



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de un Sistema de Productividad
del Posgrado de una Universidad**

TESIS

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A N

Alejandro Enriquez Andonaegui

José Juan Flores Saldaña

Oscar Martínez Ávila

Ingeniero Eléctrico Electrónico

P R E S E N T A

Omar Hernández Bravo

DIRECTOR DE TESIS

M. I. Juan Carlos Roa Beiza



Ciudad Universitaria, Cd. Mx., 2016

A G R A D E C I M I E N T O S

A la Universidad Nacional Autónoma de México, por formarnos como profesionistas.

A los profesores que nos compartieron su conocimiento; principalmente al M. I. Juan Carlos Roa Beiza, por brindarnos no solo la asesoría para la elaboración de este trabajo, sino también por el apoyo y la orientación en el momento más difícil del grupo.

Al Programa de Apoyo a la Titulación (PAT) del Palacio de Minería de la Facultad de Ingeniería, que nos brindó las facilidades para culminar una más de nuestras metas.

A mis compañeros, que gracias a su esfuerzo, colaboración, y experiencia aportadas, hemos tenido a bien elaborar la presente tesis.

Alejandro Enriquez Andonaegui

José Juan Flores Saldaña

Omar Hernández Bravo

Oscar Martínez Ávila

A mi madre María de los Ángeles Andonaegui Pineda por ser el mejor ejemplo de fortaleza e integridad que he tenido en mi vida, por la paciencia que me ha tenido a lo largo de los años y por siempre estar conmigo apoyándome.

A mis hermanos Margarita y Marco Antonio por ser mis compañeros de vida, que siempre han creído en mí.

A mis amigos y compañeros de tesis, lo logramos chavos!!!

A mi director de tesis el M.I. Juan Carlos Roa Beiza por el importante apoyo en la realización de este proyecto.

A la Universidad Nacional Autónoma de México, Facultad de Ingeniería y Profesores.

Alejandro Enriquez Andonaegui

A mis padres por todo su apoyo incondicional y por todos los valores inculcados que me han permitido ser mejor persona y profesional, a mis hermanos (José, Alfredo, Olga, Esmeralda, Javier, Elizabeth y Adrián) gracias por ser una de mis motivaciones para cada día ser mejor.

A mi esposa por el apoyo incondicional en esta etapa final de este proyecto.

A mis amigos y familiares que siempre han estado conmigo a lo largo de la realización de este proyecto y siempre, me han motivado para seguir adelante.

A mis amigos de IBD Consultores por todo el apoyo y motivación para desarrollar este proyecto.

A mis amigos Alejandro, Oscar y Omar gracias por compartir conmigo una de las metas más importantes que como universitarios tenemos, por el esfuerzo y las ganas que pusieron para lograr sacar este proyecto.

A la UNAM por darme la mejor formación profesional y es un orgullo pertenecer a la Máxima Casa de Estudios.

José Juan Flores Saldaña

A Flor y Tanis, mis padres, porque ustedes son el origen de lo que soy, sin importar las circunstancias y el tiempo.

A mi hermano, Ignacio Hugo, que sin importar las diferencias siempre serás mi compañero y amigo.

A mis familiares que me han motivado siempre; en especial a mi tío Salvador, por todo el ejemplo y apoyo que me ha brindado.

A mis amigos Fabián, Argelia, Alejandro, Juan, Oscar: porque sin ustedes esto no existiría.

A la Universidad Nacional Autónoma de México, por forjar mi vida: desde las aulas del Colegio de Ciencias y Humanidades Oriente; pasando por sus cines, teatros, museos, salas, canchas y jardines; hasta la formación profesional recibida en la Facultad de Ingeniería. Gracias profesores, en especial a usted, Mtro. Roa.

Omar Hernández Bravo

A mis padres,
a mi familia y amigos,
son las raíces que me alimentan.

A mi caramelo,
a mis hijos,
son el motor que mueven mis ilusiones.

A mi alma mater,
a nuestro director,
a mis profesores y condiscípulos,
son la comunidad que mueve mi intelecto.

