



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

**FACULTAD DE INGENIERÍA CIVIL, SISTEMAS Y DE
ARQUITECTURA**

Escuela Profesional De Ingeniería De Sistemas



TESIS

**“Herramienta de Generación de Script para Beans y Dao
en Proyectos de Lenguaje Java”**

**Para optar el Título Profesional de:
Ingeniero de Sistemas**

Autores:

**BACH. EDVIN EDUARDO HERNANDEZ CABALLERO
BACH. JOSE WILLIAMS SANDOVAL DAMIAN**

Asesor:

DR. ING. REGIS JORGE ALBERTO DIAZ PLAZA

FECHA DE SUSTENTACIÓN

16/09/2022

**Lambayeque - Perú
2022**



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**

**FACULTAD DE INGENIERÍA CIVIL, SISTEMAS Y DE
ARQUITECTURA**

Escuela Profesional De Ingeniería De Sistemas



TESIS

**“Herramienta de Generación de Script para Beans y Dao en Proyectos
de Lenguaje Java”**

Aprobado por los Miembros del Jurado:

**Dr. Ing. Ernesto Karlo Celi Arévalo
PRESIDENTE DEL JURADO**

**Mg. Ing. Robert Edgar Puican Gutiérrez
SECRETARIO**

**Mg. Ing. Oscar Efrain Capuñay Uceda
VOCAL**

**Lambayeque - Perú
2022**

AGRADECIMIENTOS:

En primer lugar, el presente proyecto de tesis agradezco a Dios por ser mi guía y acompañarme en el transcurso de mi vida, brindándome paciencia y sabiduría para culminar con éxito mis metaspropuestas.

A mis padres por ser mi pilar fundamental y haberme apoyado incondicionalmente, pese a las adversidades e inconvenientes que se presentaron.

Y finalmente le agradezco muy profundamente a mi asesor, Dr. Ing. Regis Jorge Alberto Díaz Plaza, por su dedicación y paciencia, sin sus palabras y correcciones precisas no hubiese podido lograr llegar a esta instancia tan anhelada. Gracias por su guía y todos sus consejos, los llevarégrabados para siempre en la memoria en mi futuro profesional.

Edvin Eduardo Hernández Caballero

En primer lugar, a Dios por brindarme salud, sabiduría y fortaleza para seguir adelante en nuestros proyectos y así alcanzar nuestros objetivos.

A mi familia quienes, de alguna u otra forma, me apoyaron siempre en el cumplimiento de misMetas.

Y finalmente, un agradecimiento muy especial a nuestro asesor el Dr. Ing. Regis Jorge Alberto Díaz Plaza por todo el apoyo brindado para poder concluir con nuestro proyecto satisfactoriamente brindándonos sus conocimientos.

José Williams Sandoval Damian

DEDICATORIAS

A Dios,

por ser el inspirador y darnos fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mis Padres Herelin y Lili,

que han sabido darme su ejemplo de trabajo y honradez.

A mi esposa Fiorella e hija,

por su apoyo y paciencia en este proyecto de estudio.

A mis hermanos Miguel y Jhon,

Por darme siempre esos ánimos de seguir adelante y nunca darme por vencido.

Edvin Eduardo Hernández Caballero

A Dios,

Que siempre ha estado en todo mi camino guiándome y bendiciéndome he logrado cumplir uno de mis más grandes logros: ser un profesional.

A mis Padres Duilio y Irma

Quienes, con su esfuerzo, dedicación, consejos, ánimo y sobre todo por su gran amor han hecho de mí una buena persona, reflejo de su maravilloso ejemplo.

A mis hermanos

Por todo su apoyo brindado y motivaciones.

José Williams Sandoval Damian

RESUMEN

La investigación abordó el problema que presentan los desarrolladores de sistemas de información en lenguaje Java cuando deben construir los componentes Beans y DAOs, formulando el problema ¿cómo mejorar el proceso de desarrollo de sistema de información en la implementación de los Beans y DAO? El producto final de la investigación es una herramienta para la generación de los scripts de Beans y DAOs, desarrollado con el lenguaje Java y JSP.

La base teórica que permitió el desarrollo de la investigación fue la metodología RUP (Proceso Unificado Racional) para determinar la secuencia de actividades a seguir en el desarrollo de la herramienta, y la clasificación de herramienta CASE que permitió decidir el tipo de herramienta “lower case” y el aporte de (DE MIGUEL, PIATTINI, & MARCOS, 2000) para definirla como herramienta de análisis y diseño.

ABSTRACT

The research addressed the problem that developers of information systems in Java language present when they must build Beans and DAOs components, formulating the problem, how to improve the information system development process in the implementation of Beans and DAOs? The final product of the research is a tool for generating scripts to Beans and DAOs, developed with the Java language and JSP.

The theoretical basis that allowed the development of the research was the RUP (Rational Unified Process) methodology to determine the sequence of activities to follow in the development of the tool, and the CASE tool classification that allowed deciding the type of lower-case tool and the contribution of (DE MIGUEL, PIATTINI, & MARCOS, 2000) to define it as an analysis and design tool.

INDICE

RESUMEN	5
ABSTRACT.....	6
INDICE.....	7
INDICE DE TABLAS	9
INDICE DE FIGURAS	10
INTRODUCCION.....	13
CAPITULO I PLANTEAMIENTO DE LA INVESTIGACION	14
1.1 Síntesis de la situación problemática	14
1.2 Formulación del problema científico	24
1.3 Objetivos de la investigación	25
1.4 Justificación de la investigación	25
CAPITULO II DISEÑO TEÓRICO	27
2.1 Antecedentes.....	27
2.2 Bases teóricas	28
CAPITULO III DISEÑO METODOLOGICO.....	37
3.1 Procedimiento a seguir en la investigación	37
CAPITULO IV RESULTADOS	38
4.1 Listado de actividades	38
4.2 Modelo de procesos	41
4.3 Diagramas de requerimientos	42
4.4 Análisis y Diseño	45
4.4.1 Diagrama de casos de uso de sistema.....	45
4.4.3 Análisis de los tipos de datos entre el Lenguaje Java y MySQL	46
4.4.4 Análisis del registro y acceso de datos a una tabla desde un sistema información	50
4.4.5 Diseño de Patrón de programación	53
4.5 Desarrollo	57
4.5.1 Arquitectura de la Herramienta de Desarrollo	57

4.5.2 Casos de uso	61
4.5.3 Maqueteado de vistas	77
4.4 Pruebas	86
4.4 VALIDACION DE LOS OBJETIVOS ESPECÍFICOS.....	89
DISCUSION DE LOS RESULTADOS	92
CONCLUSIONES.....	94
RECOMENDACIONES.....	95
BIBLIOGRAFIA.....	96

INDICE DE TABLAS

Tabla 1 Categorización de variable	37
Tabla 2 Secuencia de actividades	38
Tabla 3 Descripción de los tipos de datos en lenguaje Java	46
Tabla 4 Descripción de los tipos de datos en MySQL	47
Tabla 5 Equivalencia del tipo de dato en MySQL y en Java para la implementación de la herramienta de generación de Beans y DAO	49
Tabla 6 Resultado de las pruebas para verificar el cumplimiento de requerimientos funcionales	86
Tabla 7 Relación entre los requerimientos y los objetivos específicos	89
Tabla 8 Validación de objetivos	91

INDICE DE FIGURAS

Figura 1 Lenguajes de programación usado por desarrolladores en los últimos 12 meses .	15
Figura 2 Uso de lenguajes de programación hasta el 2021	15
Figura 3 Lenguajes de programación más importantes	16
Figura 4 Actividades que participa un profesional en el desarrollo de software	16
Figura 5 Tipo de software que desarrolla	17
Figura 6 Plataformas de dispositivos que desarrolla	17
Figura 7 Popularidad de lenguajes de programación	17
Figura 8 Distribución de profesionales según género	18
Figura 9 Experiencia profesional por género	19
Figura 10 Sistema operativo sobre el que desarrollan	20
Figura 11 Proyecto con una estructura Modelo Vista Controlador (MVC)	21
Figura 12 Proyecto con una estructura Spring Web MVC	21
Figura 13 Proyecto con Maven con una estructura MVC	21
Figura 14 Proyecto con Maven con estructura MVC entorno escritorio	22
Figura 15 Proyecto MVC en IDE Eclipse	22
Figura 16 Tabla ejemplo, en su implementación física mediante Workbench y su implementación en clase Beans	23
Figura 5 El gráfico del modelo iterativo muestra cómo se estructura el proceso en dos dimensiones	29
Figura 17 Modelamiento de procesos para generación de BEANs y DAOs	39
Figura 18 Diagrama de requerimientos generales	42
Figura 19 Diagrama de requerimientos - Generar BEANs	43
Figura 20 Diagrama de requerimientos - Generar DAOs	44
Figura 21 Diagrama de casos de uso de sistema de Generación de BEANs y DAOs	45

Figura 22 Patrón para el uso del método insert.....	53
Figura 23 Patrón para el uso del método delete	53
Figura 24 Patrón para el uso del método delete	54
Figura 25 Patrón para el uso del método Select con parámetro	54
Figura 26 Patrón para el uso del método Select	55
Figura 27 Patrón para el uso del método Select cálculos	55
Figura 28 Patrón para el uso del método Select cálculos	56
Figura 29 Patrón para caso especial de consultas con múltiples tablas relacionadas.....	56
Figura 30 Arquitectura de conexión a las bases de datos de las herramientas parte 1	57
Figura 31 Arquitectura de conexión a las bases de datos de las herramientas parte 2	58
Figura 32 Proyecto de Herramienta de generación de script para Beans y DAOS en proyectos de Lenguaje Java	59
Figura 33 Vista de menu.jsp	77
Figura 34 Vista Conectar_BD.jsp	77
Figura 35 Vista beans.jsp?tabla01=producto	77
Figura 36 Vista daos.jsp?tabla01=producto.....	78
Figura 37 Vista daos.jsp?opcion=1&tabla01=producto	78
Figura 38 Vista daos.jsp?opcion=2&tabla01=producto	79
Figura 39 vista parametros.jsp?opcion=3&tabla01=producto.....	79
Figura 40 vista daos.jsp con script de método delete	80
Figura 41 vista parametros_1.jsp?opcion=4&tabla01=producto	80
Figura 42 vista daos.jsp con el script del método update	81
Figura 43 vista parametros_2.jsp?opcion=5&tabla01=producto	81
Figura 44 vista daos.jsp con script del método select con parámetros.....	82
Figura 45 vista parametros_2_all.jsp?opcion=5&tabla01=producto	82

Figura 46 vista con script del método select	83
Figura 47 parametros_2_count.jsp?opcion=5&tabla01=producto	83
Figura 48 vista daos.jsp con script de select count	84
Figura 49 parametros_2_calculation.jsp?opcion=5&tabla01=producto	84
Figura 50 vista daos.jsp con scrip de Select calculation.....	85

INTRODUCCION

La investigación de tipo descriptiva, cualitativa desarrolló la herramienta de generación de Scritp para Beans y DAOs en lenguaje Java,

En el capítulo I Objeto de estudio, se presenta la descripción del contexto del desarrollo de sistemas de información a nivel mundial y la problemática en el desarrollo de sistemas información específicamente en las etapas de construcción de Beans y DAOS en lenguaje Java.

En el capítulo II Marco teórico, se presenta la revisión de los antecedentes sobre desarrollo de sistemas de información, que permitió determinar la Metodología afrontar el desarrollo de la herramienta.

En el capítulo III Metodología, se presenta el tipo de la investigación.

En el capítulo IV Resultados, permite presentar el desarrollo de la herramienta de generación de Scritp y DAOs en lenguaje Java.

La investigación presenta las conclusiones y recomendaciones.

CAPITULO I PLANTEAMIENTO DE LA INVESTIGACION

1.1 Síntesis de la situación problemática

El presente análisis de la realidad respecto al proceso de desarrollo se software JetBrains.

Jetbrains tiene más de 10 millones de usuarios entre libres y de pago, distribuidos en 213 países, “JetBrains es un proveedor de software mundial especializado en la creación de herramientas inteligentes que mejoran la productividad de equipos y desarrolladores de software. Su sede central se encuentra en Praga, en la República Checa, con centros de I+D en San Petersburgo, Moscú, Novosibirsk, Múnich, Ámsterdam y Boston” (Jetbrains, 2022)

En el 2021, presenta “El estado del ecosistema del desarrollador 2021” (Jetbrains, 2021) donde indica que “Los cinco principales lenguajes que los desarrolladores están aprendiendo en 2021 son JavaScript, Python, TypeScript, Java y Go”

Ha realizado un análisis entre los cuales indica el porcentaje utilizado por los desarrolladores:

Lenguajes de programación

Utilizado durante los últimos 12 meses
 Planeo adoptar o migrar

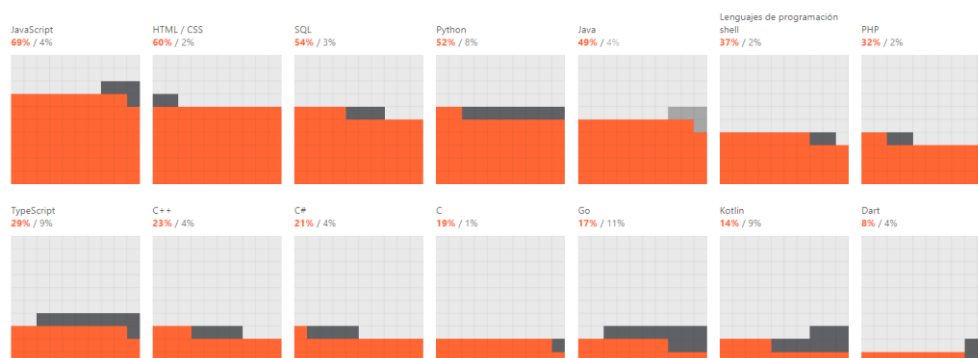


Figura 1 Lenguajes de programación usado por desarrolladores en los últimos 12 meses

EL uso de los lenguajes de programación utilizados hasta el 2021

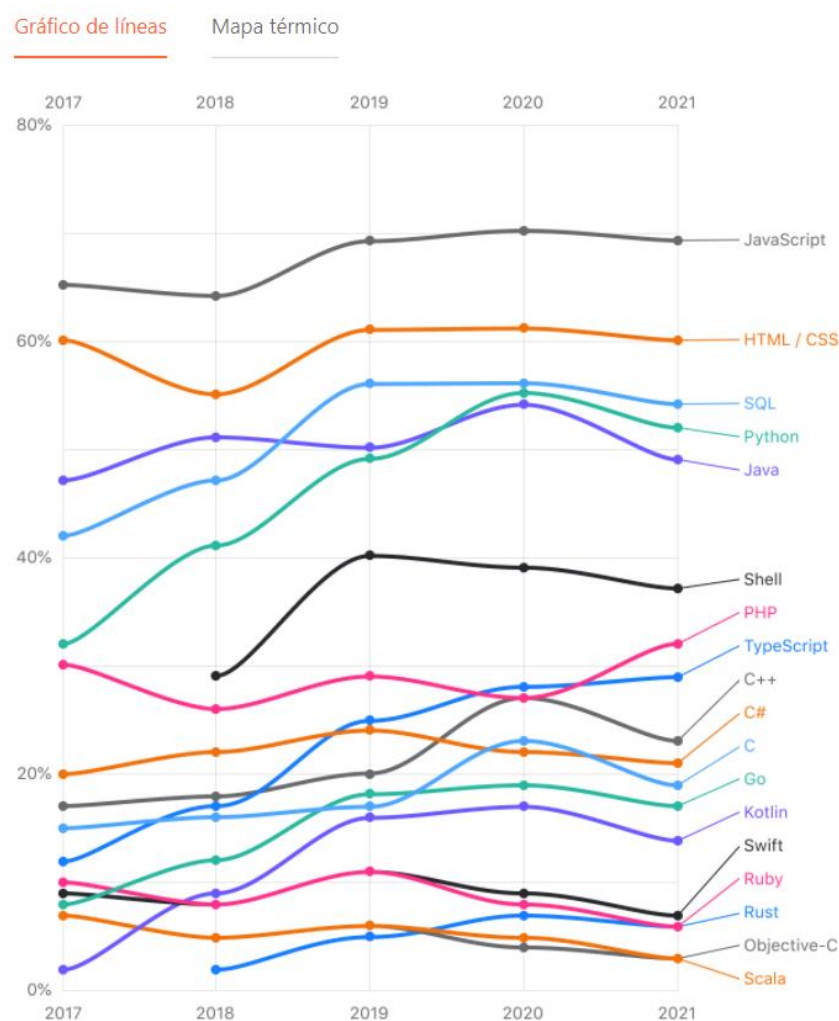


Figura 2 Uso de lenguajes de programación hasta el 2021

El lenguaje de programación más importante es JavaScript seguido de Java

¿Cuáles son sus lenguajes de programación principales?

Elija un máximo de 3 lenguajes

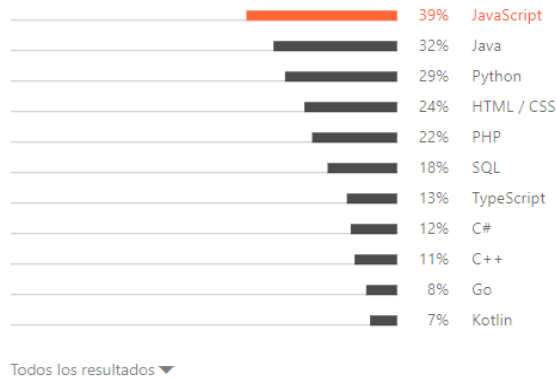


Figura 3 Lenguajes de programación más importantes

El tipo de actividades que participa

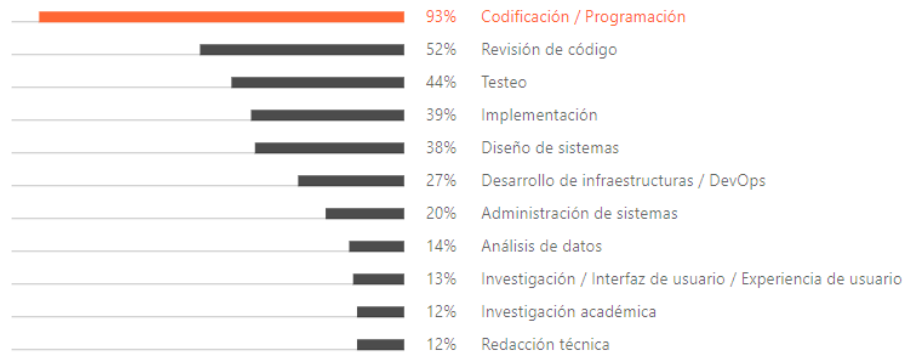


Figura 4 Actividades que participa un profesional en el desarrollo de software

Se observa que 93% de profesionales pasa por la etapa de codificación o programación.

Sobre el tipo de software que desarrolla:

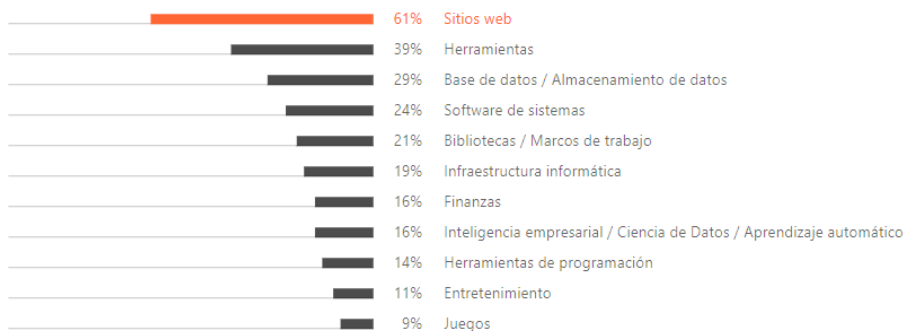


Figura 5 Tipo de software que desarrolla

El 61% de profesionales desarrolla sitios web, 39% herramientas.

Sobre las plataformas de dispositivos en que desarrolla

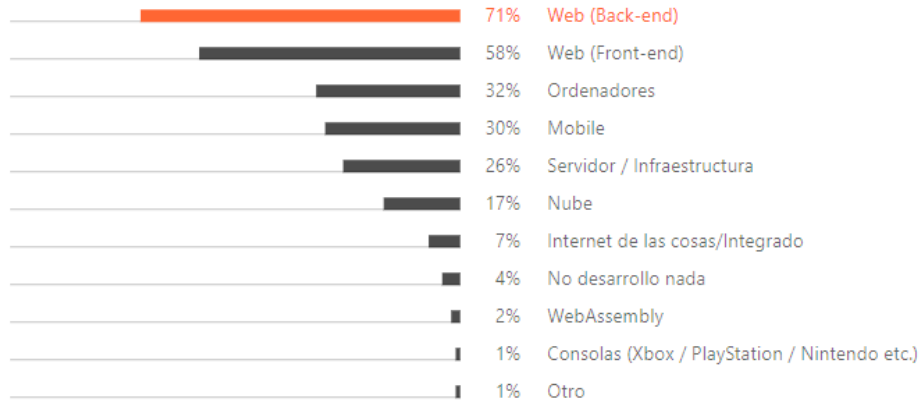


Figura 6 Plataformas de dispositivos que desarrolla

Es importante observar que de los profesionales en desarrollo el 71% lo realiza en back end, y 32% en ordenadores, siendo el lenguaje de programación en estas plataformas más utilizado Java.

Sobre la popularidad de los lenguajes de programación por países:

JavaScript	Java	Python	PHP	SQL	TypeScript	C#	C++	Go	Kotlin	
47 %	25%	25%	33 %	18%	10%	13%	9%	4%	5%	África, Oriente Medio y Asia Central
50%	29 %	21%	19%	24 %	13%	16%	5%	5%	5%	Argentina
36 %	26%	18%	15%	17%	13%	16%	7%	3%	5%	Bielorrusia
50%	28 %	25%	24 %	24 %	14%	13%	5%	3%	8%	Brasil
41 %	31 %	36 %	19%	14%	16%	15%	13%	5%	6%	Canadá
32 %	47 %	30%	9%	15%	11%	8%	13%	15%	6%	China
36 %	24 %	22 %	28 %	19%	18%	15%	10%	5%	6%	Europa del Este, Balcanes y Cáucaso
47 %	21%	20%	43 %	17%	18%	8%	7%	5%	4%	Francia
32 %	33 %	23 %	32 %	15%	19%	15%	11%	7%	8%	Alemania
40%	38 %	44 %	12%	16%	9%	5%	23 %	5%	7%	India
28 %	25%	29 %	21%	18%	18%	13%	8%	10%	9%	Japón
50%	30%	30%	23 %	24 %	14%	16%	5%	4%	7%	América Latina
52 %	35 %	29 %	23 %	25%	12%	18%	7%	2%	7%	México
37 %	24 %	24 %	34 %	18%	17%	18%	8%	4%	7%	Europa del Norte y Benelux
43 %	28 %	23 %	30%	17%	17%	13%	9%	5%	6%	Otro Europa
42 %	27 %	27 %	26%	18%	13%	13%	8%	8%	8%	Otro Sudeste Asiático y Oceanía
28 %	18%	23 %	19%	23 %	13%	15%	13%	9%	8%	Federación Rusa
40%	53 %	31 %	5%	21%	10%	6%	11%	4%	10%	Corea del Sur
43 %	34 %	24 %	31 %	23 %	15%	11%	4%	5%	7%	España
37 %	23 %	25%	21%	20%	6%	34 %	7%	4%	6%	Turquía
38 %	20%	18%	22 %	21%	14%	15%	6%	6%	4%	Ukraine
41 %	22 %	27 %	33 %	18%	16%	16%	8%	6%	4%	Reino Unido
40%	27 %	35 %	27 %	21%	13%	13%	12%	7%	6%	Estados Unidos

Figura 7 Popularidad de lenguajes de programación

Aunque en el informe se aprecia lenguajes de programación que no fueron considerados, destaca para el Perú y sus relaciones con otros países como es China 47%, India 38%, Canadá 31%, Alemania 33%, Brasil 31%, Estados Unidos 27%.

