
* El postulante **Senior** debe completar ambas partes: **Prueba Junior** y **Prueba Senior**.
* Las pruebas están estructuradas de la siguiente manera:
* **Prueba Junior:** 4 actividades
* **Prueba Senior:** 3 actividades
* **Total actividades para Senior:** 7 actividades

**PRUEBA JUNIOR**

A continuación, se presenta la estructura de las tablas las cuales servirán de base para desarrollar las cuatro actividades a tratar más adelante para esta prueba:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| company | Datatype |  | version | Datatype |
| **id\_company** | number | **version\_id** | number |
| **codigo\_company** | String | **app\_ id** | number |
| **name\_company** | String | **version** | string |
| **description\_company** | String | **version\_description** | String |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| version\_company | Datatype |  | application | Datatype |
| **version\_company\_id** | number |  | **app\_id** | number |
| **company\_id** | number |  | **app\_code** | String |
| **version\_id** | number |  | **app\_name** | String |
| **version\_company\_description** | string |  | **app\_description** | String |

Adicional, se cuenta con las siguientes condiciones asociadas a estas tablas:

* La tabla Company cuenta con un identificador único,que no se deberá repetir (id\_company) y cuenta con un código único para cada empresa.
* La tabla application cuenta con un identificador único,que no se deberá repetir (app\_id) y cuenta con un código único para cada aplicación.
* La tabla version cuenta con un identificador único,que no se deberá repetir (version\_id) ,cabe resaltar que solo se puede tener una version por aplicación.
* La tabla version\_company cuenta con un identificador único,que no se deberá repetir (version\_company\_id) ,cabe resaltar que solo se puede tener una aplicacion por compañía.

**Actividades a desarrollar para el aspirante Junior**

Teniendo en cuenta lo anterior, usted deberá realizar las siguientes actividades:

* Elabore un **gráfico que ilustre la relación entre las tablas presentadas anteriormente**, tomando en cuenta las reglas expuestas. El gráfico debe representar claramente la relación entre las tablas y especificar: Primary Key, Foreign Key y Unique Key de los atributos que apliquen.



🔁 Relaciones entre tablas

📦 **Company**

* id\_company: Clave primaria (PK).
* codigo\_company: Código único para cada empresa (UNIQUE).

Esta tabla representa las empresas registradas.

**🔗 Version\_Company**

Tiene relaciones con Company y Version:

* company\_id: Clave foránea (FK) hacia Company(id\_company).
* version\_id: FK hacia Version(version\_id).

También tiene su propio ID (version\_company\_id, PK).

Representa qué versión de la aplicación está asociada a cada empresa.

Regla: Una empresa puede tener una sola versión (por la restricción UNIQUE en company\_id, como mencionaste antes).

**📥 Version**

* version\_id: Clave primaria.
* app\_id: Clave foránea que apunta a Application(app\_id).

Tiene restricción UNIQUE (app\_id) → una sola versión por aplicación.

Representa una versión específica de una aplicación.

**🧩 Application**

* app\_id: Clave primaria.
* codigo\_app: Código único de la aplicación.

Representa una aplicación registrada en el sistema.

🔁 Resumen de relaciones

Tabla Se conecta con Relación

version\_company company FK (company\_id)

version\_company version FK (version\_id)

version application FK (app\_id)

📌 Restricciones claves en el modelo

Una aplicación tiene una sola versión (UNIQUE app\_id en version).

Una empresa puede tener una sola versión asociada (UNIQUE company\_id en version\_company).

Esto hace que cada empresa tenga como máximo una versión de una aplicación, y cada aplicación una única versión.