Oscar Martínez Ávila



ÍNDICE

CAPÍTULO I – ENTORNO DEL PROBLEMA	1
1.1 Introducción	1
1.2 Descripción de los procedimientos administrativos que se van a incluir en el sistema web	7
1.2.1 Procedimiento de Estructura y personal académico del programa	7
1.2.2 Procedimiento de seguimiento de estudiantes	10
1.2.3 Procedimiento de Infraestructura de posgrados	12
1.2.4 Procedimiento de Resultados y Vinculación	14
1.2.5 Procedimiento de CVU	16
1.3 Normatividad de la institución a la que se debe apegar el proyecto	17
1.4 Alcance y acotamiento del problema	24
CAPÍTULO II – MARCO TEÓRICO	29
2.1 Características, ventajas y desventajas de la metodología CASE y bases de datos relacionales	29
2.1.1 Bases de datos relacionales	29
2.1.2 Metodología CASE	38
2.2 Características, ventajas y desventajas de la arquitectura cliente-servidor	44
2.3 Características, ventajas y desventajas de J2EE 6 y JBoss AS 7	57
2.3.1 Introducción de J2EE 6	57
2.3.2 Introducción de JBOSS AS 7	64
2.4 Características, ventajas y desventajas de Oracle Database Server	70



2.5 Características, ventajas y desventajas de SPRING MVC 3 y JSF 2.1	78
2.5.1 Introducción de SPRING MVC 3	78
2.5.2 Introducción de JSF 2.2	83
CAPÍTULO III – DESCRIPCIÓN DEL SISTEMA WEB	89
3.1 Análisis del problema	89
3.1.1 Situación actual	89
3.1.2 Solución propuesta	91
3.2 Requerimientos generales y particulares del usuario	98
3.3 Prototipos del Sistema	105
3.4 Arquitectura del sistema web	109
CAPÍTULO IV – DISEÑO Y CONSTRUCCIÓN DEL SISTEMA WEB	119
4.1 Modelo esencial ambiental del sistema web	119
4.1.1 Diagrama de contexto	120
4.1.2 Casos de uso	122
4.1.3 Diagrama entidad relación	136
4.1.4 Diccionario de datos	141
4.1.5 Diagrama de clases	154
4.2 Construcción y diseño de la base de datos	160
4.3 Construcción y diseño del sistema web	185
4.4 Aplicación de pruebas y mantenimiento	200
4.5 Evaluación del sistema web	213
CONCLUSIONES	225
REFERENCIAS BIBLIOGRÁFICAS	227



CAPÍTULO I

ENTORNO DEL PROBLEMA

1.1 Introducción.

La formación de recursos humanos de alto nivel académico representa uno de los principales objetivos de los programas académicos de la universidad buscando contribuir al desarrollo científico y tecnológico del país. La universidad con el apoyo de los programas del CONACYT, en particular el Programa Nacional de Posgrados de Calidad (PNPC) han fortalecido la calidad de un gran número de programas de la universidad, por tal motivo la universidad está buscando incrementar la cantidad de los programa académicos incorporándolos al programa PNPC.

Por medio del PNPC la universidad otorga becas para los estudiantes inscritos en los posgrados reconocidos en el PNPC, pero también por medio del programa se reciben recursos complementarios que permiten a los posgrados contar con la infraestructura física y humana así como con el equipamiento de espacios docentes y laboratorios requeridos por el PNPC. La Universidad procura el aseguramiento de la calidad de los posgrados para garantizar, primeramente, su permanencia en el PNPC, y posteriormente, su fortalecimiento para incrementar su grado de consolidación, infraestructura científica, el número de investigadores miembros del Sistema Nacional de Investigadores (SNI), productividad y alcance, finalmente, los niveles de competitividad internacional como evidencia de su pertinencia, viabilidad y capacidad académica en la formación de recursos humanos de clase mundial.



ENTORNO DEL PROBLEMA



La universidad se encuentra en un estado del sureste, que es considerado uno de los de mayor atraso científico y tecnológico, que lo colocan en el nada honroso penúltimo lugar del país en capacidad científica.

La Universidad es la institución de educación superior más importante del estado, cuya función es formar recursos humanos en diferentes niveles, que propicie el desarrollo económico, social e industrial de la entidad. La Universidad ha modificado sustancialmente sus políticas internas y se ha encaminado a lograr los niveles de calidad de los programas educativos que imparte, garantizando las competencias de sus investigaciones. Con políticas agresivas de aseguramiento de la calidad, la Universidad ha logrado acreditar en los últimos años más del 95% de sus programas educativos de Licenciatura, lo que ha propiciado que más del 90% de su matrícula reciba programas de buena calidad.