En el análisis según género:



Figura 8 Distribución de profesionales según género

Se aprecia que el mayor porcentaje de profesionales en la industria de desarrollo de software son hombres.

Sobre la experiencia profesional:

Hombre	Mujer	Otro	
81 %	15%	4%	No tengo ninguna experiencia profesional en codificación
88 %	10%	3%	Menos de 1 año
92 %	6%	2%	1-2 años
94 %	4%	2%	3-5 años
96 %	2%	2%	6-10 años
96 %	2%	2%	11-16 años
97 %	2%	2%	16 años o más

Figura 9 Experiencia profesional por género

Es importante que al ingreso a la industria de desarrollo de software el 81% de hombres y el 15 % de mujeres no tienen experiencia.

Después de un año, los hombres incrementan su cuota en el mercado ocupando el 88%, mientras que el 5% de mujeres abandona el mercado bajando al 10%.

A largo plazo, se observa que son los hombres que se mantienen en el mercado, sin embargo, el 13% de mujeres abandonan la industria de desarrollo de software.

Sobre el sistema operativo

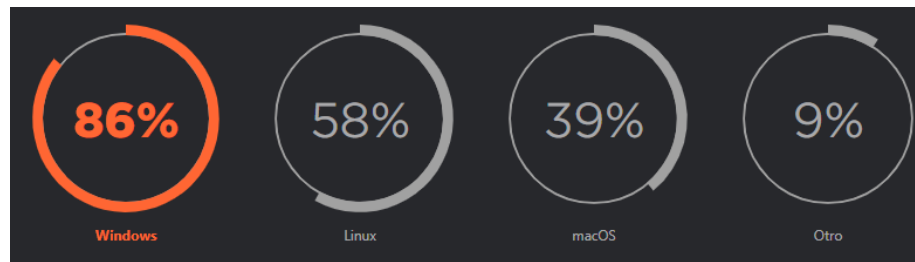


Figura 10 Sistema operativo sobre el que desarrollan

El mercado de los sistemas operativos está cubierto por Windows y Linux, los desarrolladores utilizan estas plataformas de preferencia.

Sobre la tecnología Java (Java, 2022), “El 97% de los escritorios empresariales ejecutan Java, el 89% de los escritorios (o computadoras) en Estados Unidos ejecutan Java, 9 millones de desarrolladores de Java en todo el mundo, la primera opción para los desarrolladores, la primera plataforma de desarrollo, 3 mil millones de teléfonos móviles ejecutan Java, el 100% de los reproductores de Blu-ray incluyen Java, 5 mil millones de Java Cards en uso, 125 millones de dispositivos de televisión ejecutan Java, 5 de los 5 principales fabricantes de equipos originales utilizan Java ME”.

La perspectiva de la industria del software se ha vuelto muy atractiva, se evidencia que el lenguaje Java es el utilizado en el back end, para sitios web, escritorio, y otras aplicaciones, la actividad que casi todos, el 93% de profesionales, han pasado en su formación es la codificación o programación.

En el desarrollo de sistemas de información, los desarrolladores diseñan su proceso de desarrollo de software, a partir de ciclos de vida o metodologías incluyendo la necesidad de diseñar una metodología particular e individual para la organización.

En la etapa de Desarrollo, los desarrolladores proponen la estructura de la aplicación, como se presenta las siguientes figuras:

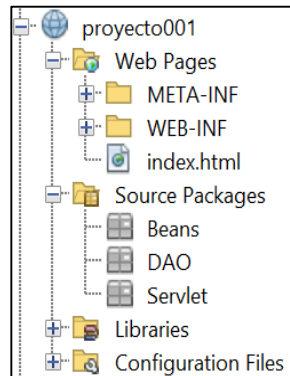


Figura 11 Proyecto con una estructura Modelo Vista Controlador (MVC)

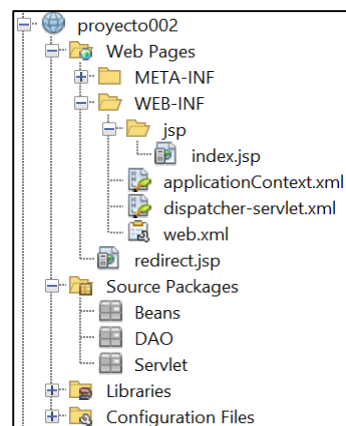


Figura 12 Proyecto con una estructura Spring Web MVC

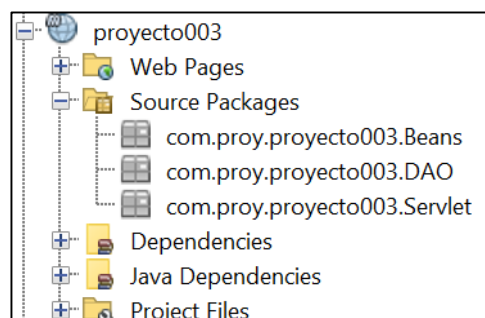


Figura 13 Proyecto con Maven con una estructura MVC

Los tres proyectos figuras 11, 12 y 13, tienen en común el desarrollo de Beans, clase que servirá para almacenar datos y ser reutilizable, y DAO, clase que

permite el acceso a los datos, se utilizan no solamente en proyectos para entorno web, también en proyectos para escritorio como se aprecia en las figuras 14 y 15:

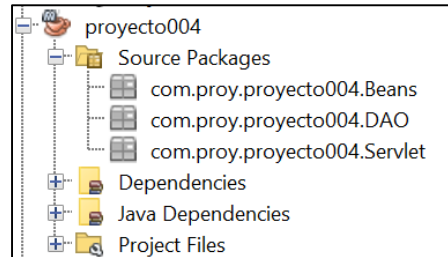


Figura 14 Proyecto con Maven con estructura MVC entorno escritorio

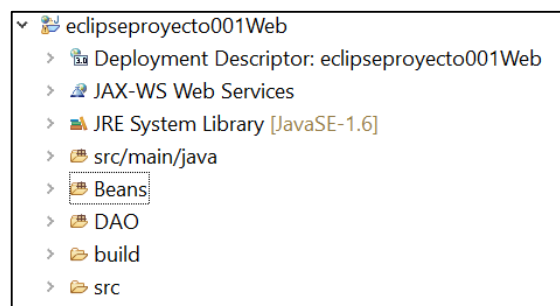


Figura 15 Proyecto MVC en IDE Eclipse

En esta etapa de desarrollo, los desarrolladores tienen como tarea la construcción de estos componentes, convirtiéndose en una tarea repetitiva, aquí se identifica una de las primeras características del problema, estas tareas repetitivas pueden generar Síndrome de Burnout en programadores, cuyas característica son: falta de motivación y debilitamiento de pasión por el código, emociones entumecidas o apagadas, sensación de aislamiento y soledad, falta de concentración, insomnio y problemas para dormir, pérdida de disfrute de las cosas que solías amar.

En la implementación de los Beans, el desarrollador tiene que estar atento a los cambios en la tabla, para modificar el Beans correspondiente, generándose

un retrabajo, cuando una tabla tiene un número de campos pequeños el control de cambios puede ser soportado, pero cuando el número de campos es elevado el control de cambios llega a ser tedioso, como se muestra en la figura 006, un cambio en la tabla implicaría cambio en el Beans de la misma tabla.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI
idejemplo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
campo1	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
campo2	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
campo3	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

package Beans;

public class Beans_Ejemplo {
    int idejemplo;
    String campo1;
    int campo2;
    String campo3;