* Genere un **script** para realizar la **creación** de la **tabla ’VERSION\_COMPANY’,** tenga presente **sus respectivas llaves(Primary key,Foreign key,Unique key).**
* Genere un **procedimiento** que realice la **inserción de datos en las 4 tablas anteriormente expuestas**, este procedimiento **obtendrá la información de un cursor que lo deberás llamar ‘CTemporal’**, el cual consultará toda la información de la tabla **TMP\_LLENAR\_CAMPOS.**

**Nota**: Usted deberá llenar la tabla TMP\_LLENAR\_CAMPOS ,respecto a la interpretación que haya tenido del punto 1.

Para su facilidad, se proporciona script que crea la tabla TMP\_LLENAR\_CAMPOS:

CREATE TABLE TMP\_LLENAR\_CAMPOS (id\_company NUMBER ,codigo\_company VARCHAR2,name\_company VARCHAR2,description\_company VARCHAR2,version\_id NUMBER,app\_id NUMBER,version VARCHAR2,version\_description VARCHAR2,version\_company\_id NUMBER,company\_id NUMBER,version\_id NUMBER,version\_company\_description VARCHAR2,app\_id NUMBER,app\_code VARCHAR2,app\_name VARCHAR2,app\_description VARCHAR2);

* Realice una **API REST** donde implemente un **CRUD** para: consultar, eliminar, actualizar e insertar datos en la tabla ‘COMPANY’.

Adicional, usted deberá exponer un método de consumo ‘GET’ que reciba como parámetro el código de la compañía y devuelva un JSON con los campos: codigo\_company, name\_company, app\_name y version.



**PRUEBA SENIOR**

El **postulante Senior deberá realizar las 4 actividades anteriores (Prueba Junior)**, y **adicional**, debe de **completar la prueba con los siguientes 3 puntos**:

* Basándose en la información completada en el **punto 3 de la prueba Junior:**

Desarrolle el código que elimine los datos duplicados para garantizar que el listado final contenga sólo registros únicos. Además, adjunte las líneas de código que demuestren cómo se realizó la eliminación y generación de datos (Argumentar la respuesta)

**Para eliminar duplicados en la tabla company, utilizamos una consulta con ROW\_NUMBER() que asigna un número incremental a cada registro con el mismo codigo\_company. Luego, eliminamos todas las filas cuyo número sea mayor que 1, asegurando que solo quede un registro único por código. Este método es eficiente y estándar en SQL Server, permitiendo mantener integridad y limpieza en los datos. La implementación en Java ejecuta esta consulta a través de JdbcTemplate y reporta cuántos registros fueron eliminados**

* Cree una aplicación de calculadora con todas sus validaciones y con todas sus operaciones. Esta calculadora debe de permitir operaciones con decimales, y adicional, debe permitir obtener la raíz cuadrada.
* Desarrolle un código que genera una o varias palabras del siguiente Array de caracteres:
* 

cadena = [w,e,r,f,b,h,j,i,u,y,t,r,e,d,f,g,y,u,i,o,l,k,m,n,b,v,f,r,e,w,s,x,f,g,y,u,i,k,m,n,b,v,f,r,e,w,w,r,t,y,u,i,o,k,m,n,b,v,w,s,x,c,f,g,h,u,i,o,p,l,k,n,b,v,f,d,e,w,a,z,x,c,g,h,u,i,o,p,u,y,t,r,e,w,q,s,d,f,g,k,j,v,c,x]



El código debe combinar los elementos del array para formar palabras válidas. Explique brevemente el enfoque utilizado para seleccionar y unir los caracteres.

Definición del conjunto permitido:

Se creó un Set llamado allowedChars que contiene todos los caracteres permitidos, tal como los listaste. Esto permite hacer búsquedas rápidas (en tiempo constante) para verificar si un carácter está autorizado.

Filtrado de palabras:

Antes de seleccionar una palabra aleatoria, se filtran las palabras disponibles para quedarse sólo con aquellas que contienen únicamente caracteres que están en el conjunto permitido. Esto se logra con:

tsx

Copiar

Editar

const filtered = words.filter(word =>

[...word].every(ch => allowedChars.has(ch.toLowerCase()))

);

Aquí se descompone cada palabra en caracteres y se valida que todos los caracteres sean permitidos.

Selección aleatoria sólo de palabras válidas:

Si hay palabras válidas después del filtro, se selecciona aleatoriamente una entre ellas para mostrarla. Si no hay ninguna válida, se muestra un mensaje de error.