El apoyo y los incentivos otorgados por la Universidad han permitido incrementar el número de miembros en el SNI contando actualmente con 85 y estableciendo medidas y procesos que garanticen la permanencia en el programa del PNPC.

El establecimiento de procesos de apoyo a los investigadores ha permitido incrementar el éxito en la asignación de recursos destinados a la investigación, y fortalecer la infraestructura científica, incrementando en el último año los apoyos en este rubro comparado con el año anterior.



El PNPC reconoce la calidad y pertinencia de los programas de posgrado en sus dos orientaciones¹:

- Los programas de posgrado con orientación a la investigación que se ofrecen en los niveles de especialidad, maestría y doctorado en las diferentes áreas del conocimiento.
- Los programas de posgrado con orientación profesional que se ofrecen en los niveles de especialidad, maestría y doctorado, con la finalidad de estimular la vinculación con los sectores de la sociedad.

El Padrón del Programa Nacional de Posgrados de Calidad, está conformado por cuatro niveles para ambas orientaciones: la profesional y la de investigación.

- **Competencia internacional.** Programas que tienen una productividad académica relevante de estudiantes y profesores con colaboraciones en el ámbito internacional, en particular:
 - Productividad académica asociada a las líneas de generación y aplicación del conocimiento.
 - La codirección de tesis con académicos de instituciones nacionales o internacionales.
 - Movilidad de Profesores en instituciones nacionales o internacionales.
 - Coautoría de artículos con académicos extranjeros o de otras instituciones nacionales.
 - Colaboración en proyectos de investigación o desarrollo tecnológico con académicos nacionales e internacionales.
 - Participación en comités tutoriales de estudiantes en el programa y en otras instituciones nacionales e internacionales.

¹ CONACYT (2016). Sistema de consultas.
<http://svrtmp.main.conacyt.mx/ConsultasPNPC/intro.php>



- Arbitrajes solicitados por revistas prestigeadas.
- Participación como evaluadores de proyectos nacionales e internacionales.
- Participación en sociedades científicas o colegios de profesionistas.
- Estancias posdoctorales.
- **Consolidado.** Programas que tienen reconocimiento nacional por la pertinencia y la tendencia ascendente de sus resultados en la formación de recursos humanos de alto nivel, en la productividad académica y en la colaboración con instituciones nacionales y con otros sectores de la sociedad.
- **En Desarrollo.** Programas con una prospección académica positiva, sustentada en el plan de mejora continua y en las metas factibles de alcanzar en el mediano plazo.
- **Reciente creación.** Programas que satisfacen los criterios y estándares básicos del marco de referencia del PNPC y que su creación tenga una antigüedad que no exceda de 4.5 años para programas de doctorado y hasta 2.5 años para maestría y especialidad.

Los posgrados en desarrollo y de reciente creación tienen como objetivo impulsar la calidad en nuevos posgrados en áreas de interés regional y en áreas estratégicas que favorezcan el crecimiento de las capacidades científicas, tecnológicas y de innovación en las regionales o en el país.

El número de posgrados (maestría y doctorado) reconocidos en el PNPC se ha incrementado de manera muy significativa en el último año pasando de 5 a inicios de 2013 a 18 en septiembre de 2014.



CAPÍTULO I



Si bien es cierto que el número de posgrados reconocidos por el PNPC de la Universidad se ha incrementado de manera muy significativa y alentadora en el último año, es también cierto que el seguimiento y control de los procesos académicos se ha dificultado de manera igualmente significativa, alentado por la inexperiencia de algunos núcleos académicos básicos (NAB) pero también por la dispersión geográfica de las sedes administrativas de los posgrados. Los primeros resultados de ingreso, reinscripción y trámites administrativos hacia el CONACyT para el otorgamiento de las becas a los estudiantes han presentado algunas dificultades en algunos posgrados. Estas dificultades han llamado la atención de la Universidad y están estableciendo medidas correctivas y preventivas para garantizar que los procesos académicos de los posgrados sean controlados y administrados de manera tal que no represente un obstáculo en su desarrollo y no se convierta en un elemento que impida su permanencia en el PNPC.