    public int getId ejemplo() {
        return idejemplo;
    }

    public void setId ejemplo(int idejemp
        this.idejemplo = idejemplo;
    }

    public String getCampo1() {
        return campo1;
    }

    public void setCampo1(String campo1)
        this.campo1 = campo1;
    }

    public int getCampo2() {
        return campo2;
    }

    public void setCampo2(int campo2) {
        this.campo2 = campo2;
    }

    public String getCampo3() {
        return campo3;
    }

    public void setCampo3(String campo3)
        this.campo3 = campo3;
    }
}

```

Figura 16 Tabla ejemplo, en su implementación física mediante Workbench y su implementación en clase Beans

El Beans, que es una construcción abstracta de datos, también puede ser diseñada con los campos a partir de relación entre tablas, para cumplir un requerimiento funcional, por ejemplo, para presentar los datos de una escuela

profesional se relaciona con la tabla facultad pues se requiere el campo NombreDeFacultad, diseñándose el Beans Escuela_Facultad, el desarrollador debe estar muy atento a los cambios en las tablas que se encuentran relacionadas.

En la implementación de los DAO, que es un componente intermedio entre el procedimiento almacenado, y la estructura de datos, su función en la aplicación es gestionar los datos para la inserción, actualización, eliminación y selección, el DAO está sujeto a cambios provenientes de: a) la modificación de las estructuras de datos del sistema de información como: Beans, ArrayList, List o variable, y b) de los cambios en los procedimientos almacenados, además los DAOs deben ser sometidos a la prueba de funcionamiento, esto lo convierte en un componente crítico en el proceso desarrollo del sistema de información. Los desarrolladores tienden a asignar mucho tiempo a su desarrollo y más aún a la gestión de cambios.

La implementación de Beans y DAO son actividades críticas del desarrollo de sistemas de información, en los cuales se observa problemas como actividades repetitivas, retrabajo y mejorar la gestión de cambios, por lo que la investigación permitirá el estudio de estas actividades y el planteamiento de mejorar el desarrollo de estas actividades.

1.2 Formulación del problema científico

¿cómo mejorar el proceso de desarrollo de sistema de información en el desarrollo de los Beans y DAO en lenguaje java?

1.3 Objetivos de la investigación

Objetivo General

Generar los componentes Beans y DAO de un proyecto en lenguaje Java con una herramienta de generación de Script.

Objetivos específicos

Desarrollar módulo de generación de Script para Beans en proyectos de lenguaje JAVA.

Desarrollar módulo de generación de Script para DAO en proyectos de lenguaje JAVA.

1.4 Justificación de la investigación

Se ha identificado tres criterios para justificar la investigación:

Implicaciones prácticas: el producto de la investigación pretende disminuir el problema de actividades repetitivas, retrabajo y gestión de cambios en la implementación de Beans y DAO con lenguaje Java.

Utilidad metodológica: la investigación ayuda a crear un nuevo instrumento para el proceso de desarrollo de sistemas de información.

Conveniencia: es conveniente para desarrolladores de sistemas de información en lenguaje java, tendrán a disposición una herramienta para generación de Beans y DAO, para los investigadores que tendrán una

herramienta que pasará de investigación y desarrollo, que puede pasar a la etapa de producción como un posible producto nuevo en el mercado.

2.1 Antecedentes

Según (López Vallejos, 2011), en su investigación “Análisis del impacto del uso de herramientas CASE en el desarrollo de software dentro de las pequeñas y medianas empresas de Querétaro”, se determinó “las ventajas que la organización observó haciendo uso de alguna herramienta CASE, se obtuvo como resultado que las organizaciones obtenían una mayor productividad, así como mayor calidad del sistema de información entregado, disminución en el tiempo de desarrollo, estandarización, reducción de costo y mayor apoyo en el mantenimiento”. El aporte de esta investigación, desde el 2011 demostraba que una herramienta en el proceso de producción del software es importante respecto a la productividad, calidad, tiempo de desarrollo, estandarización bajo costo y mantenimiento del software.

En la investigación “Análisis de la importancia de las herramientas case en el desarrollo de software el desarrollo de software” (HUARACA BUÑAY, 2018), concluye que “La utilización de Herramientas CASE maximiza la calidad en todo el ciclo de vida al desarrollar aplicaciones informáticas mejorando la productividad y automatizando los procesos de creación, normalización y generación de la base de datos, generación de documentación, generación de diagramas, generación de código, generación de pantallas y generación de paneles de trabajo”,

Los resultados de (CHAVARRÍA-BÁEZ & OCOTITLA ROJAS, 2016) en “Sobre el uso de herramientas CASE para la enseñanza de bases de datos”, analizaron herramientas case para la enseñanza de bases de datos como TabletERD, LDT (Logical Design Tool), Powermodeller, ANACON, Visio (Microsoft), ER/Studio (Embarcadero Technology), ER-Win Data Modeler (Computer Associates), PowerDesigner (Sybase), Easy CASE, resaltando “Herramienta para el Aprendizaje del Álgebra Relacional. Es una herramienta de apoyo que fue desarrollada en la Universidad de Valladolid, la cual permite facilitar el aprendizaje del álgebra relacional y poder realizar consultas sobre cualquier BD”, clasificando como “CASE Académico. Son herramientas software que diversas instituciones desarrollan y utilizan como apoyo en la enseñanza-aprendizaje de sus estudiantes”, “CASE Libre o gratuito. Son herramientas software que desarrollan uno o varios autores y manifiestan que dicho software puede ser copiado, estudiado, modificado, utilizado libremente bajo cualquier fin y redistribuido con o sin cambios o mejoras” y “CASE por costo. Son herramientas software que desarrollan diversas instituciones para satisfacer determinadas necesidades de una sociedad específica, que de acuerdo a las funcionalidades que puedan brindar las herramientas se ve reflejado en el costo de las mismas”.

2.2 Bases teóricas

Metodología RUP

En la investigación se consideró utilizar la metodología RUP (Proceso Unificado Racional) y como lenguaje de modelado, el Lenguaje Unificado de Modelado (UML).

Conceptualizado en (Jacobson, Booch, & Rumbaugh, 2000) “EL Proceso Racional Unificado, es un proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos”.

La metodología define cinco flujos de trabajo: requisitos, análisis, diseño, implementación y prueba, los cuales se ejecutan en cuatro fases: inicio (o concepción), elaboración, construcción y transición.

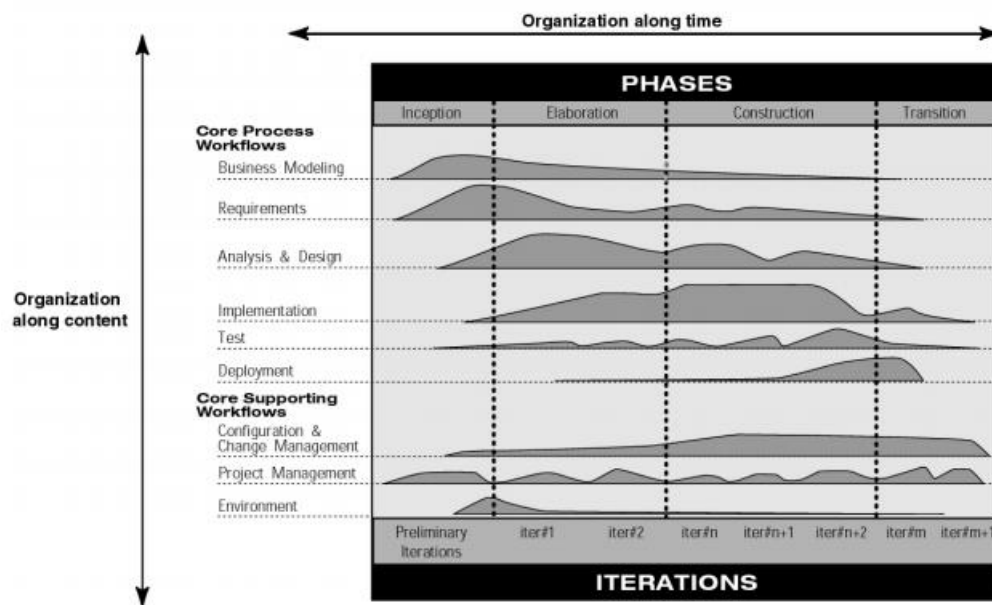


Figura 17 El gráfico del modelo iterativo muestra cómo se estructura el proceso en dos dimensiones.
Fuente: IBM (2020).

Fase de inicio: “Durante la fase inicial, usted establece el caso de negocio del sistema y delimita el alcance del proyecto. Para lograr esto, debe identificar todas las entidades externas con las que el sistema interactuará (actores) y

definir la naturaleza de esta interacción a un alto nivel. Esto implica identificar todos los casos de uso y describir algunos importantes. El caso de negocio incluye criterios de éxito, evaluación de riesgos y estimación de los recursos necesarios y un plan de fases que muestre las fechas de los principales hitos”. (IBM, 2020)

Fase Elaboración: “El propósito de la fase de elaboración es analizar el dominio del problema, establecer una base arquitectónica sólida, desarrollar el plan del proyecto y eliminar los elementos de mayor riesgo del proyecto. Para lograr estos objetivos, debe tener la vista de "una milla de ancho y una pulgada de profundidad" del sistema. Las decisiones arquitectónicas deben tomarse con comprensión de todo el sistema: su alcance, funcionalidad principal y requisitos no funcionales, tales como requisitos de desempeño”. (IBM, 2020)

Fase construcción: “Durante la fase de construcción, todos los componentes restantes y las características de la aplicación se desarrollan e integran en el producto y todas las características se prueban minuciosamente. La fase de construcción es, en cierto sentido, un proceso de fabricación. donde se pone énfasis en la gestión de recursos y el control de las operaciones para optimizar los costos, los cronogramas y calidad. En este sentido, la mentalidad gerencial pasa por una transición desde el desarrollo de la propiedad intelectual durante el inicio y la elaboración, hasta el desarrollo de productos desplegables durante la construcción y la transición”. (IBM, 2020)

Fase transición: “El propósito de la fase de transición es la transición del producto de software a la comunidad de usuarios. Una vez que el producto ha dado al usuario final, generalmente surgen problemas que requieren que desarrolle nuevas versiones, corrija algunos problemas o se termine las funciones que fueron pospuestas”. Traducido de (IBM, 2020)

Modelamiento de negocio: “Uno de los principales problemas con la mayoría de los esfuerzos de ingeniería empresarial es que la ingeniería de software y la comunidad de ingenieros empresariales no se comunican adecuadamente entre sí. Esto conduce a la salida del negocio. La ingeniería no se utiliza correctamente como entrada para el esfuerzo de desarrollo de software y viceversa. RUP aborda esto proporcionando un lenguaje y un proceso común para ambas comunidades, así cómo crear y mantener la trazabilidad directa entre los modelos de software y de negocio. En Modelado de negocio documentamos los procesos de negocios mediante los llamados casos de uso de negocio. Esto asegura una común comprensión entre todas las partes interesadas de qué proceso de negocio necesita ser apoyado en la organización. Los casos de uso de negocio se analizan para comprender cómo la empresa debe respaldar los procesos del negocio. Esto es documentado en un modelo de objeto de negocio. Muchos proyectos pueden optar por no realizar modelos de negocio”. (IBM, 2020)

Requerimiento: “El objetivo del flujo de trabajo de Requerimientos es describir lo que debe hacer el sistema y permite a los desarrolladores y que el cliente esté de acuerdo con esa descripción. Para lograr esto, obtenemos,

organizamos y documentamos la funcionalidad requerida y limitaciones; rastrear y documentar las compensaciones y decisiones”. (IBM, 2020).

Análisis y diseño: “El objetivo del flujo de trabajo de análisis y diseño es mostrar cómo se realizará el sistema en la fase implementación” (IBM, 2020).

Implementación: “El propósito de la implementación es definir la organización del código, en términos de subsistemas de implementación organizados en capas. Implementar clases y objetos en términos de componentes, probar los componentes desarrollados como unidades” (IBM, 2020).

Pruebas: “Los propósitos de las pruebas son: verificar la correcta integración de todos los componentes del software, que todos los requerimientos estén correctamente implementados, identificar y garantizar que los defectos se aborden antes de la implementación del software” (IBM, 2020).

Despliegue: “El propósito del flujo de trabajo de implementación es producir lanzamientos de productos con éxito y entregar el software a los usuarios finales” (IBM, 2020).

CASE: Computer Aided Software Engineering. (Ingeniería de Software Asistida por Computadora.)

HERRAMIENTA CASE: Una herramienta del software que automatiza (por lo menos en parte) una parte del ciclo de desarrollo de software. (EcuRed:Enciclopedia_cubana, 2022)

“Las herramientas de ingeniería de software asistida por computadora (CASE) se definen como herramientas de software que brindan soporte automatizado para una parte del proceso de desarrollo de sistemas”, traducido de (Hoffer, George, & Valacich, 2008)

Clasificación

Actualmente, existen muchas clasificaciones:

Considerando la categorización de (Hoffer, George, & Valacich, 2008):

Upper CASE

Middle CASE

Lower CASE

Debido a sus similitudes, a veces Upper y Middle CASE se denominan simplemente Upper CASE. En general, Upper CASE es una herramienta para una vista de alto nivel del desarrollo de software, mientras que Lower CASE se usa principalmente como una herramienta en la fase de programación y prueba.

Upper CASE

Diagramación de flujo de datos

Diagramación modelo entidad relación ER

Diagramación de flujo de trabajo

Diagramación Orientada a Objetos

Diccionario de datos

Reglas de negocio

Herramientas de análisis

Lower CASE

Generación de código

Generación de formularios

Generación de reportes

Generación de documentos

Utilitarios de importación y exportación

Codificación de programas

Pruebas

Según (DE MIGUEL, PIATTINI, & MARCOS, 2000) presenta:

1) Herramientas de análisis y diseño

Permiten crear y modificar diagramas Entidad-Relación, de flujo de datos, de clases, etc. Son importantes también las herramientas de prototipado. Éstas incluyen diseñadores de formularios, de menús, de informes, y lenguajes de especificación ejecutables. Un aspecto a destacar es la capacidad de análisis y verificación de especificaciones soportado por la herramienta (sintáctica y semánticamente), como, por ejemplo, la capacidad de normalizar hasta la tercera forma normal.

2) Generación de código y documentación

Generan código a partir de las especificaciones de diseño. Además, soportan la generación automatizada de documentación a partir de la información almacenada.

3) Herramientas de prueba

4) Herramientas de gestión de la configuración

5) Herramientas de ingeniería inversa

- Ingeniería inversa de datos, extraen información de código fuente y construye diagramas orientados a objetos o Entidad-Relación.
- Ingeniería inversa de procesos, permiten aislar la lógica de las entidades y las reglas del negocio a partir del código.
- Reestructuración de código fuente, modifican el formato o implantan un formato estándar.
- Re documentación, permiten generar diagramas para mejorar la comprensión del código.
- Análisis de código, generan, por ejemplo, la indentación automática

BPMN

“El objetivo principal de BPMN es proporcionar una notación que sea fácilmente comprensible para todos los usuarios comerciales, desde los analistas comerciales que crean los borradores iniciales de los procesos, hasta los desarrolladores técnicos responsables de implementar la tecnología que realizará esos procesos, y finalmente, a los empresarios que administrarán y monitorearán

Procesos. Por lo tanto, BPMN crea un puente estandarizado para la brecha entre el diseño del proceso empresarial y la implementación del proceso.” (Object Management Group, 2011)

“Esta especificación representa la fusión de las mejores prácticas dentro de la comunidad de modelos de negocios para definir el notación y semántica de diagramas de colaboración, diagramas de proceso y diagramas de coreografía. La intención de BPMN es estandarizar un modelo de proceso de negocios y una notación frente a muchas notaciones y puntos de vista de modelado diferentes. Al hacerlo, BPMN proporcionará un medio simple de comunicar la información del proceso a otros usuarios comerciales, implementadores de procesos, clientes y proveedores” (Object Management Group, 2011)

CAPITULO III DISEÑO METODOLOGICO

3.1 Procedimiento a seguir en la investigación

La investigación se desarrolló con la tipología de No experimental, descriptiva y transversal

3.2 Población y muestra

El objeto observado es el proceso desarrollo de Script y DAO en lenguaje Java.

3.3 Categorización de las variables

Considerando la realidad problemática, antecedentes y base teórica, la investigación desarrolló una herramienta en lenguaje Java o JSP que permite generar código para ser utilizado por los profesionales de la industria del software. Será una herramienta de generación de código tipo Lowe CASE.

Tabla 1 Categorización de variable

Variable de estudio	Objetivo específico	Cumplimiento con requerimientos
HERRAMIENTA DE GENERACIÓN DE SCRIPT PARA BEANS Y DAO EN PROYECTOS DE LENGUAJE JAVA	5.1. Desarrollar módulo de generación de Script para Beans en proyectos de lenguaje JAVA	Dicotómica
	5.2. Desarrollar módulo de generación de Script para Beans en proyectos de lenguaje JAVA	Dicotómica

Fuente: Elaboración propia

3.3 Técnicas, instrumentos, equipos, materiales

El logro de los resultados fue utilizando el software NetBeans.

CAPITULO IV RESULTADOS

4.1 Listado de actividades

Tabla 2 Secuencia de actividades

Fase	Iteración	Disciplina	Entregable	Estado	Secuencia
Inicio	Iteración 01	Modelamiento de Proceso	BPM del proceso versión 01	por revisar	1
		Requerimientos			
		Análisis y diseño	Diagrama de caso de uso de sistema versión 01	por revisar	2
		Desarrollo			
		Pruebas	Creación de base de datos para pruebas	por revisar	3
		Hito 01	Revisión		
	Iteración 02	Modelamiento de Proceso	BPM del proceso versión 02	Aprobado	1
		Requerimientos	Diagramas de requerimientos versión 01	por revisar	4
		Análisis y diseño	Diagrama de caso de uso de sistema versión 02	Aprobado	2
		Desarrollo			
		Pruebas	Creación de base de datos para pruebas	Aprobado	3
Elaboración	Iteración 01		Listado de casos de prueba	por revisar	5
		Hito 02	Revisión		
		Modelamiento de Proceso	BPM del proceso versión 02	Aprobado	1
		Requerimientos	Diagramas de requerimientos versión 02	Aprobado	4
			Diagrama de caso de uso de sistema versión 02	Aprobado	2
		Análisis y diseño	Análisis de los tipos de datos entre el Lenguaje Java y MySQL versión 01	por revisar	6
		Desarrollo			
		Pruebas	Creación de base de datos para pruebas	Aprobado	3
			Listado de casos de prueba	por revisar	5
		Hito 03	Revisión		

Construcción	Iteración 02	Modelamiento de Proceso	BPM del proceso versión 02	Aprobado	1
			Diagramas de requerimientos versión 02	Aprobado	4
			Diagrama de caso de uso de sistema versión 02	Aprobado	2
		Análisis y diseño			
			Análisis de los tipos de datos entre el Lenguaje Java y MySQL versión 02	Aprobado	6
		Desarrollo	Arquitectura del sistema de información	por revisar	7
		Pruebas	Creación de base de datos para pruebas	Aprobado	3
			Listado de casos de prueba	Aprobado	5
		Hito 04	Revisión		
	Iteración 01	Modelamiento de Proceso	BPM del proceso versión 02	Aprobado	1
			Diagramas de requerimientos versión 02	Aprobado	4
			Diagrama de caso de uso de sistema versión 02	Aprobado	2
		Análisis y diseño			
			Análisis de los tipos de datos entre el Lenguaje Java y MySQL	Aprobado	6
		Desarrollo	Patrón para el uso de métodos	por revisar	8
			Arquitectura del sistema de información	Aprobado	7
			Conexión a base de datos	por revisar	9
		Pruebas	Creación de base de datos para pruebas	Aprobado	3
			Listado de casos de prueba	Aprobado	5
		Hito 05	Revisión		
	Iteración 02	Modelamiento de Proceso	BPM del proceso versión 02	Aprobado	1
			Diagramas de requerimientos versión 02	Aprobado	4
			Diagrama de caso de uso de sistema versión 02	Aprobado	2
		Análisis y diseño			
			Análisis de los tipos de datos entre el Lenguaje Java y MySQL	Aprobado	6
		Desarrollo	Patrón para el uso de métodos	por revisar	8
			Arquitectura del sistema de información	Aprobado	7
			Conexión a base de datos	Aprobado	10
			Generación de Beans	por revisar	11

Iteración 03		Generación de DAOs	por revisar	12
	Pruebas	Creación de base de datos para pruebas	Aprobado	3
		Listado de casos de prueba	Aprobado	5
	Hito 05	Revisión		
	Modelamiento de Proceso	BPM del proceso versión 02	Aprobado	1
	Requerimientos	Diagramas de requerimientos versión 02	Aprobado	4
		Diagrama de caso de uso de sistema versión 02	Aprobado	2
	Análisis y diseño	Análisis de los tipos de datos entre el Lenguaje Java y MySQL	Aprobado	6
	Desarrollo	Patrón para el uso de métodos	Aprobado	8
		Arquitectura del sistema de información	Aprobado	7
		Conexión a base de datos	Aprobado	10
		Generación de Beans	Aprobado	11
		Generación de DAOs	Aprobado	12
	Pruebas	Creación de base de datos para pruebas	Aprobado	3
		Listado de casos de prueba	Aprobado	5
		Pruebas	Aprobado	13
	Hito 05	Revisión		

La tabla presenta el resumen de actividades y la secuencia que se desarrollaron y fueron concouyendo.

4.2 Modelo de procesos

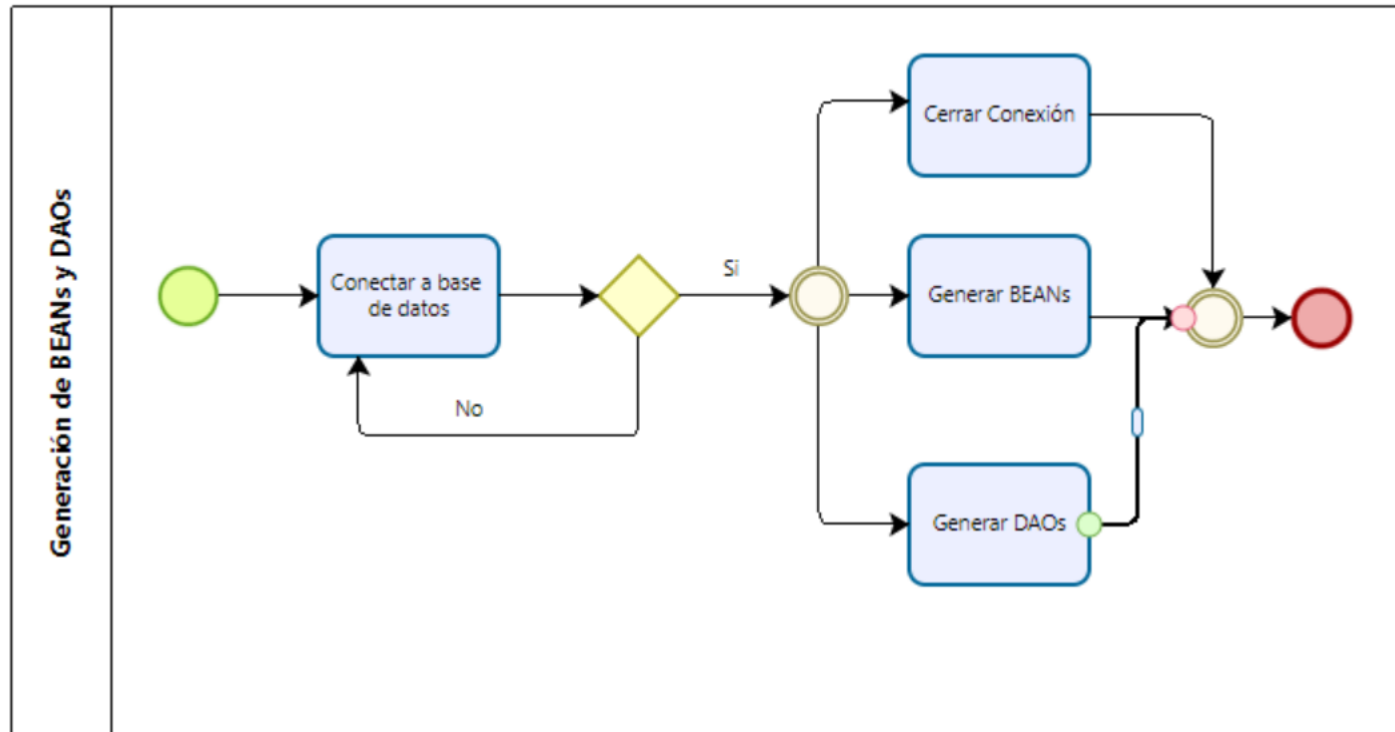


Figura 18 Modelamiento de procesos para generación de BEANs y DAOs

4.3 Diagramas de requerimientos

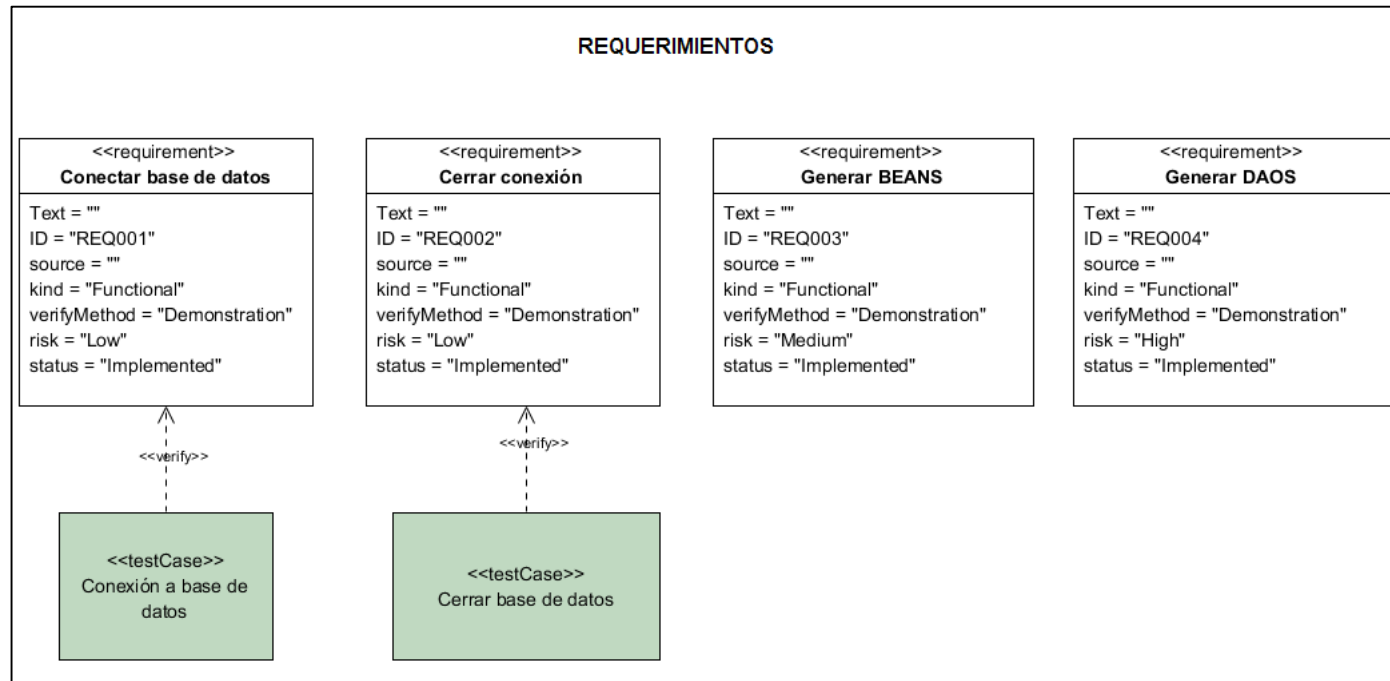


Figura 19 Diagrama de requerimientos generales

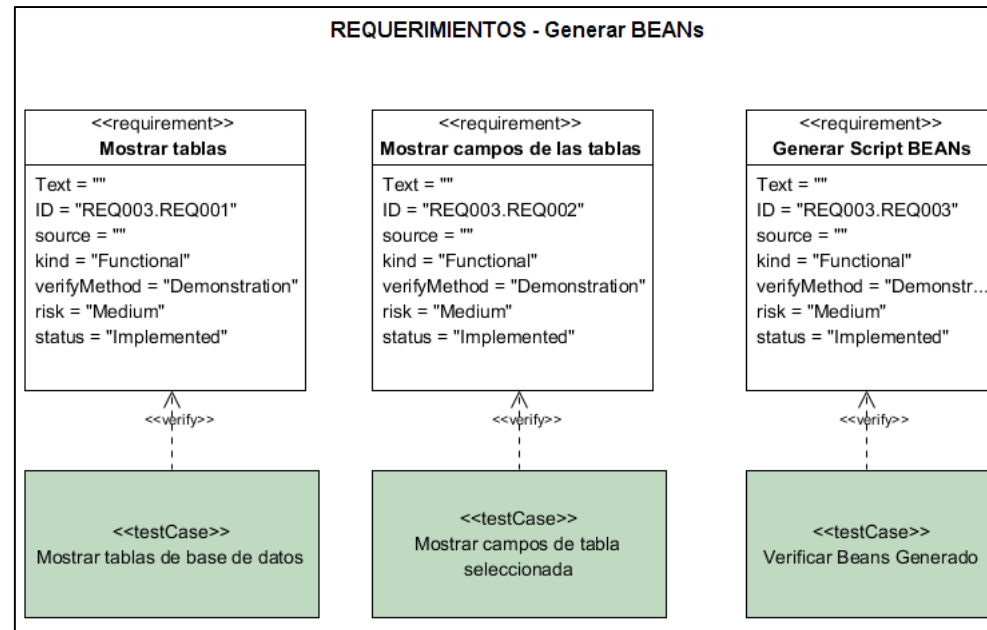


Figura 20 Diagrama de requerimientos - Generar BEANs

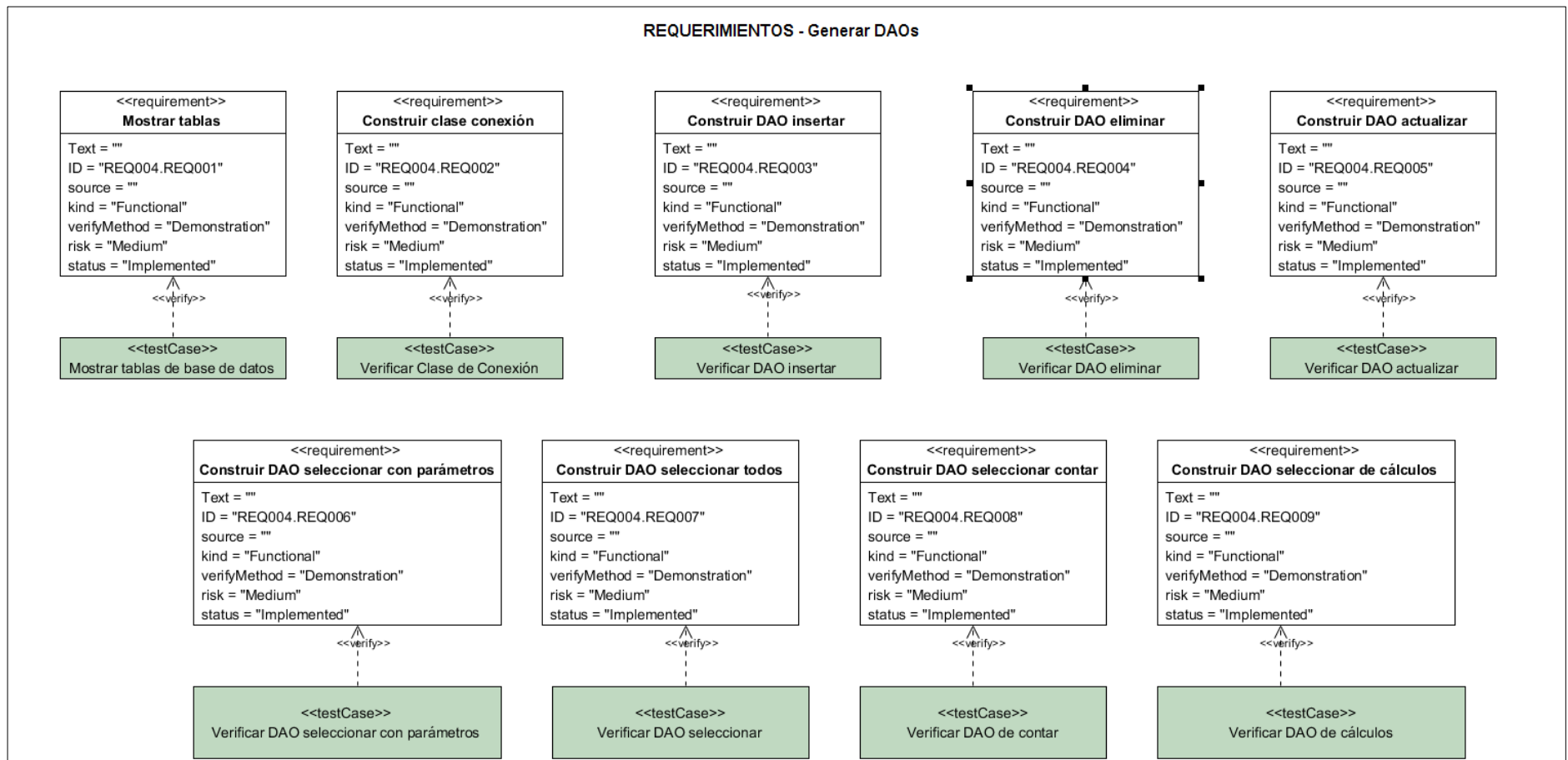


Figura 21 Diagrama de requerimientos - Generar DAOs

4.4 Análisis y Diseño

4.4.1 Diagrama de casos de uso de sistema

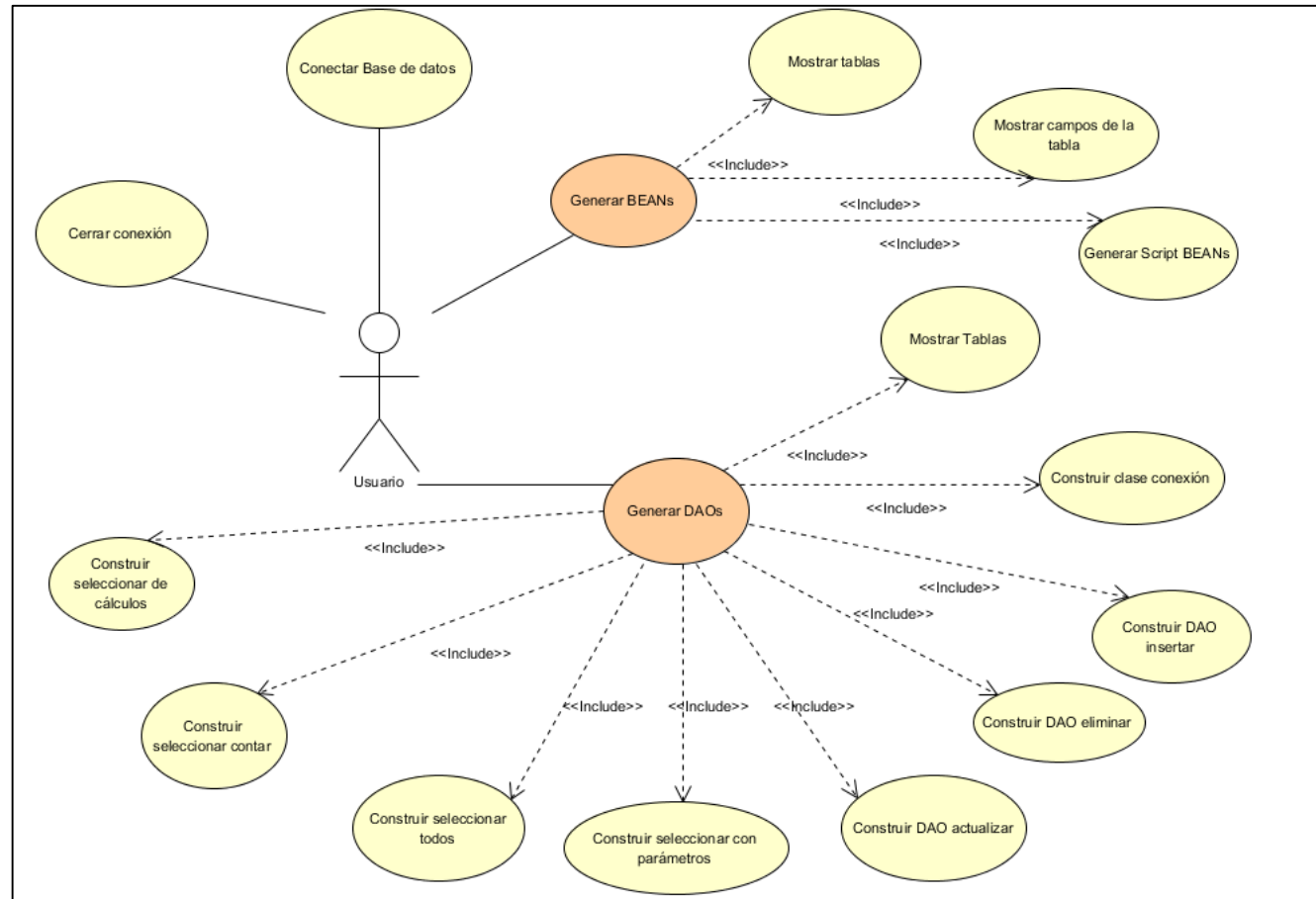


Figura 22 Diagrama de casos de uso de sistema de Generación de BEANs y DAOs

4.4.3 Análisis de los tipos de datos entre el Lenguaje Java y MySQL

Tabla 3 Descripción de los tipos de datos en lenguaje Java

NOMBRE		TIPO	BITS	BYTES	VALOR MINIMO	VALOR MAXIMO	VALOR POR DEFECTO
OBJETOS	PRIMITIVOS						
Byte	byte	byte	4	1 byte	-128	127	0
Short	short	Entero	8	2 bytes	-32768	32767	0
Integer	int	Entero	32	4 bytes	2x10(9)		0
Long	long	Entero	32	8 bytes	-9223372036854775808	-9223372036854775807	0
Float	float	Decimal Simple	32	4 bytes	+/-3.402823473E+38	+/-1.40239846E-45	0.0
Double	double	Decimal doble	64	8 bytes	+/-1.79769313486231570E+308	+/-4.94065645841246544E-324	0.0
Character	char	Carácter simple	16	2 bytes			
Boolean	boolean	1 o 0	1	1 byte			FALSO
String							

Tabla 4 Descripción de los tipos de datos en MySQL

Nº	NOMBRE	SIGNO	SINONIMO	TIPO	VALOR MINIMO	VALOR MAXIMO	FORMATO
1	BIT		TINYINT	Bit	1	64	
2	BOOL, BOOLEAN		TINYINT	Boolean			
3	TINYINT	SIN SIGNO CON SIGNO		Integer	-128 0	127 255	
4	SMALLINT	SIN SIGNO CON SIGNO		Integer	-32768 0	32767 65535	
5	MEDIUMINT	SIN SIGNO CON SIGNO		Integer	-8388608 0	8388607 16777215	
6	INT	SIN SIGNO CON SIGNO		Integer	-2147483648 0	2147483647 4294967295	
7	INTEGER		INT	Integer			
8	BIGINT	SIN SIGNO CON SIGNO		Integer	-9223372036854775808 0	9223372036854775807 18446744073709551615	
9	FLOAT	SIN SIGNO CON SIGNO		Float	3.402823466E+38 1.175494351E-38 Basado en la IEEE, actualmente depende de su hardware y sistema operativo	-1.175494351E-38,0 3.402823466E+38	
10	DOUBLE	SIN SIGNO CON SIGNO		Double	-1.7976931348623157E+308 Basado en la IEEE, actualmente depende de su hardware y sistema operativo	-2.2250738585072014E-308, 0	
11	DECIMAL				65 DÍGITOS, PRECISION		A
12	DATE						0000-00-00
13	TIME						00:00:00
14	DATETIME						0000-00-00 00:00:00
15	TIMESTAMP						0000-00-00 00:00:00
16	YEAR						0000
17	CHAR						Un carácter
18	VARCHAR						0 a 65535 caracteres
19	BINARY		Char				
20	VARBINARY		Varchar				
21	TEXT						65,535 ($2^{16} - 1$) caracteres

22	LONGTEXT	4GB ($2^{32} - 1$) caracteres
23	MEDIUMTEXT	16,777,215 ($2^{24} - 1$) caracteres
24	TINYTEXT	255 ($2^8 - 1$) Caracteres
25	BLOB	65,535 ($2^{16} - 1$) bytes
26	TINYBLOB	255 ($2^8 - 1$) bytes
27	LOB	4,294,967,295 o 4GB ($2^{32} - 1$) bytes
28	MEDIUMBLOB	16,777,215 ($2^{24} - 1$) bytes

Esta tabla presenta descriptivamente las características de los tipos de datos con la finalidad de relacionarlo con un tipo de dato del lenguaje de programación Java

En el ámbito de la investigación para la selección de los tipos de datos no se ha considerado los datos tipo Enum, Set, Spatial y JSON, el criterio de selección fue por el tipo de dato que es sería usado en desarrollo de sistemas de información.

Tabla 5 Equivalencia del tipo de dato en MySQL y en Java para la implementación de la herramienta de generación de Beans y DAO

Nº	MySQL	Java
1	BIT	int
2	BOOL, BOOLEAN	int
3	TINYINT	int
4	SMALLINT	int
5	MEDIUMINT	int
6	INT	int
7	INTEGER – INT()	int
8	BIGINT	int
9	FLOAT	float
10	DOUBLE	double
11	DECIMAL	double
12	DATE	String
13	TIME	String
14	DATETIME	String
15	TIMESTAMP	String
16	YEAR	String
17	CHAR	Char
18	VARCHAR	String
19	BINARY	String
20	VARBINARY	String
21	TEXT	String
22	LONGTEXT	String
23	MEDIUMTEXT	String
24	TINYTEXT	String
25	BLOB	byte
26	TINYBLOB	byte
27	LOB	byte
28	MEDIUMBLOB	byte

De las tablas a y b, se diseña la presente tabla con la equivalente de los tipos de datos entre MySQL y el lenguaje de programación Java, esto sirvió para la creación de los Beans.

4.4.4 Análisis del registro y acceso de datos a una tabla desde un sistema de información

Esta actividad permitió identificar las formas en que puede un desarrollador diseñar la arquitectura de un sistema de información.

Criterio: Acceso a tablas desde el sistema de información

- a) Directamente desde la página web, esta forma es la más fácil de utilizar, no es muy ordenada, y actualmente es poco recomendada.
- b) Desde los DAOs a través de sus métodos con consulta SQL, la consulta SQL se realiza en el método, es decir, no se utilizan procedimientos almacenados, registrándose o accediendo al contenido de las tablas desde el método del DAO.
- c) Desde los DAOs a través de sus métodos con llamadas a los procedimientos almacenados, las consultas SQL se realizan los procedimientos almacenados, el registro o acceso a las tablas se realiza a través de los procedimientos almacenados.

El criterio C, fue el que se consideró para el diseño de la herramienta de generación de Script para BEANs y DAOs, este criterio fue tomado en cuenta en los siguientes análisis de los tipos de consultas SQL.

Criterio: Tipo de consulta insert SQL

Durante la ejecución de un sistema de información se realiza inserción de registros, la inserción puede ser total o parcial en los campos de las tablas,

para evitaría realizar diferentes tipos de procedimientos almacenados de tipo insert y como consecuencia diferente tipos de métodos insert en el DAO, el criterio que se consideró para generar un método insert con el total de campos en el DAO.

Criterio: Tipo de consulta delete SQL

Los procedimientos tipos delete para una tabla, puede ser de diferentes tipos, dependerá de los parámetros en las cláusulas where, order by y group by, haciendo que los métodos delete en el DAO varíen según la cantidad y tipos de parámetros, este criterio se consideró para generar diferentes tipos de métodos delete, en el DAO.

Criterio: Tipo de consulta update SQL

Durante la ejecución de un sistema de información se realiza actualizaciones de registros, la actualización puede ser total o parcial en los campos de las tablas, dependerá de los parámetros en las cláusulas where, para evitaría realizar diferentes tipos de procedimientos almacenados de tipo update y como consecuencia diferente tipos de métodos update en el DAO, este criterio se consideró para generar un método update en el DAO.

Criterio: Tipo de consulta select SQL

Este tipo de consulta select se puede analizar desde cuatro tipos:

- a) Consulta Select con parámetros, son procedimientos almacenados que permiten seleccionar registros según los parámetros que reciban, dependerá de los parámetros en las cláusulas where, order by y group by,

este criterio se consideró para generar este tipo de consulta de diferentes métodos select con parámetros en el DAO.

- b) Consulta Select sin parámetros, este procedimiento almacenado permite seleccionar todos los registros, el criterio se consideró para generar este tipo de consulta de método select sin parámetros en el DAO.
- c) Consulta Select que cuentan registros, este procedimiento almacenado devuelve un único valor numérico, puede ser de diferentes tipos dependerá de los parámetros en las cláusulas where, este criterio se consideró para generar diferentes métodos select que cuenta registros con parámetros en el DAO.
- d) Consulta Select para cálculos, este procedimiento almacenado devuelve un único valor numérico, puede ser de diferentes tipos, dependerá de los parámetros en las cláusulas where, este criterio se consideró para generar diferentes métodos select para en el DAO.
- e) Consulta Select que relacionan diferentes tablas, para este tipo de consultas se indicará que se generen como views en el motor de base de datos, y dependiendo del tipo se convertirá en un tipo de requerimiento tipo consulta select tipo a, b, c o d. Este criterio se consideró para sea incorporado en el criterio Acceso a tablas desde el sistema de información, accediéndose también a las vistas solamente con fines de consulta.

4.4.5 Diseño de Patrón de programación

a) Método insert

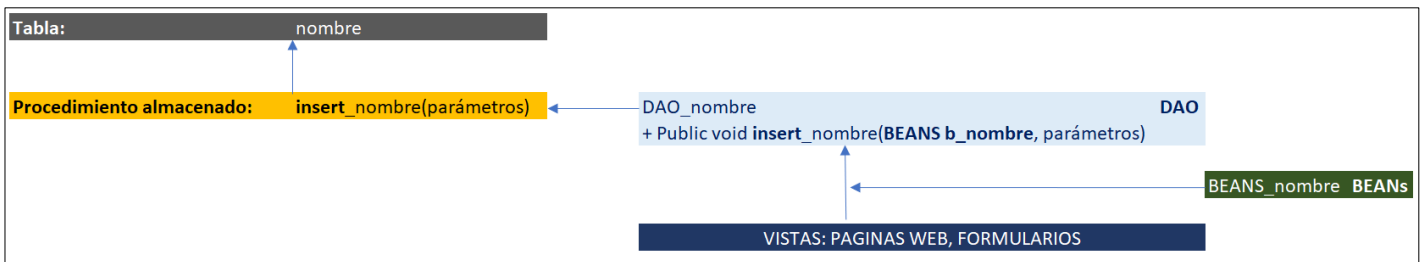


Figura 23 Patrón para el uso del método insert

En la figura 22, Patrón para el uso del método insert, se observa que el flujo se inicia en las vistas, que remitirán los datos al método insert, para el conjunto de datos se utilizará el Beans y para los parámetros se utilizará variables.

b) Método delete

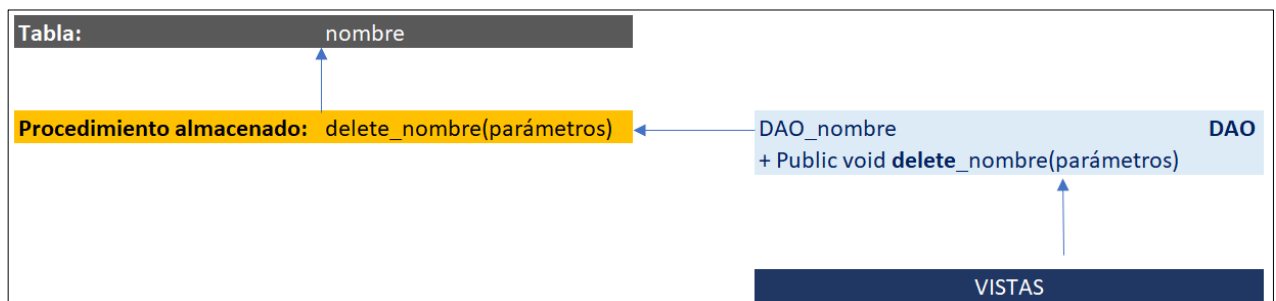


Figura 24 Patrón para el uso del método delete

En la figura 23, Patrón para el uso del método delete, se observa que el flujo se inicia en las vistas, que remitirán los parámetros al método insert, para los parámetros se utilizará variables.

c) Método update



Figura 25 Patrón para el uso del método delete

En la figura 24, Patrón para el uso del método update, se observa que el flujo se inicia en las vistas, que remitirán los datos al método insert, para el conjunto de datos se utilizará el Beans y para los parámetros se utilizará variables.

d) Método select con parámetros

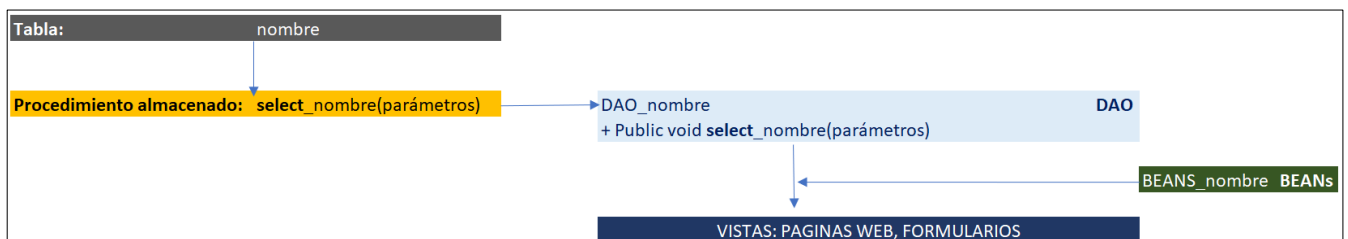


Figura 26 Patrón para el uso del método Select con parámetro

En la figura 25, Patrón para el uso del método Select con parámetros, se observa que el procedimiento recupera los datos según los parámetros que recibió, el DAO recibe cada registro como BEANS y lo adiciona a una lista que retornará los datos a la vista.

e) Método select

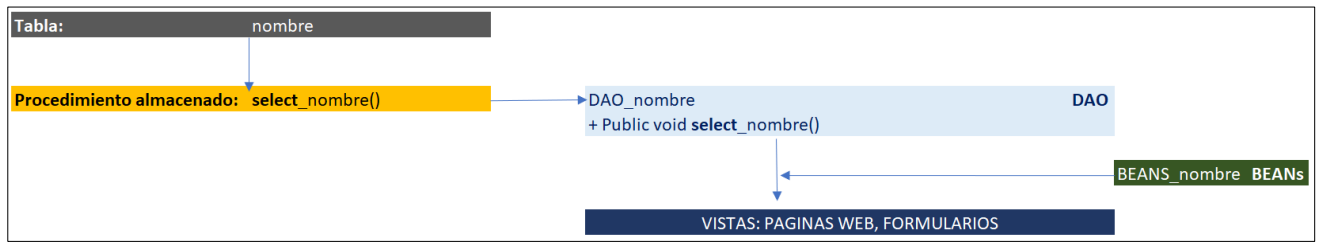


Figura 27 Patrón para el uso del método Select

En la figura 26, Patrón para el uso del método Select, se observa que el procedimiento recupera los datos, el DAO recibe cada registro como BEANS y lo adiciona a una lista que retornará los datos a la vista.

f) Método select cálculos

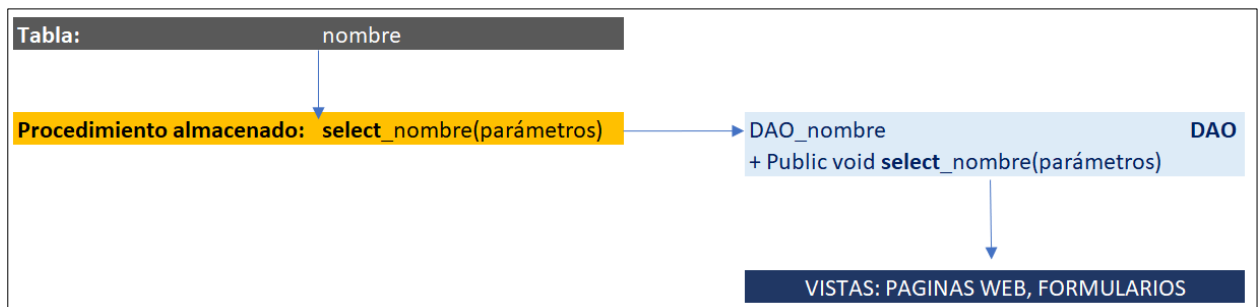


Figura 28 Patrón para el uso del método Select cálculos

En la figura 27, Patrón para el uso del método Select que está relacionado con el procedimiento almacenado que realiza un cálculo y, el DAO recupera el dato, por lo generar el dato es tipo numérico, se almacena en una variable y es remitida a la vista.

f) Método select cálculos

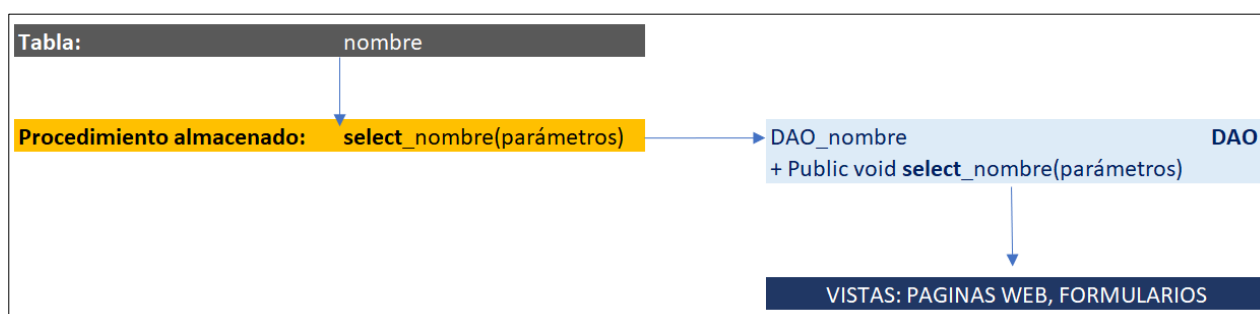


Figura 29 Patrón para el uso del método Select cálculos

En la figura 28, Patrón para el uso del método Select cálculos, que cuenta la cantidad de registros, el DAO recupera el dato tipo numérico en un variable y es remitida a la vista.

Caso especial de consultas con múltiples tablas relacionadas

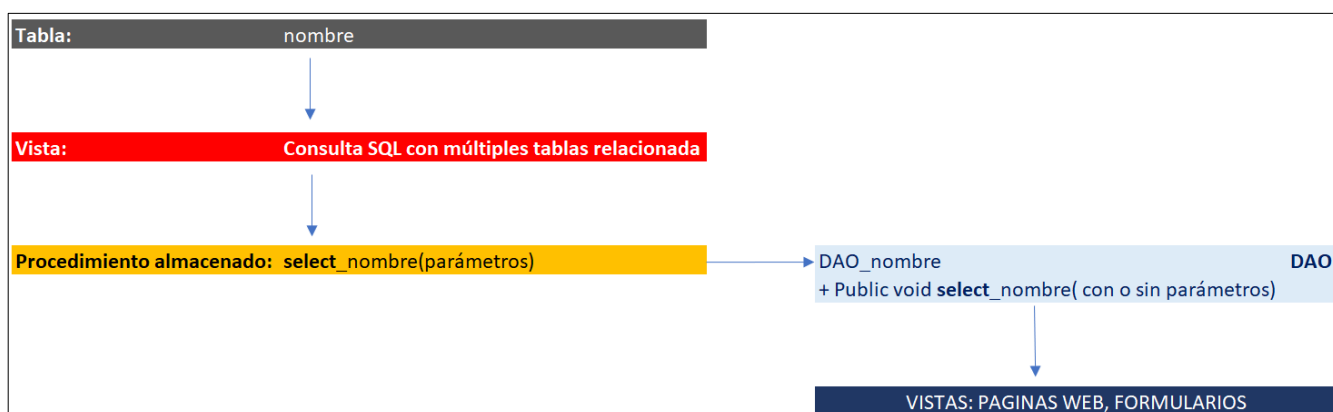


Figura 30 Patrón para caso especial de consultas con múltiples tablas relacionadas

En la figura 29, Patrón para caso especial de consultas con múltiples tablas relacionadas, estas consultas se desarrollan generando los resultados como una vista, luego se desarrolla el procedimiento almacenado a la vista, el DAO recupera mediante el select correspondiente el conjunto de datos, o el dato tipo numérico en un variable y es remitida a la vista del usuario.

4.5 Desarrollo

4.5.1 Arquitectura de la Herramienta de Desarrollo

Criterios para el diseño:

- a) La herramienta no debería tener conexión a base de datos
- b) El ámbito de la herramienta permitirá conexión a base de datos tipos MySQL
- c) La herramienta debía desarrollarse en entorno web
- d) La herramienta debe ejecutarse en servidores Tomcat, Tomee, Glassfish y Payara
- e) Tener presente el principio KISS que establece los sistemas funcionan mejor si se mantienen simples.
- f) El usuario tiene base de datos creadas en su servidor MySQL

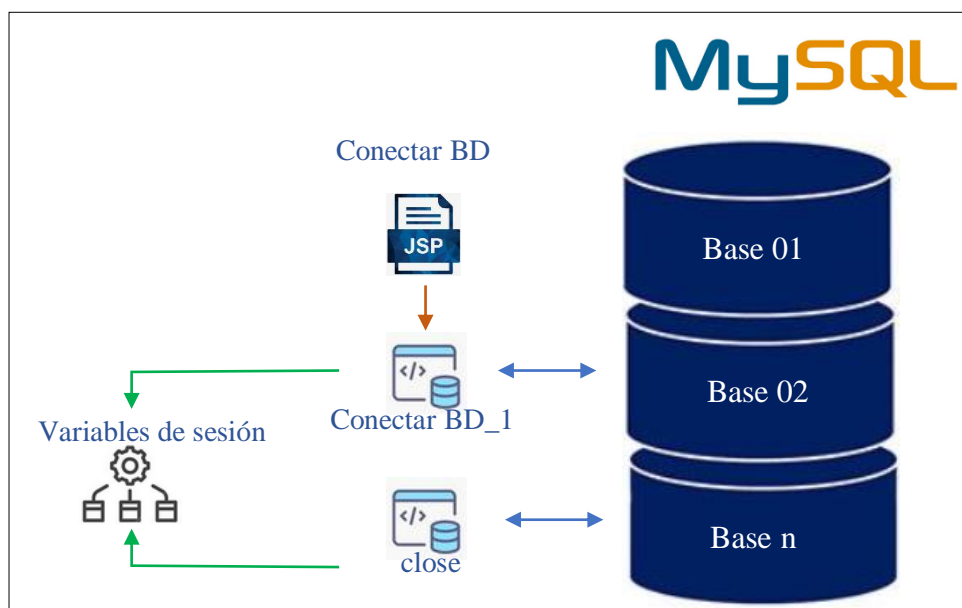


Figura 31 Arquitectura de conexión a las bases de datos de las herramientas parte 1

Conectar BD: Es una vista que permite al usuario a través de la página conectar_bd.jsp debe ingresar los parámetros para conectarse a la base de datos: usuario, clave, base de datos, host, puerto y la clase del driver.

Conectar BD_1: es un script conectar_bd_1.jsp para realizar la conexión a la base de datos y crear las variables de sesión.

Close: es un script close.jsp para anular las variables de sesión.

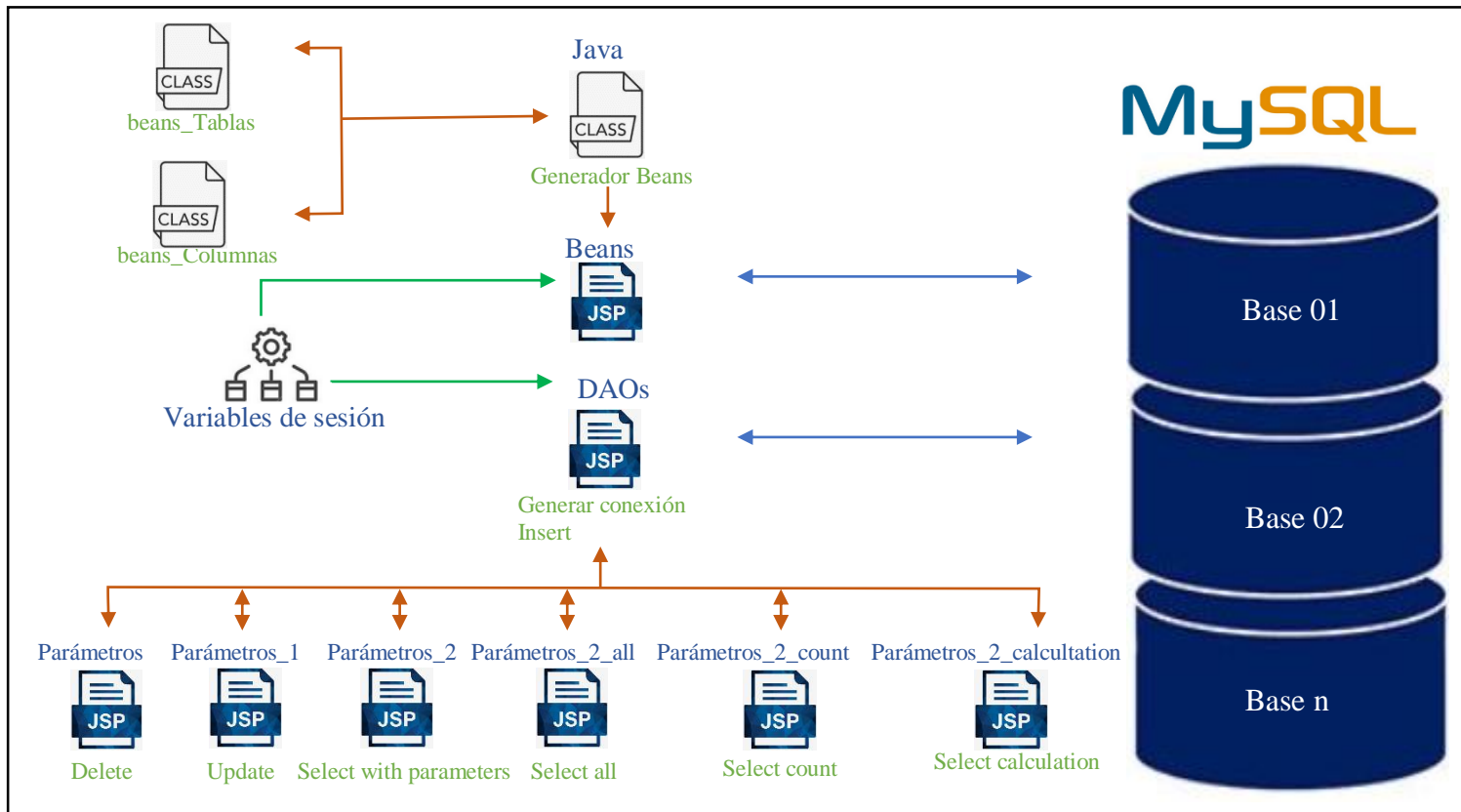


Figura 32 Arquitectura de conexión a las bases de datos de las herramientas parte 2

Beans: es una vista que permite seleccionar la tabla o vista para generar el código del beans de la tabla.

DAOs: es una vista que permite seleccionar las tablas o vista para generar el código de los métodos del DAO, el resultado de los códigos de los métodos se presenta en esta vista. Los códigos que no necesitan parámetros al ser invocados se generan sin necesidad de otra vista, estos son el código de la clase que permite conectarse a una base de datos necesaria para los otros métodos y el código del método insert. Los métodos que necesitan parámetros utilizan otras vistas.

Parámetros: es una vista que permite al usuario crear los parámetros necesarios para el método Delete, construye el método y lo retorna a la vista DAOs.

Parámetros_1: es una vista que permite al usuario crear los parámetros necesarios para el método Update, construye el método y lo retorna a la vista DAOs.

Parámetros_2: es una vista que permite al usuario crear los parámetros necesarios para el método Select, construye el método y lo retorna a la vista DAOs.

Parámetros_2_all: es una vista que permite al usuario crear el método Update que no necesita parámetros, construye el método y lo retorna a la vista DAOs.

Parámetros_2_count: es una vista que permite al usuario crear los parámetros necesarios para el método Select que retorna un valor numérico de tipo entero resultado de un procedimiento de contar, construye el método y lo retorna a la vista DAOs.

Parámetros_2_calculation: es una vista que permite al usuario crear los parámetros necesarios para el método Select que retorna un valor numérico de tipo double resultado de procedimiento de cálculo, construye el método y lo retorna a la vista DAOs.

GeneradorBeans: es una clase que permitirá recuperar todas las tablas y vistas de la base de datos y los campos de la tabla o vista seleccionada.

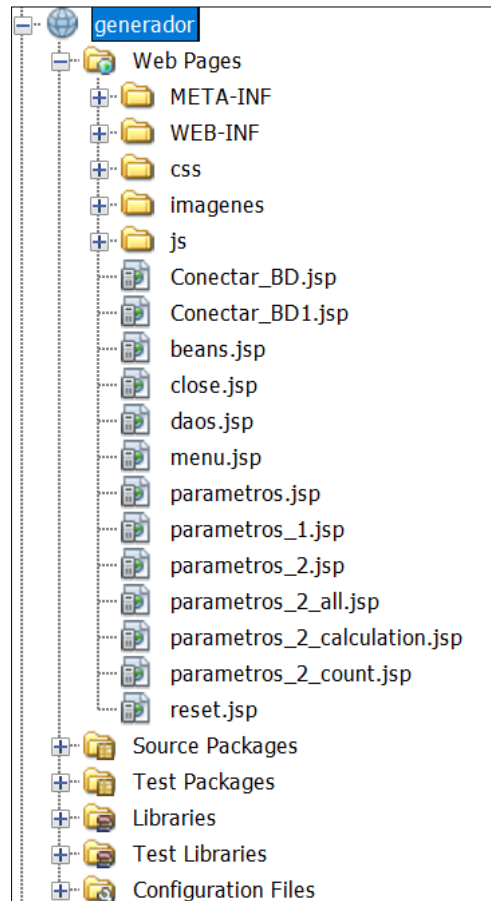


Figura 33 Proyecto de Herramienta de generación de script para Beans y DAOS en proyectos de Lenguaje Java

La herramienta se desarrolló mediante el proyecto “Generador”, en que se priorizó el cumplimiento de los requerimientos funcionales y se utilizó los estilos que fueron necesarios.

4.5.2 Casos de uso

Requerimiento: REQ001 Conectar base de datos
Vista: Conectar_BD.jsp
Objetivos: Mostrar el resultado de la conexión a la base de datos: satisfactoria o no satisfactoria
Pre condiciones: Las variables de sesión usuario, clave, base de datos, host, puerto y clase se han creado.
Acciones: <ol style="list-style-type: none">1. Crear las variables locales para recibir las variables de sesión Conectar_BD_1.jsp2. Si las variables de sesión son diferentes de null, recibir su valor3. Si la variable de sesión resultado es 1, asignar a la variable global el valor “Conexión satisfactoria”, sino “Conexión no satisfactoria”4. Mostrar los valores de las variables de sesión en los elementos input y el resultado de la conexión.5. El usuario ingresa los valores para las variables de sesión6. El usuario remite los valores de las variables de sesión al script de conexión.
Post Condiciones Muestra el resultado de la conexión: Satisfactoria o No satisfactoria.

Requerimiento: REQ002 Cerrar conexión
Script: Conectar_BD_1.jsp
Objetivos: Anular las variables de sesión.
Pre condiciones: Las variables de sesión usuario, clave, base de datos, host, puerto y clase contienen valor.
Acciones: <ol style="list-style-type: none"> 1. Anular las variables de sesión 2. Redireccionar al menu.jsp
Post Condiciones

Requerimiento: REQ003 Generar BEANs
Java: generadorBeans.java
Objetivos: Recuperar el listado de tablas y vistas de una base de datos. Recuperar el listado de las columnas de una tabla
Pre condiciones: Las variables de sesión usuario, clave, base de datos, host, puerto y clase contienen valor. Se ha creado un beans_Tablas que va contener el nombre de una tabla recuperada
Acciones: Método recuperar tablas <ol style="list-style-type: none"> 1. Con las variables de sesión, se realiza la conexión a la base de datos 2. Crear una lista del tipo Beans_Tabla 3. Recuperar el listado de tablas y vistas de una tabla 4. Retorna la lista a la vista beans.jsp Método recuperar campos de una tabla <ol style="list-style-type: none"> 1. Con las variables de sesión, se realiza la conexión a la base de datos 2. Crear una lista del tipo Beans_Columnas 3. Recupera los campos y tipo de campos de una tabla 4. Retorna la lista a la vista beans.jsp
Post Condiciones

Requerimiento: REQ003.REQ001 Mostrar tablas
Vista: beans.jsp
Objetivos: Mostrar el listado de tablas y vistas de una base de datos.
Pre condiciones: El listado de tablas se ha recuperado en generadorBeans.java
Acciones: <ol style="list-style-type: none"> 1. Crear las variables locales para recibir las variables de sesión 2. Recibir variables de sesión y asignarles el valor de las variables locales 3. Si el resultado de la conexión es satisfactorio, recuperar el listado de las tablas 4. Mostrar las tablas y vistas de la base de datos en beans.jsp
Post Condiciones

Requerimiento: REQ003.REQ002 Mostrar campos de las tablas
Vista: beans.jsp
Objetivos: Mostrar el listado de campos de una tabla seleccionada.
Pre condiciones: El listado de tablas se ha visualiza en beans.jsp
Acciones: <div> 5. El usuario selecciona el nombre de una tabla y lo redirecciona a beans.jsp </div> <div> 6. Si existe tabla seleccionada, recuperar el listado de las columnas </div> <div> 7. Mostrar las columnas y vistas de la base de datos en beans.jsp </div>
Post Condiciones

Requerimiento: REQ003.REQ003 Generar Script BEANs
Vista: beans.jsp
Objetivos: Generar el script de beans de la tabla seleccionada.
Pre condiciones: El listado de los campos de una tabla seleccionada se ha visualiza en beans.jsp
Acciones: <div> 8. El usuario selecciona los campos para el generar el beans y lo redirecciona a beans.jsp 9. Captura el número de campos seleccionados 10. Genera el código del beans en el elemento correspondiente de la vista beans.jsp </div>
Post Condiciones

Requerimiento: REQ004 Generar DAOS
Java: generadorBeans.java
Objetivos: Recuperar el listado de tablas y vistas de una base de datos. Recuperar el listado de las columnas de una tabla
Pre condiciones: Las variables de sesión usuario, clave, base de datos, host, puerto y clase contienen valor. Se ha creado un beans_Tablas que va contener el nombre de una tabla recuperada
Acciones: Método recuperar tablas <ol style="list-style-type: none"> 5. Con las variables de sesión, se realiza la conexión a la base de datos 6. Crear una lista del tipo Beans_Tabla 7. Recuperar el listado de tablas y vistas de una tabla 8. Retorna la lista a la vista beans.jsp Método recuperar campos de una tabla <ol style="list-style-type: none"> 1. Con las variables de sesión, se realiza la conexión a la base de datos 2. Crear una lista del tipo Beans_Columnas 3. Recupera los campos y tipo de campos de una tabla 4. Retorna la lista a la vista beans.jsp
Post Condiciones

Requerimiento: REQ004.REQ001 Mostrar tablas
Vista: daos.jsp
Objetivos: Mostrar el listado de tablas y vistas de una base de datos.
Pre condiciones: El listado de tablas se ha recuperado en generadorBeans.java
Acciones: <ol style="list-style-type: none"> 1. Crear las variables locales para recibir las variables de sesión 2. Recibir variables de sesión y asignarles el valor de las variables locales 3. Si el resultado de la conexión es satisfactorio, recuperar el listado de las tablas 4. Mostrar las tablas y vistas de la base de datos en beans.jsp
Post Condiciones

Requerimiento: REQ004.REQ002 Construir clase conexión
Vista: daos.jsp
Objetivos: Generar el script de la clase conexión.
Pre condiciones: Las variables de sesión se han inicializado.
Acciones: <div> 5. Verificar que el usuario ha seleccionado la opción para generar la clase conexión. </div> <div> 6. Genera el código de la clase conexión en el elemento correspondiente de la vista daos.jsp </div>
Post Condiciones

Requerimiento: REQ004.REQ003 Construir DAO insertar
Vista: daos.jsp
Objetivos: Generar el script del método de inserción.
Pre condiciones: Las variables de sesión se han inicializado.
Acciones: <div> 7. Verificar que el usuario ha seleccionado la opción para generar el método insertar. </div> <div> 8. Genera el código del método insertar en el elemento correspondiente de la vista daos.jsp </div>
Post Condiciones

Requerimiento: REQ004.REQ004 Construir DAO eliminar
Vista: parametros.jsp (ventana emergente)
Objetivos: Generar el script del método de eliminación.
Pre condiciones: Las variables de sesión se han inicializado. El usuario ha seleccionado la opción de eliminación en daos.jsp
Acciones: <ol style="list-style-type: none"> 1. Recuperar el nombre de la tabla donde se aplicará el método delete. 2. El usuario agrega elementos html para definir los parámetros del método delete, esto se realiza mediante una función JavaScript 3. Genera el código del método delete 4. Asigna el código del método delete el elemento correspondiente en la página daos.jsp 5. Cierra la venta emergente parametros.jsp y retorna a la vista daos.jsp
Post Condiciones

Requerimiento: REQ004.REQ005 Construir DAO actualizar
Vista: parámetros_1.jsp (ventana emergente)
Objetivos: Generar el script del método de actualización.
Pre condiciones: Las variables de sesión se han inicializado. El usuario ha seleccionado la opción de actualización en daos.jsp
Acciones: <ol style="list-style-type: none"> 1. Recuperar el nombre de la tabla donde se aplicará el método actualizar. 2. El usuario agrega elementos html para definir los parámetros del método delete, esto se realiza mediante una función JavaScript 3. Genera el código del método delete 4. Asigna el código del método delete el elemento correspondiente en la página daos.jsp 5. Cierra la venta emergente parámetros_1.jsp y retorna a la vista daos.jsp
Post Condiciones

<p>Requerimiento: REQ004.REQ006 Construir DAO seleccionar con parámetros</p>
<p>Vista: parámetros_2.jsp (ventana emergente)</p>
<p>Objetivos:</p> <p>Generar el script del método de selección con parámetros.</p>
<p>Pre condiciones:</p> <p>Las variables de sesión se han inicializado.</p> <p>El usuario ha seleccionado la opción de Selección con parámetros en daos.jsp</p>
<p>Acciones:</p> <ol style="list-style-type: none"> 1. Recuperar el nombre de la tabla donde se aplicará el método select con parámetros. 2. El usuario agrega elementos html para definir los parámetros del método select con parámetros, esto se realiza mediante una función JavaScript 3. Genera el código del método select con parámetros 4. Asigna el código del método select con parámetros al elemento correspondiente en la página daos.jsp 5. Cierra la venta emergente parámetros_2.jsp y retorna a la vista daos.jsp
<p>Post Condiciones</p>

Requerimiento: REQ004.REQ007 Construir DAO seleccionar todos
Vista: parámetros_2_all.jsp (ventana emergente)
Objetivos: Generar el script del método de selección todos.
Pre condiciones: Las variables de sesión se han inicializado. El usuario ha seleccionado la opción de Selección todos en daos.jsp
Acciones: <ol style="list-style-type: none"> 1. Recuperar el nombre de la tabla donde se aplicará el método select todos. 2. Genera el código del método select 3. Asigna el código del método select al elemento correspondiente en la página daos.jsp 4. Cierra la venta emergente parámetros_2_all.jsp y retorna a la vista daos.jsp
Post Condiciones

Requerimiento: REQ004.REQ007 Construir DAO seleccionar contar
Vista: parámetros_2_count.jsp (ventana emergente)
Objetivos: Generar el script del método de selección contar.
Pre condiciones: Las variables de sesión se han inicializado. El usuario ha seleccionado la opción de Selección contar en daos.jsp
Acciones: <ol style="list-style-type: none"> 1. Recuperar el nombre de la tabla donde se aplicará el método select contar. 2. El usuario agrega elementos html para definir los parámetros del método select contar, esto se realiza mediante una función JavaScript 3. Genera el código del método select para contar 4. Asigna el código del método select contar al elemento correspondiente en la página daos.jsp 5. Cierra la venta emergente parámetros_2_count.jsp y retorna a la vista daos.jsp
Post Condiciones

Requerimiento: REQ004.REQ008 Construir DAO seleccionar de cálculos
Vista: parámetros_2_calculation.jsp (ventana emergente)
Objetivos: Generar el script del método de selección cálculos.
Pre condiciones: Las variables de sesión se han inicializado. El usuario ha seleccionado la opción de Selección contar en daos.jsp
Acciones: <ol style="list-style-type: none"> 1. Recuperar el nombre de la tabla donde se aplicará el método select contar. 2. El usuario agrega elementos html para definir los parámetros del método select contar, esto se realiza mediante una función JavaScript 3. Genera el código del método select para contar 4. Asigna el código del método select contar al elemento correspondiente en la página daos.jsp 5. Cierra la venta emergente parámetros_2_count.jsp y retorna a la vista daos.jsp
Post Condiciones

4.5.3 Maquetado de vistas

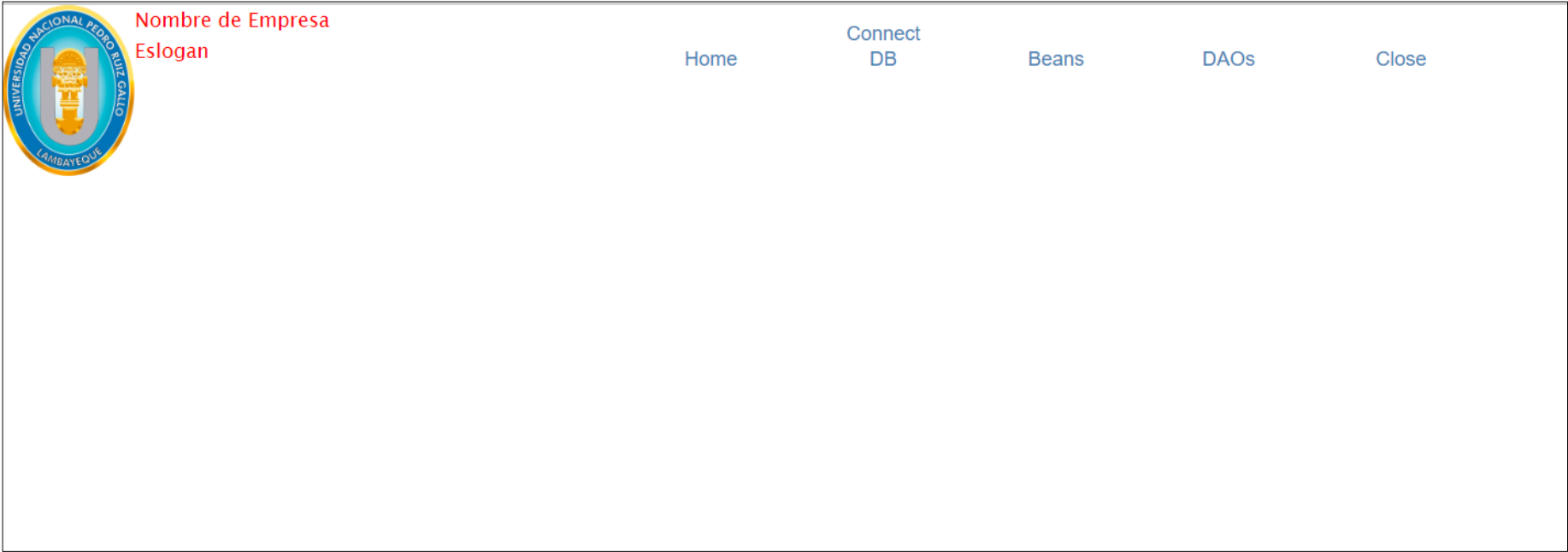


Figura 34 Vista de menu.jsp

La vista de menu.jsp permite al usuario acceder a las opciones que permiten demostrar el cumplimiento de los requerimientos funcionales

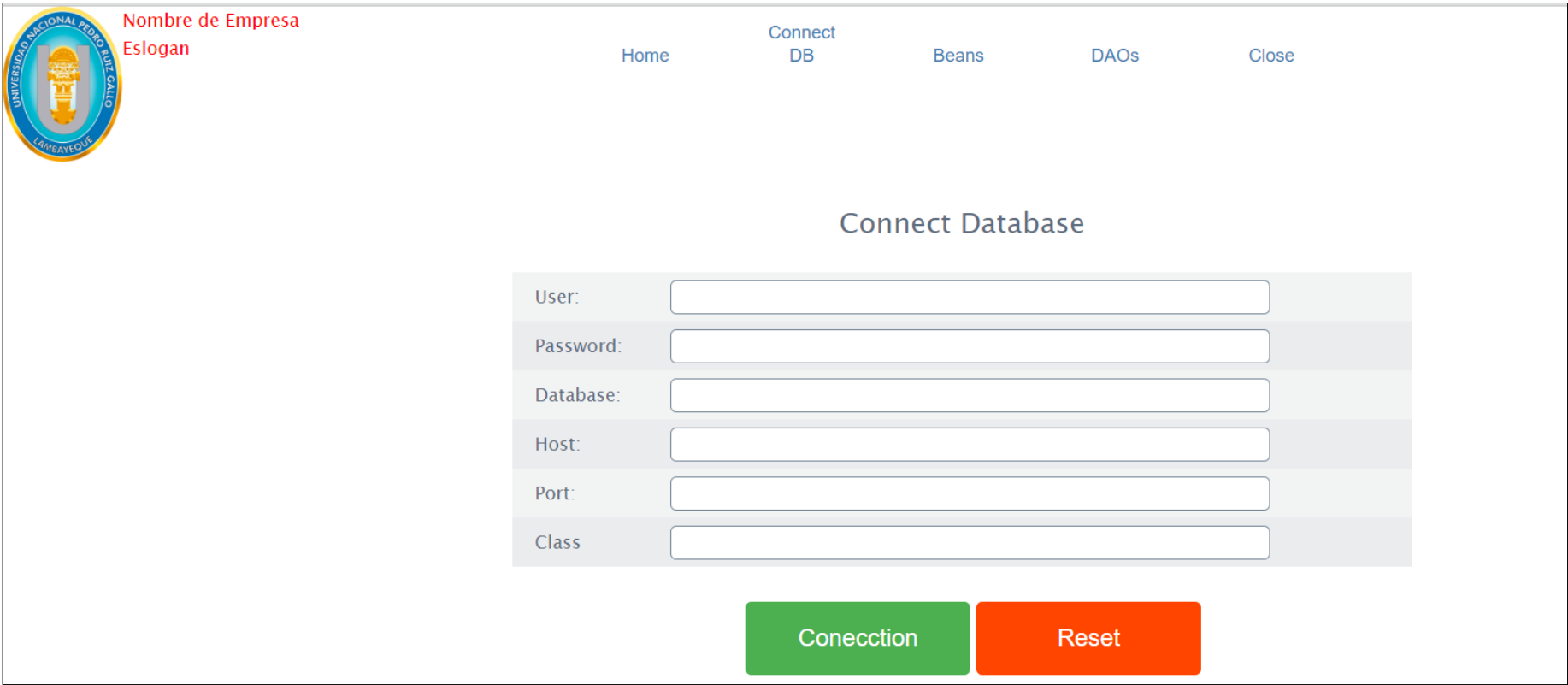


Figura 35 Vista Conectar_BD.jsp

La vista Conectar_BD.jsp corresponde al requerimiento REQ001, Conectar base de datos, el usuario deberá ingresar el usuario, la clave, base de datos, url, puerto y nombre de la clase que le permitirá conectarse a la base de datos.

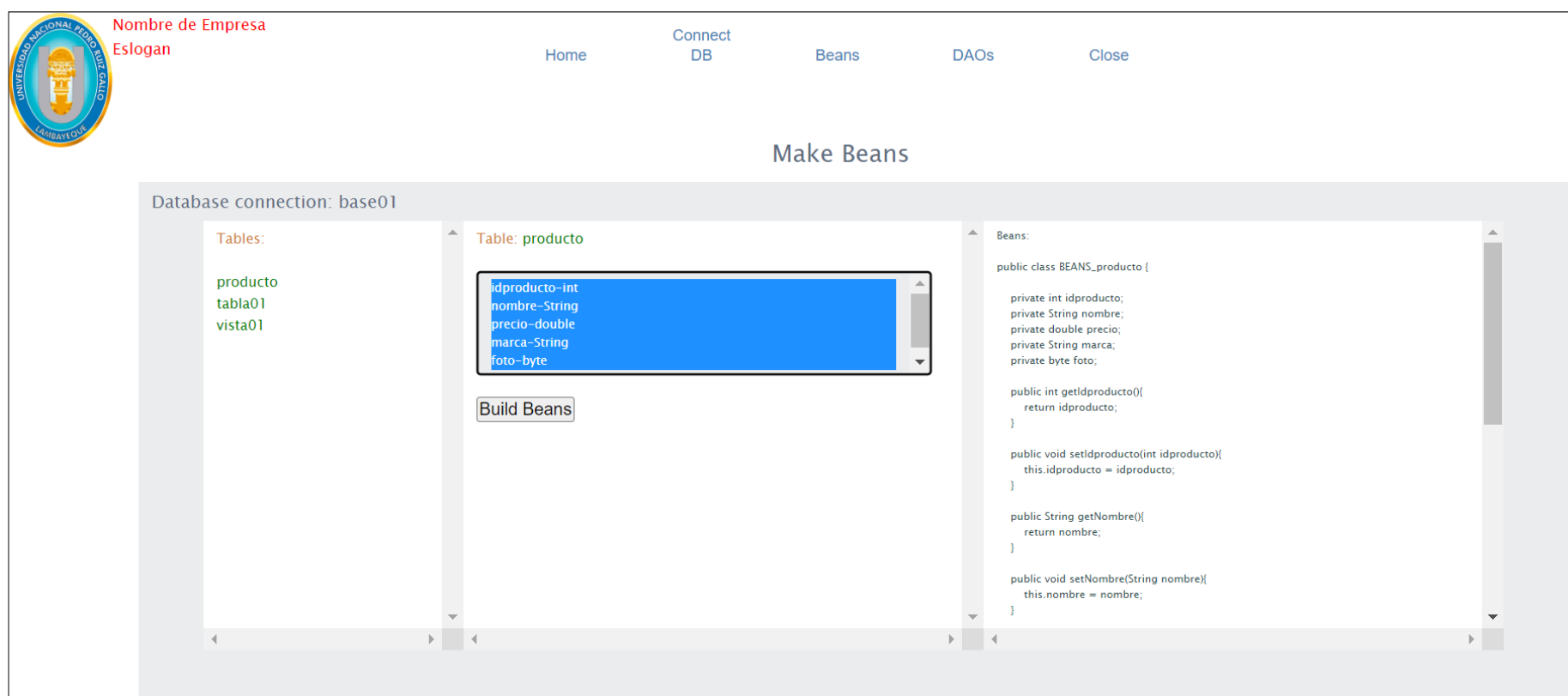


Figura 36 Vista beans.jsp?tabla01=producto

La vista beans.jsp, corresponde al requerimiento REQ003 Generar Beans, en la parte izquierda de la vista el usuario puede seleccionar la tabla o vista para generar el beans cumpliendo el REQ003.001 mostrar tablas, en la parte central el usuario puede seleccionar los campos de la tabla cumpliendo el REQ003.002 y en la parte derecha el script del beans generado cumpliendo el REQ003.003.

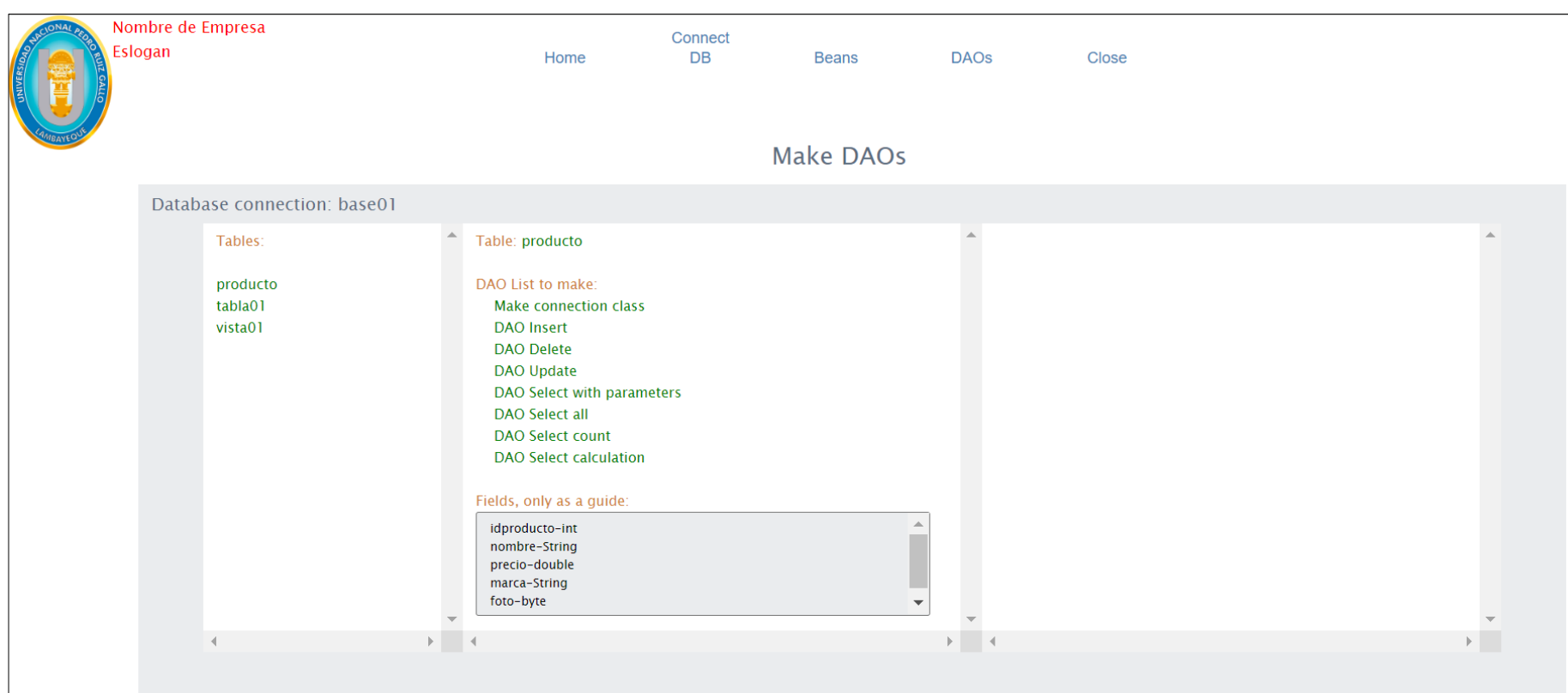


Figura 37 Vista daos.jsp?tabla01=producto

La vista daos.jsp corresponde al requerimiento REQ004 Generar DAOS, en la parte izquierda se presenta el listado de tablas y vistas cumpliendo el requerimiento REQ004.001, en la parte central se presenta l listado de los métodos para que el usuario seleccione, y en la parte inferior de la parte central se lista los campos de la tabla o vista seleccionada.

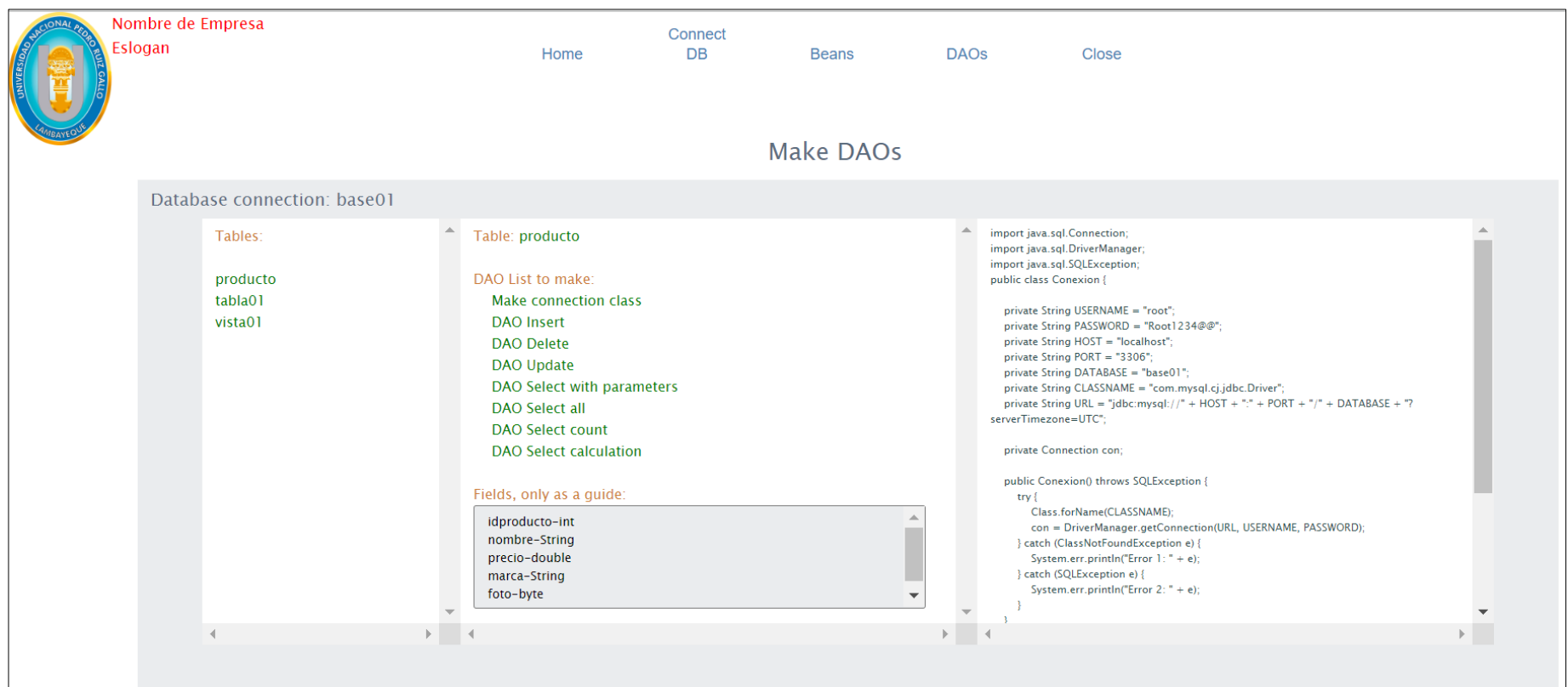


Figura 38 Vista daos.jsp?opcion=1&tabla01=producto

En la vista daos.jsp el usuario puede seleccionar la opción Make connection class, para generar el script de conexión, el resultado se presenta en el lado derecho de la vista, cumpliendo el requerimiento REQ004.REQ002

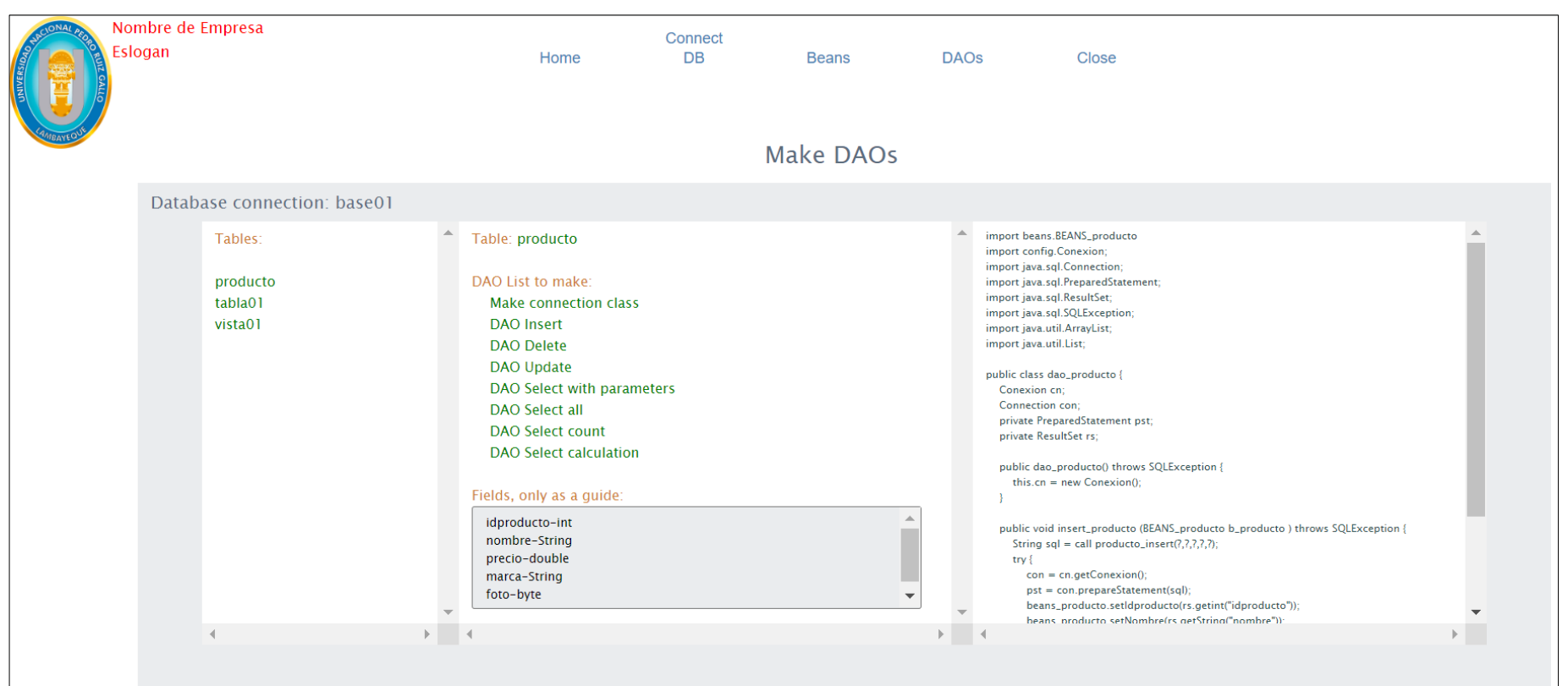


Figura 39 Vista daos.jsp?opcion=2&tabla01=producto

En la vista daos.jsp el usuario puede seleccionar la opción DAO insert, para generar el script de inserción, el resultado se presenta en el lado derecho de la vista, cumpliendo el requerimiento REQ004.REQ003

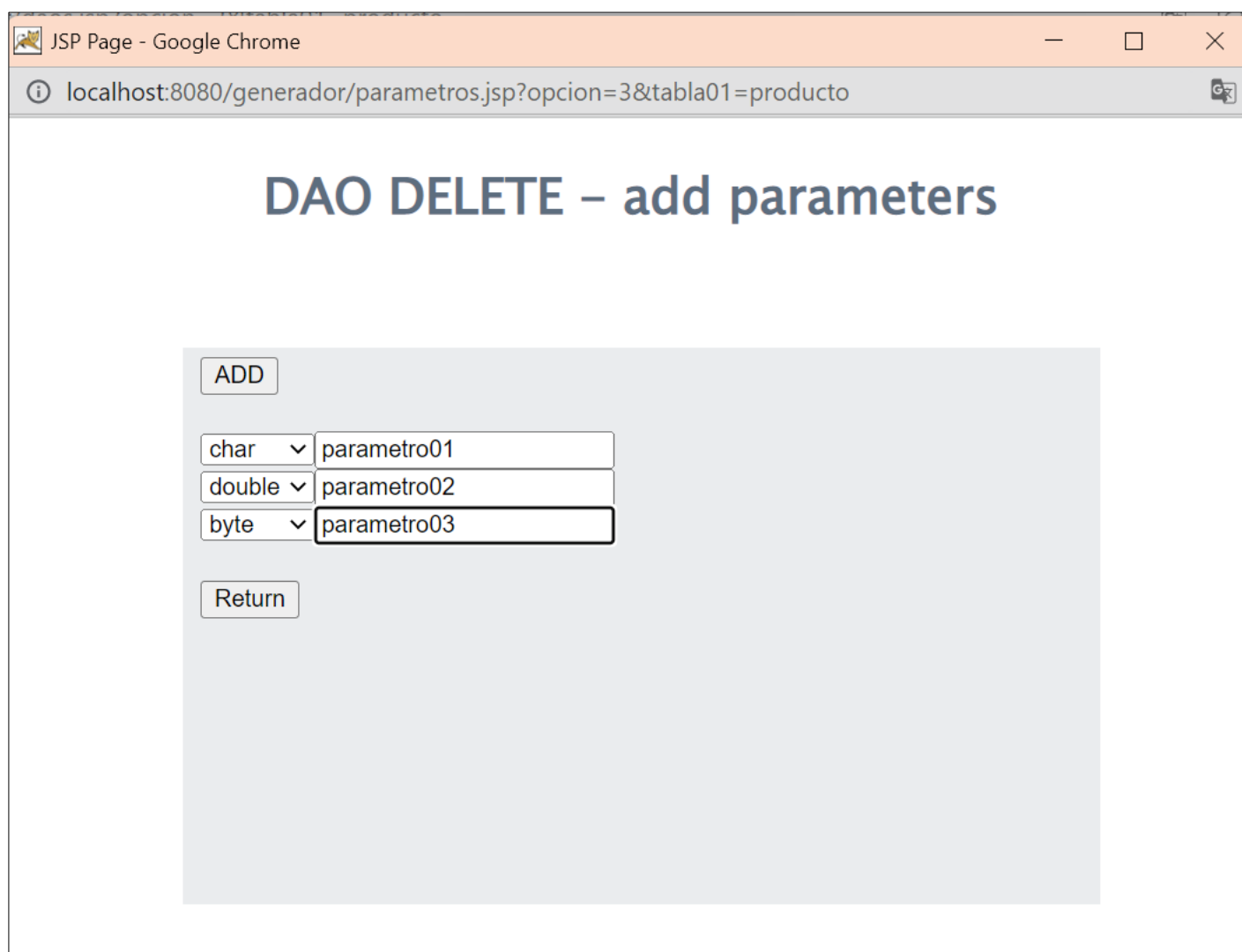


Figura 40 vista parametros.jsp?opcion=3&tabla01=producto

El usuario puede seleccionar la opción DAO Delete, este hipervínculo le abrirá la ventana emergente parametros.jsp al cual le llega el nombre de la tabla, en esta vista el usuario puede agregar los parámetros que se utilizarán en método delete al momento de retornar a la vista daos.jsp

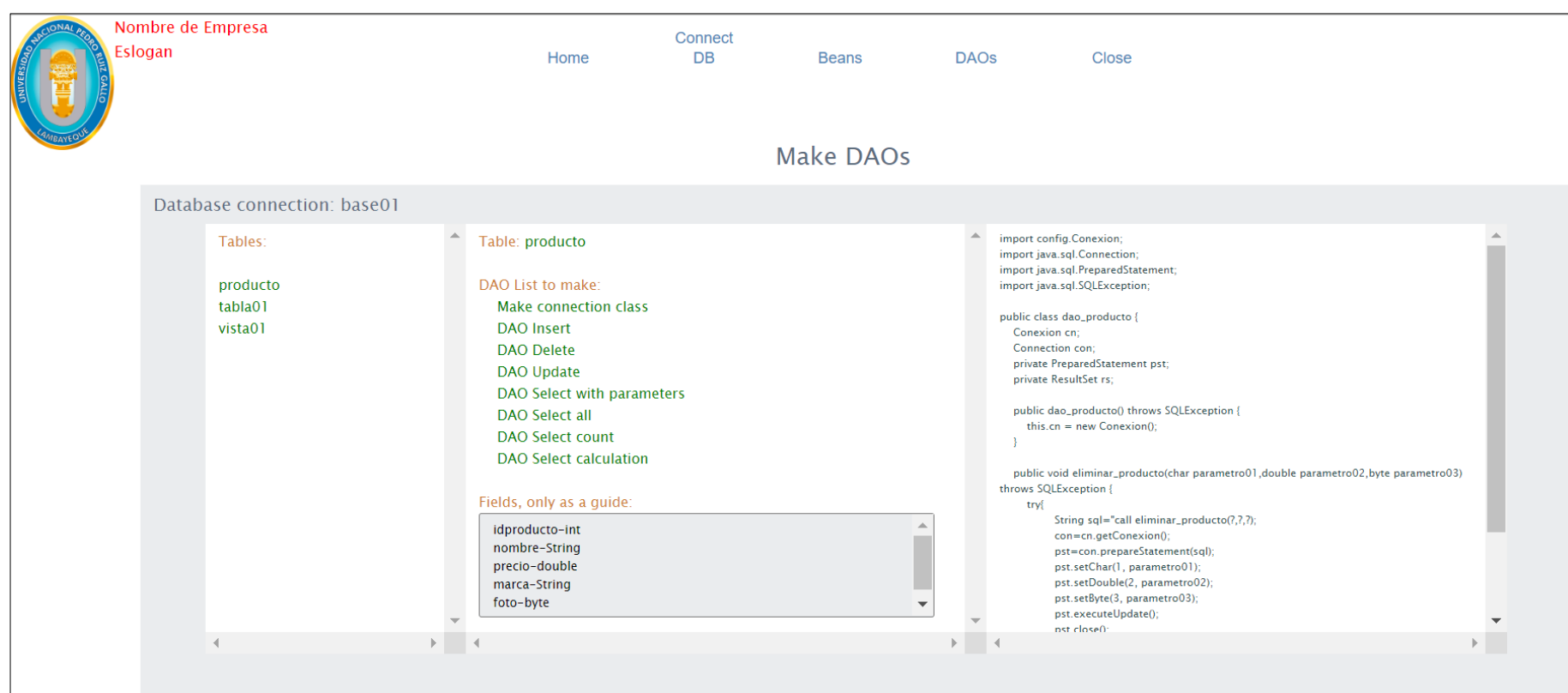


Figura 41 vista daos.jsp con script de método delete

La figura 40, presenta a la vista daos.jsp con el script del método delete, generara en la vista parametros.jsp cumpliendo el requerimiento REQ004.004.

JSP Page - Google Chrome

localhost:8080/generador/parametros_1.jsp?opcion=4&tabla01=producto

DAO UPDATE – add parameters

ADD

idproducto ▾	int ▾	parametro001
nombre ▾	String ▾	parametro002
precio ▾	double ▾	parametro003

☒ ¿Campo principal de la tabla es correlativo?

return

Figura 42 vista parametros_1.jsp?opcion=4&tabla01=producto

La figura 41 presenta la vista parametros_1.jsp en la cual el usuario adicionar los parámetros y relacionarlo con el campo correspondiente, por ejemplo, el parametro001 estará relacionado con el campo idproducto y es de tipo int

Nombre de Empresa
Eslogan

Home Connect DB Beans DAOs Close

Make DAOs

Database connection: base01

Tables:

- producto
- tabla01
- vista01

Table: producto

DAO List to make:

- Make connection class
- DAO Insert
- DAO Delete
- DAO Update
- DAO Select with parameters
- DAO Select all
- DAO Select count
- DAO Select calculation

Fields, only as a guide:

- idproducto-int
- nombre-String
- precio-double
- marca-String
- foto-byte

```
import config.Conexion;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class dao_producto {
    Conexion cn;
    Connection con;
    private PreparedStatement pst;

    public dao_producto() throws SQLException {
        this.cn = new Conexion();
    }

    public void producto_actualizar (BEANS_producto b_producto, int parametro001, String parametro002, double parametro003) throws SQLException {
        try{
            String sql="call producto_actualizar(?,?,?);";
            con=cn.getConexion();
            pst=con.prepareStatement(sql);
            pst.setInt(1,parametro001);
            pst.setString(2,parametro002);
            pst.setDouble(3,parametro003);
            pst.setString(4,b_producto.getNombre());
            pst.setDouble(5,b_producto.getPrecio());
            pst.setString(6,b_producto.getMarca());
        }
    }
}
```

Figura 43 vista daos.jsp con el script del método update

La figura 42, presenta a la vista daos.jsp con el script del método update, generara en la vista parámetros_1.jsp cumpliendo el requerimiento REQ004.005.

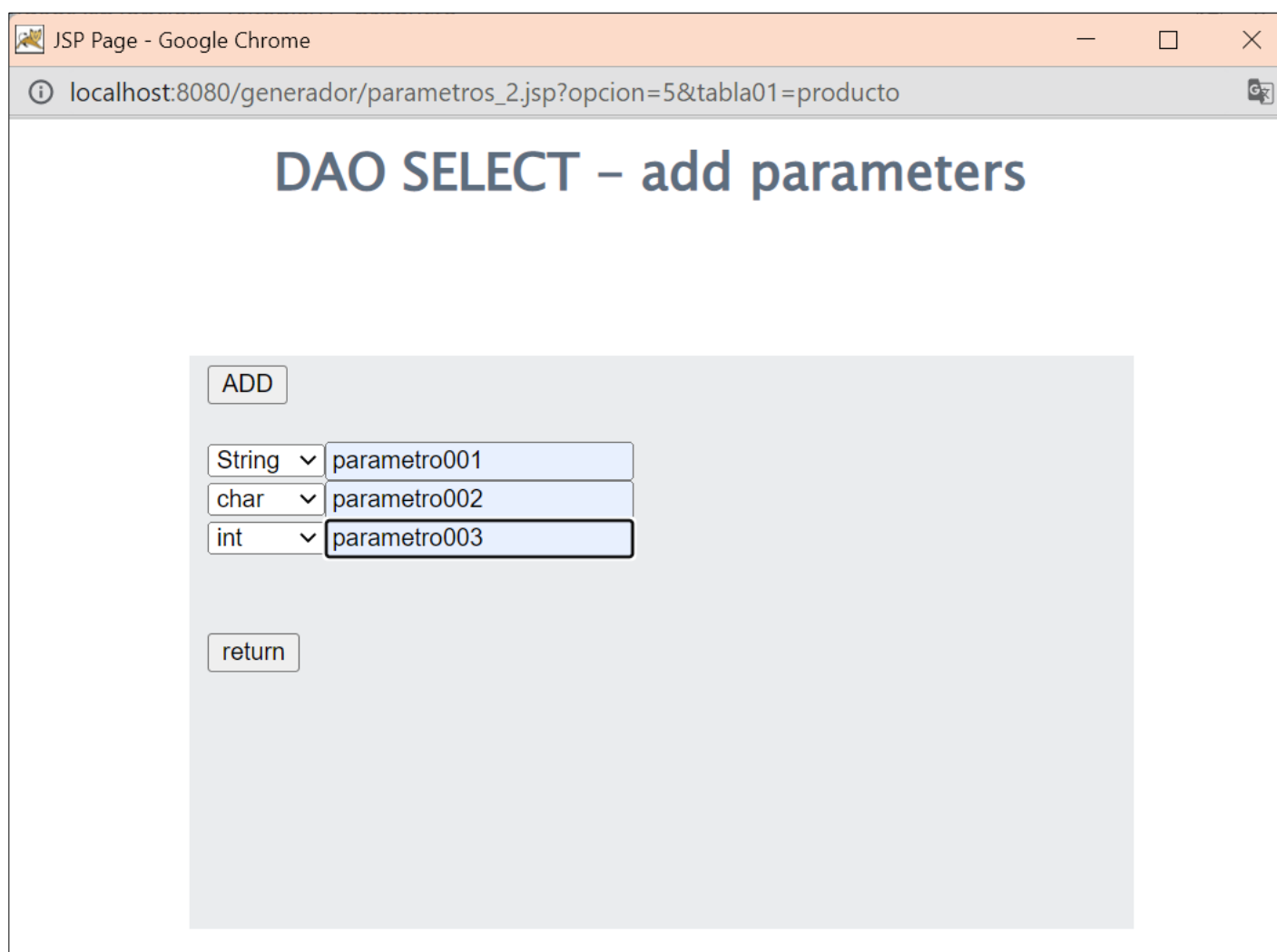


Figura 44 vista parametros_2.jsp?opcion=5&tabla01=producto

La figura 43 presenta la vista parametros_2.jsp en la cual el usuario adicionar los parámetros para el método delete.

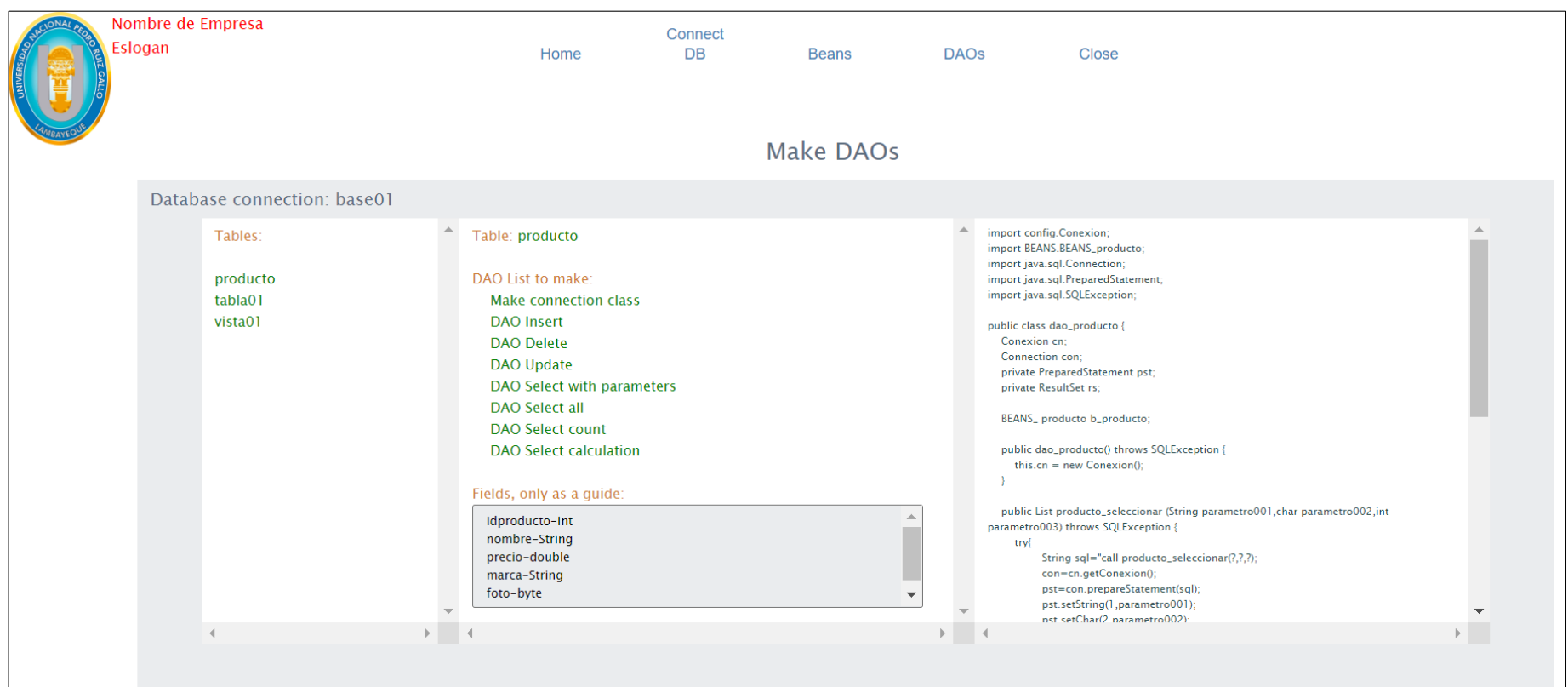


Figura 45 vista daos.jsp con script del método select con parámetros

La figura 44, presenta a la vista daos.jsp con el script del Select con parámetros, generara en la vista parametros_2.jsp cumpliendo el requerimiento REQ004.006.

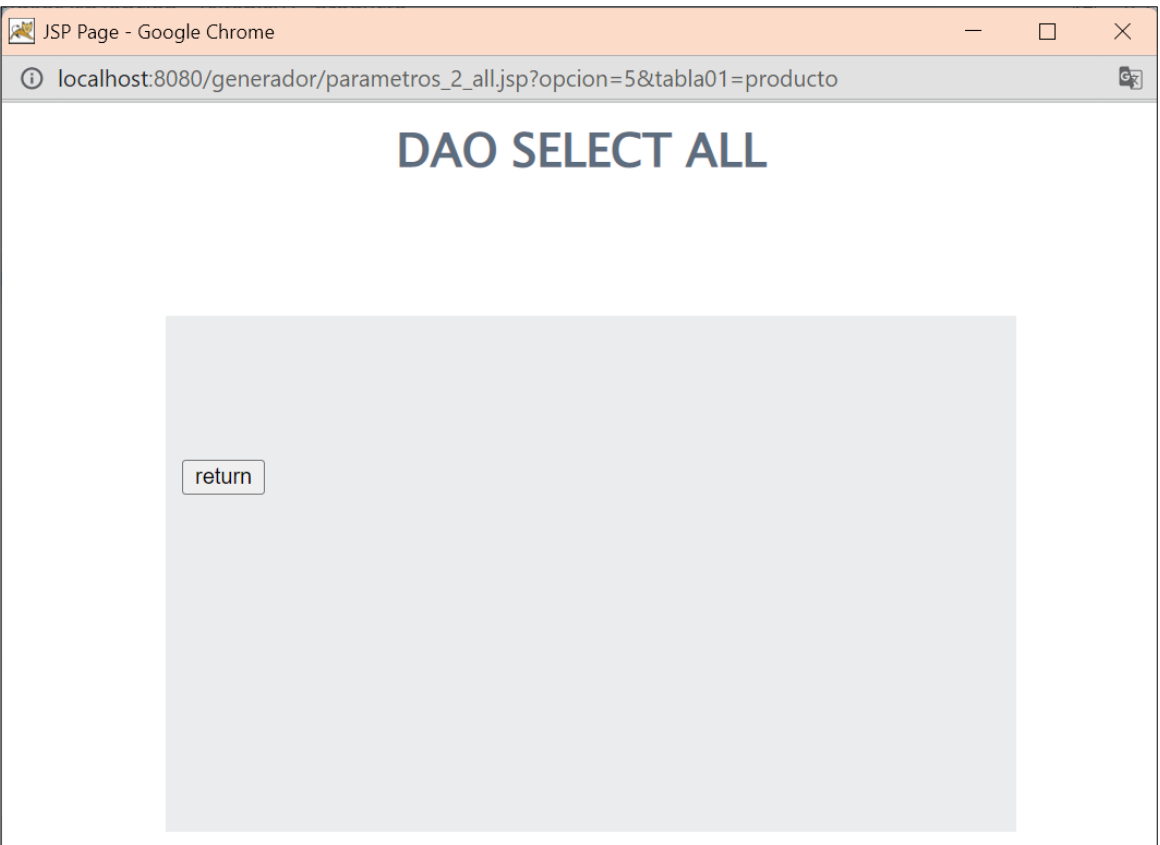


Figura 46 vista parametros_2_all.jsp?opcion=5&tabla01=producto

La figura 45 presenta la vista parametros_2_all.jsp, se diseñó considerando el mismo procedimiento que se utilizó para los otros métodos, no se adicionan parámetros.

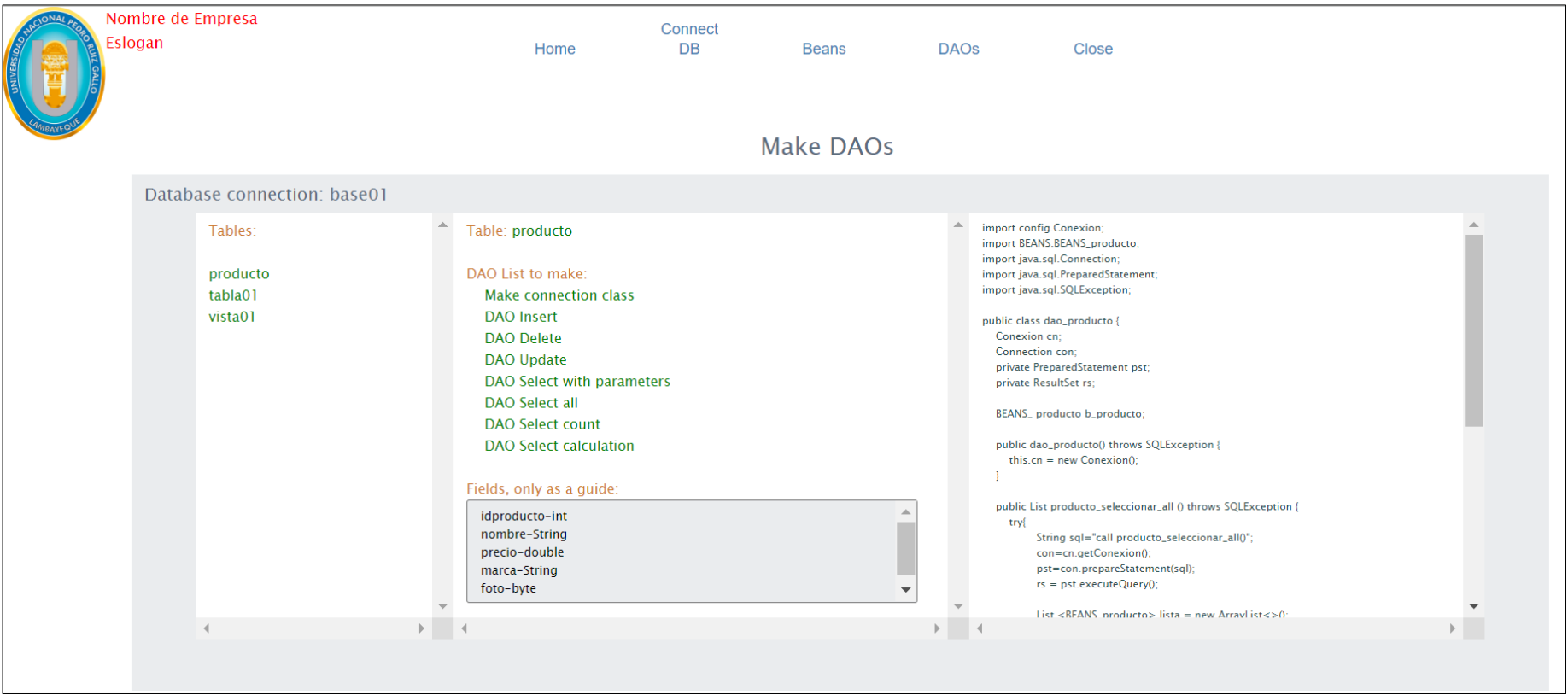


Figura 47 vista con script del método select

La figura 46, presenta a la vista daos.jsp con el script del Select sin parámetros, generara en la vista parametros_2.jsp cumpliendo el requerimiento REQ004.007.

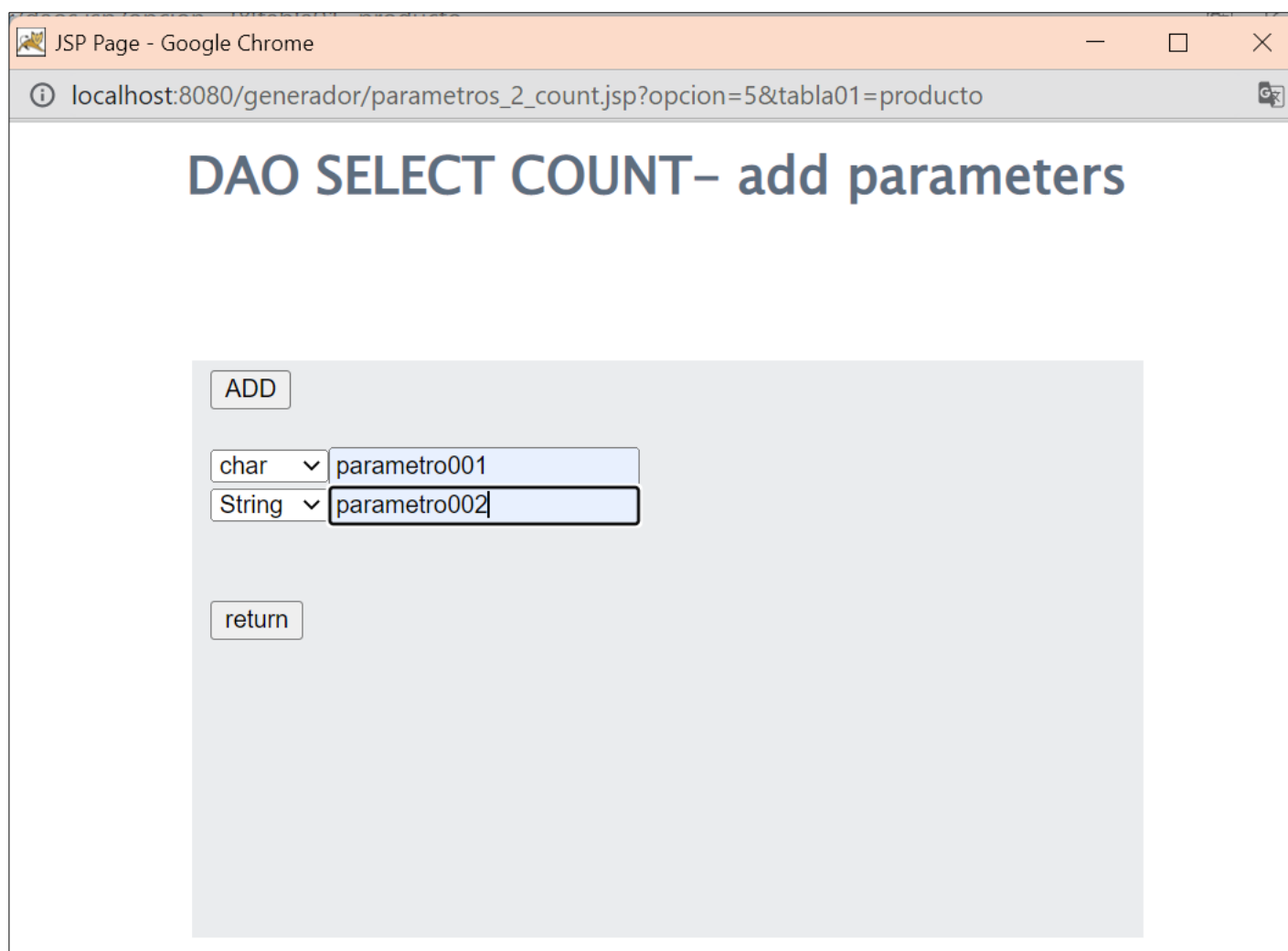


Figura 48 parametros_2_count.jsp?opcion=5&tabla01=producto

La figura 47 presenta la vista parametros_2_count.jsp en la cual el usuario adiciona los parámetros para el método select que permite contar una cantidad de registros.

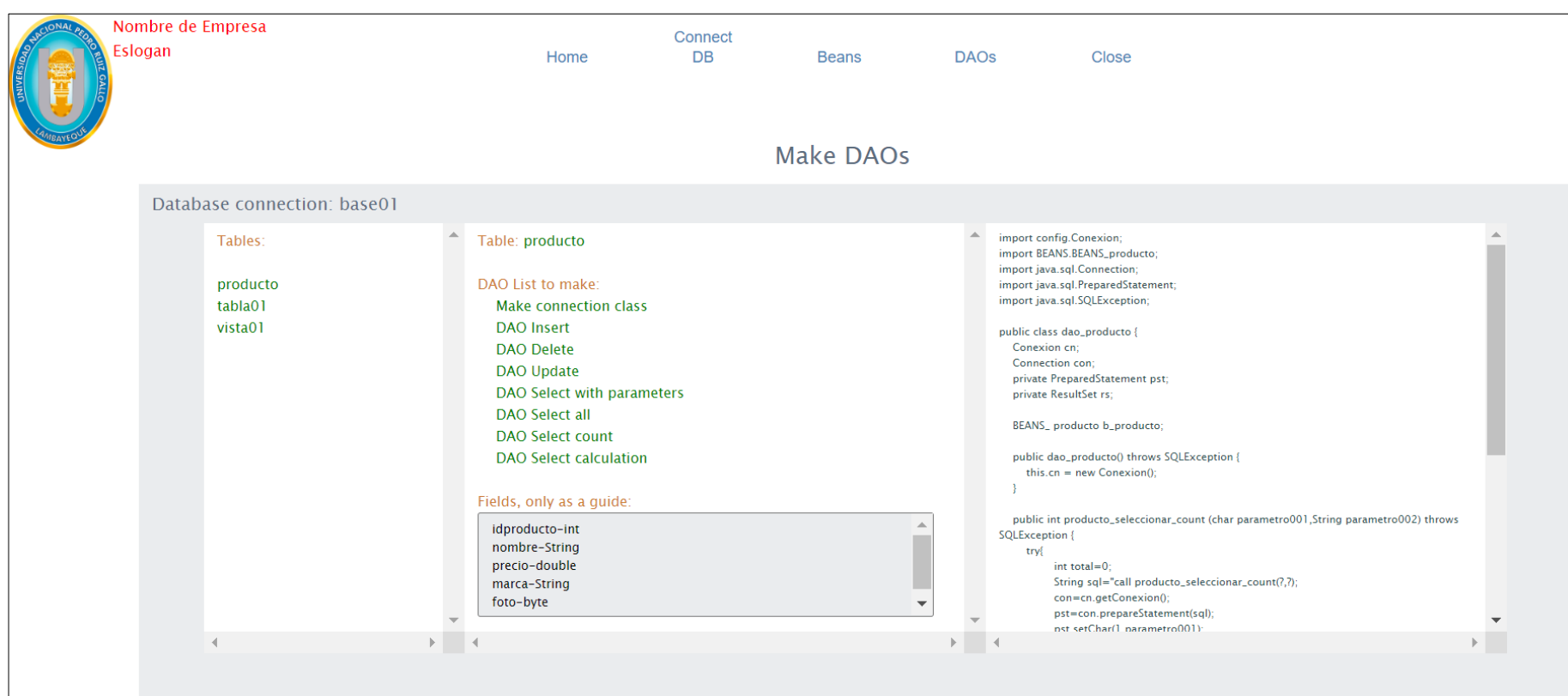


Figura 49 vista daos.jsp con script de select count

La figura 48, presenta a la vista daos.jsp con el script del Select count, generara en la vista parametros_2_count.jsp cumpliendo el requerimiento REQ004.008.

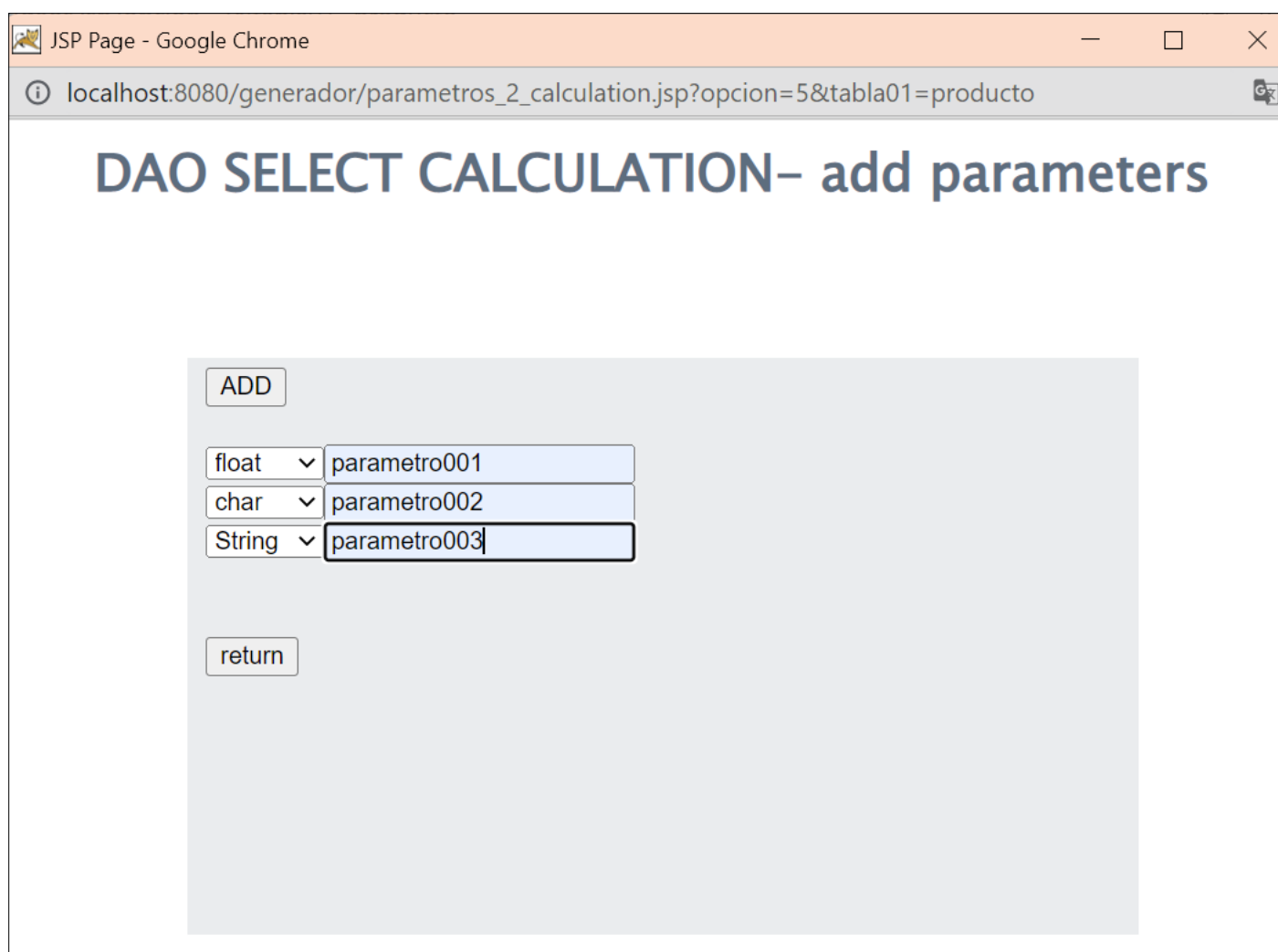


Figura 50 parametros_2_calculation.jsp?opcion=5&tabla01=producto

La figura 49 presenta la vista parametros_2_calculation.jsp en la cual el usuario adiciona los parámetros para el método select que permite recuperar el resultado de un cálculo.

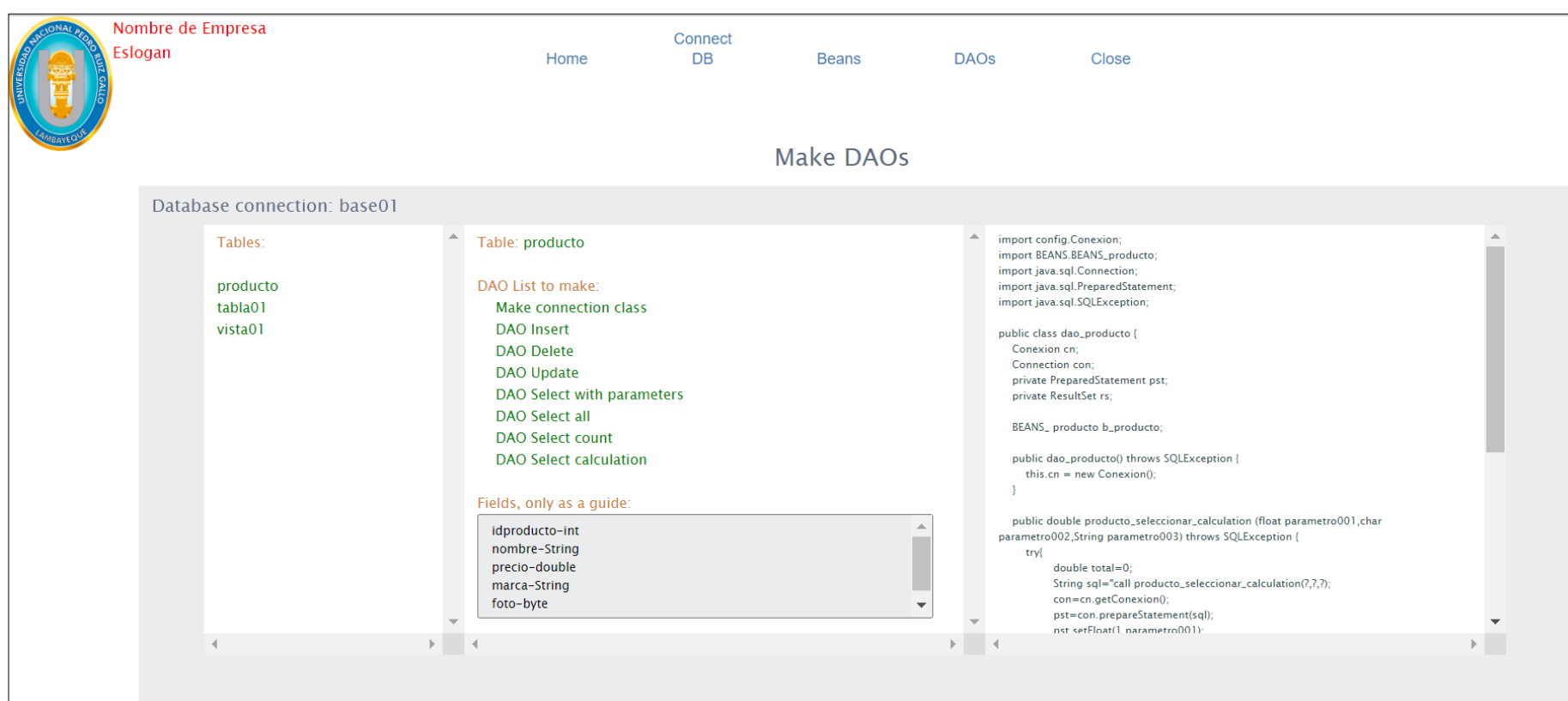


Figura 51 vista daos.jsp con scrip de Select calculation

La figura 50, presenta a la vista daos.jsp con el script del Select calculation, generara en la vista parametros_2_ccalculation.jsp cumpliendo el requerimiento REQ004.009.

4.4 Pruebas

Tabla 6 Resultado de las pruebas para verificar el cumplimiento de requerimientos funcionales

Código de requerimiento	Requerimiento	Prueba	Contenido	Resultado
REQ001	Conectar base de datos	Conexión a base de datos	1. Verificar conexión satisfactoria	1. Desarrollado
			2. Verificar conexión no satisfactoria	2. Desarrollado
REQ002	Cerrar conexión	Cerrar base de datos	1. Verificar cierre de base dados	1. Desarrollado
REQ003.001	Mostrar tablas de base de datos	Mostrar tablas de base de datos	1. Verificar que se visualicen las tablas de la base de datos seleccionada	1. Desarrollado
REQ003.002	Mostrar campos de las tablas	Mostrar campos de tabla seleccionada	1. Verificar que se visualicen los campos de la tabla seleccionada	1. Desarrollado
REQ003.003	Generar Script BEANs	Verificar Beans Generado	1. Verificar que se genere la clase con el script de beans de la tabla seleccionada	1. Desarrollado
			2. Verificar que se genere la clase con el script de beans de la vista seleccionada	2. Desarrollado
REQ004.001	Mostrar tablas	Mostrar tablas de base de datos	1. Verificar que se visualicen las tablas de la base de datos seleccionada	1. Desarrollado

REQ004.002	Construir clase conexión	Verificar Clase de Conexión	1. Verificar que se genere la clase conexión	1. Desarrollado
REQ004.003	Construir DAO insertar	Verificar DAO insertar	1. Verificar que se genere la clase con el método de insert	1. Desarrollado
REQ004.004	Construir DAO eliminar	Verificar DAO eliminar	1. Verificar que se adicionen los parámetros	1. Desarrollado
			2. Verificar que se genere la clase con el método delete	2. Desarrollado
REQ004.005	Construir DAO actualizar	Verificar DAO actualizar	1. Verificar que se adicionen los parámetros	1. Desarrollado
			2. Verificar que se genere la clase con el método update	2. Desarrollado
REQ004.006	Construir DAO seleccionar con parámetros	Verificar DAO seleccionar con parámetros	1. Verificar que se adicionen los parámetros	1. Desarrollado
			2. Verificar que se genere la clase con el método select con parámetros	2. Desarrollado
REQ004.007	Construir DAO seleccionar todos	Verificar DAO seleccionar	1. Verificar que se genere la clase con el método select	1. Desarrollado
REQ004.008	Construir DAO seleccionar contar	Verificar DAO de contar	1. Verificar que se adicionen los parámetros	1. Desarrollado

			2. Verificar que se genere la clase con el método select con para contar	2. Desarrollado
	Construir DAO seleccionar de cálculos	Verificar DAO de cálculos	1. Verificar que se adicionen los parámetros	1. Desarrollado
REQ004.009			2. Verificar que se genere la clase con el método select para cálculos	2. Desarrollado

4.4 VALIDACION DE LOS OBJETIVOS ESPECÍFICOS

Considerando:

Objetivo General

Generar los componentes Beans y DAO de un proyecto en lenguaje Java con una herramienta de generación de Script.

Objetivos específicos

Obj. Esp. 01: Desarrollar módulo de generación de Script para Beans en proyectos de lenguaje JAVA.

Obj. Esp. 02: Desarrollar módulo de generación de Script para DAO en proyectos de lenguaje JAVA.

Tabla 7 Relación entre los requerimientos y los objetivos específicos

Código de requerimiento	Requerimiento	Objetivo específico relacionado con el requerimiento
REQ001	Conectar base de datos	Obj. Esp. 01 - Obj. Esp. 02
REQ002	Cerrar conexión	Obj. Esp. 01 - Obj. Esp. 02
REQ003.001	Mostrar tablas de base de datos	Obj. Esp. 01
REQ003.002	Mostrar campos de las tablas	Obj. Esp. 01
REQ003.003	Generar Script BEANs	Obj. Esp. 01
REQ004.001	Mostrar tablas	Obj. Esp. 02
REQ004.002	Construir clase conexión	Obj. Esp. 02
REQ004.003	Construir DAO insertar	Obj. Esp. 02

REQ004.004	Construir DAO eliminar	Obj. Esp. 02
REQ004.005	Construir DAO actualizar	Obj. Esp. 02
REQ004.006	Construir DAO seleccionar con parámetros	Obj. Esp. 02
REQ004.007	Construir DAO seleccionar todos	Obj. Esp. 02
REQ004.008	Construir DAO seleccionar contar	Obj. Esp. 02
REQ004.009	Construir DAO seleccionar de cálculos	Obj. Esp. 02

Los requerimientos funcionales tienen relación directa con los objetivos específicos, el objetivo específico 01 Desarrollar módulo de generación de Script para Beans en proyectos de lenguaje JAVA, es logrado por el cumplimiento de los requerimientos funcionales REQ001 a REQ003.003, y el objetivo específico 02 Desarrollar módulo de generación de Script para DAO en proyectos de lenguaje JAVA, es logrado por el cumplimiento de los requerimientos funcionales REQ001, REQ002 y todos los requisitos del REQ004.

Tabla 8 Validación de objetivos

Variable	Requerimiento del proceso	Cantidad de requerimientos funcionales	Permite cumplir objetivo
Generar los componentes Beans y DAO de un proyecto en lenguaje Java con una herramienta de generación de Script	Desarrollar módulo de generación de Script para Beans en proyectos de lenguaje JAVA.	05	Si
	Desarrollar módulo de generación de Script para DAO en proyectos de lenguaje JAVA.	11	Si

La validación del objetivo general, generar los componentes Beans y DAO de un proyecto en lenguaje Java con una herramienta de generación de Script, se realizó validando los objetivos específicos.

DISCUSION DE LOS RESULTADOS

El resultado de la investigación es una herramienta Case, tipo lower case, que contribuye según (López Vallejos, 2011), a disminuir el tiempo de desarrollo, estandarizar y mejorar la gestión de cambio, o lo planteado por ” (HUARACA BUÑAY, 2018), automatizando el proceso de creación y generación de código, que puede ser utilizado en el proceso de enseñanza de desarrollo de sistemas de información planteado por (CHAVARRÍA-BÁEZ & OCOTITLA ROJAS, 2016).

La decisión de utilizar la metodología RUP (Jacobson, Booch, & Rumbaugh, 2000) (Jacobson, Booch, & Rumbaugh, 2000), acompañada con la técnica de Modelamiento de Procesos de Negocio (Object Management Group. 2011), dieron la posibilidad de construir un camino de actividades, se considerando varias iteraciones que permitieron presentar avances e ir mejorando con la siguiente iteración, en este ciclo de iteraciones no se fijaron la cantidad de iteraciones inicialmente, sino que conforme se avanzaba se determinaba la conclusión de la actividad con la entrega de su producto correspondiente.

Como producto desarrollado inicialmente, se decidió que el ámbito debería no privilegiar los requerimientos no funcionales, o de diseño, así como, requerimientos para el despliegue de la aplicación o requerimientos de seguridad, estos requerimientos deberán ser considerado en el momento que se decida implementar la herramienta en un servidor y ponerlo a disposición de la comunidad.

Se ha aportado al crecimiento del conocimiento contribuyendo con la herramienta de desarrollo de script para Beans y DAOs en lenguaje Java, demostrando que utilizar metodologías de desarrollo de software tipificadas como clásicas, si pueden ser utilizadas en iniciativas para aprovechar oportunidades en el mercado de tecnologías.

La herramienta apertura oportunidades de crecimiento en el mismo tipo de base de datos de desarrollo como la generación de procedimientos almacenados y vistas, y oportunidades para extender la herramienta a otros tipos de base de datos.

CONCLUSIONES

Las presentes conclusiones involucran el cumplimiento de los objetivos específicos y la justificación de la investigación:

La metodología RUP, con una secuencia de actividades y entregables seleccionado de forma adecuada, permitió desarrollar la iniciativa de desarrollar el software de herramientas case que permite generar script para Beans y DAOs en proyectos de lenguaje Java

La herramienta permite generar Script y DAOs una vez conectado a la base de datos MySQL, lo que permite disminuir actividades repetitivas o de retrabajo y de gestión de cambios en el desarrollo de sistemas de información con lenguaje Java.

RECOMENDACIONES

En el proceso de construcción de sistemas de información cada vez es necesaria la disminución del tiempo de desarrollo, herramientas que ayuden a este propósito se requerirán por lo que se debe alentar iniciativas que permitirán incursionar en este sector a futuros ingenieros de sistemas.

La herramienta que permite generar Script y DAOs conectado a la base de datos MySQL, puede extenderse a la generación de procedimientos almacenados, que permitirá no solamente la disminución del tiempo de desarrollo, sino la trazabilidad entre los componentes del sistema de información.

BIBLIOGRAFIA

CHAVARRÍA-BÁEZ, L., & OCOTITLA ROJAS, N. (2016). Sobre el uso de herramientas CASE para la enseñanza de bases de datos. México: Memorias de la Sexta Conferencia Iberoamericana de Complejidad, Informática y Cibernética (CICIC 2016).

DE MIGUEL, A., PIATTINI, M., & MARCOS, E. (2000). Diseño de Bases de Datos Relacionales. Madrid: Alfaomega.

du Plessis, A. (2005). A method for CASE tool evaluation. Information & Management, 93-102.

EcuRed:Enciclopedia_cubana. (15 de enero de 2022). Obtenido de https://www.ecured.cu/Herramienta_CASE

Hoffer, J., George, J., & Valacich, J. (2008). Modern Systems Analysis and Design. HUARACA BUÑAY, R. (2018). ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN EL DESARROLLO DE SOFTWARE. Milagro, Ecuador: Universidad Estatal de Milagro.

IBM. (setiembre de 2020). Rational Unified Process. Obtenido de https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Java. (15 de enero de 2022). Obtenido de <https://www.java.com/es/about/>

Jetbrains. (2021). El estado del ecosistema del desarrollador 2021. Praga: JetBrains.

Jetbrains. (2012 de enero de 2022). Obtenido de jetbrains.com/es-es/

López Vallejos, L. (2011). Análisis del impacto del uso de herramientas CASE en el desarrollo de software dentro de las pequeñas y medianas empresas de Querétaro. Querétaro: Universidad Autónoma de Querétaro.

Object Management Group. (2011). *Business Process Model and Notation (BPMN)*
V 2.0. Milford: OMG.



ACTA DE SUSTENTACIÓN VIRTUAL N° 020-2022-FICSA-D



Siendo las 8:30 am horas del día 16 de setiembre del 2022, se reunieron vía plataforma virtual, <https://meet.google.com/rmh-iykt-qot?pli=1>, los miembros de jurado de la tesis titulada: "HERRAMIENTA DE GENERACIÓN DE SCRIPTS PARA BEANS Y DAO EN PROYECTOS DE LENGUAJE JAVA", con código IS_V_2022_002, designado por Resolución Decanal Virtual N° 054-2022-UNPRG-FICSA con la finalidad de Evaluar y Calificar la sustentación de la tesis antes mencionada, conformado por los siguientes docentes:

DR. ING. ERNESTO KARLO CELI ARÉVALO
MG. ING. ROBERT EDGAR PUICAN GUTIÉRREZ
MG. ING. OSCAR EFRAIN CAPUÑAY UCEDA

PRESIDENTE
SECRETARIO
VOCAL

Asesorado por DR. ING. REGIS JORGE ALBERTO DIAZ PLAZA

El acto de sustentación fue autorizado por OFICIO VIRTUAL N° 095-2022-UIFICSA, la tesis fue presentada y sustentada por los Bachilleres: HERNANDEZ CABALLERO EDVIN EDUARDO y SANDOVAL DAMIAN JOSE WILLIAMS, tuvo una duración de 110 minutos. Después de la sustentación, y absueltas las preguntas y observaciones de los miembros del jurado; se procedió a la calificación respectiva:

HERNANDEZ CABALLERO EDVIN EDUARDO	16	DIECISEIS	BUENO
SANDOVAL DAMIAN JOSE WILLIAMS	17	DIECISIETE	BUENO

Por lo que quedan APTOS para obtener el Título Profesional de INGENIERO DE SISTEMAS de acuerdo con la Ley Universitaria 30220 y la normatividad vigente de la Facultad de Ingeniería Civil De Sistemas y de Arquitectura de la Universidad Nacional Pedro Ruiz Gallo.

Siendo las 10:20 am horas; se dio por concluido el presente acto académico, dándose conformidad al presente acto, con la firma de los miembros del jurado.

DR. ING. ERNESTO KARLO CELI ARÉVALO
PRESIDENTE

MG. ING. ROBERT EDGAR PUICAN GUTIÉRREZ
SECRETARIO

MG. ING. OSCAR EFRAIN CAPUÑAY UCEDA
VOCAL

DR. ING. REGIS JORGE ALBERTO DIAZ PLAZA
ASESOR



DR. ING. SERGIO BRAVO IDROGO
DECANO



“Año de la universalización de la salud”.

CONSTANCIA DE APROBACION DE ORIGINALIDAD DE TESIS

Según Res. N° 659-2020-R

Yo, Dr. Ing. Regis Jorge Alberto Díaz Plaza, **asesor de tesis de los bachilleres:**

EDVIN EDUARDO HERNANDEZ CABALLERO

JOSE WILLIAMS SANDOVAL DAMIAN

TITULADA:

HERRAMIENTA DE GENERACIÓN DE SCRIPT PARA BEANS Y DAO EN
PROYECTOS DE LENGUAJE JAVA

Luego de la revisión exhaustiva del documento constato que la misma tiene un índice de similitud de 8% verificable en el reporte de similitud del programa TURNITIN.

El suscrito analizó dicho reporte y concluyó que cada una de las coincidencias detectadas NO CONSTITUYEN PLAGIO. A mi leal saber y entender la tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad Nacional Pedro Ruiz Gallo.

Se expide la presente según lo dispuesto en la Resolución N° 659-2020-R, de fecha 8 de setiembre de 2020 formativa para la obtención de Grados y Títulos de la UNPRG:

Lambayeque, 12 de julio del 2022

ATENTAMENTE,

Dr. Ing. Regis Jorge Alberto Díaz Plaza
DNI. 16620941

Se adjunta:
Recibo digital de Turnitin
Revisión de informe en Turnitin

TESIS_HERNANDEZ_SANDOVAL

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

8%

FUENTES DE INTERNET

0%

PUBLICACIONES

2%

TRABAJOS DEL
ESTUDIANTE

FUENTE QUE CONTIENE COINCIDENCIAS

1

sedici.unlp.edu.ar

Fuente de Internet

2%

2%

★ sedici.unlp.edu.ar

Fuente de Internet

Excluir citas

Activo

Excluir bibliografía

Activo

Excluir coincidencias < 15 words



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	Edvin Eduardo Hernandez Caballero JOSE WILLIAMS SANDO...
Título del ejercicio:	PRE
Título de la entrega:	TESIS_HERNANDEZ_SANDOVAL
Nombre del archivo:	tesis_Hernandez-Sandoval.pdf
Tamaño del archivo:	2.95M
Total páginas:	95
Total de palabras:	11,788
Total de caracteres:	67,844
Fecha de entrega:	12-jul.-2022 11:20p. m. (UTC-0500)
Identificador de la entre...	1869938835