El aseguramiento de la calidad de los posgrados actualmente reconocidos por el PNPC y los que se encuentran en proceso de evaluación, requiere de procesos eficientes que permitan conocer de manera oportuna y preventiva los procesos académicos tales como ingreso, trayectoria escolar, eficiencia terminal, seguimiento de egresados y producción científica que faculte a la institución a tomar medidas puntuales y a tiempo para garantizar su permanencia en el PNPC. Como consecuencia de esto la Universidad requiere establecer un Sistema de Productividad de Posgrados a través de una plataforma automatizada que alerte y de evidencia de los procesos académicos del posgrado de la universidad, y se ha convertido en una necesidad imperiosa para garantizar el seguimiento de sus procesos académicos.

De tal manera que nuestra tesis estará agrupada en 4 capítulos.



En el primer capítulo abarcaremos la situación actual de los programas académicos, explicaremos los conceptos básicos de los mismos.

Dentro del segundo capítulo explicaremos el por qué de la utilización de JEE 6, metodología Case, Spring MVC 3, JSF 2.2, JBoss AS 7 y Oracle Database Server 11g, mostrando las características, ventajas y desventajas de cada una de estas metodologías, frameworks y lenguajes.

En el capítulo tres de la presente tesis hacemos referencia al análisis del problema, con el cual una vez entendido, plantearemos la posible solución a través de una aplicación.

En el último capítulo se muestra como se llevó a cabo la implementación de esta aplicación entrando a los detalles como la base de datos, la interfaz al usuario y el resultado final reflejado en los reportes que servirán para retroalimentar a los coordinadores de los posgrados.

Por último daremos nuestras conclusiones sobre las diferencias y logros obtenidos de llevar el seguimiento del proceso administrativo de la calidad de los programas académicos de manera manual a llevarlo a cabo de una forma automática apoyándonos con una aplicación específica para esto.



1.2 Descripción de los procedimientos administrativos que se van a incluir en el sistema web.

1.2.1 Procedimiento de estructura y personal académico del programa.



Figura 1.2.1 Procedimiento administrativo de la estructura y personal académico del programa.

En la figura 1.2.1 se muestra el procedimiento administrativo de la estructura y personal académico del programa, se enfoca en definir la organización del programa del posgrado y que posea la siguiente: Los profesores tienen el grado requerido, poseen formación docente y en el uso de las tecnologías de la información y las comunicaciones (TICs) para favorecer el aprendizaje. Tienen



ENTORNO DEL PROBLEMA



experiencia y una alta productividad en investigación o innovación, generando por lo menos un producto anual ya sean publicaciones, patentes o desarrollos tecnológicos.

Los profesores se encuentran involucrados en actividades de vinculación con la sociedad y ejercen un liderazgo esencial en el proceso de innovación académica y social.

Los estudiantes del programa son seleccionados de entre los mejores egresados, mediante procesos objetivos, explícitos, transparentes y éticos. El programa atrae e incluye a los mejores estudiantes con base en sus méritos, sin importar su origen social, y mantiene una demanda de ingreso creciente y mayor a la admisión, por lo que existe una selección demostrada de estudiantes.

El plan de estudios responde a la orientación y misión de la universidad, al desarrollo del campo de conocimiento y considera y prepara a los estudiantes en la evolución en el área. Enuncia con claridad las competencias a desarrollar, tanto en el ámbito académico, como en el profesional.

La organización curricular posee una orientación clara y está planeada expresamente para desarrollar una visión panorámica del campo, de sus métodos de investigación y desarrollo, y de su historia conceptual, favoreciendo la autonomía intelectual, la creatividad, la capacidad de realizar investigación original e innovación y de transferencia de conocimiento al contexto de la práctica. Formando, en su caso, para aplicar o desarrollar el conocimiento de frontera.



CAPÍTULO I



El mapa curricular del plan de estudios está integrado por asignaturas obligatorias y optativas, que favorece la multidisciplinaria. El programa establece de manera equilibrada actividades docentes, de investigación, innovación y de vinculación con la sociedad, cuenta con la infraestructura y el financiamiento indispensable para el desarrollo de sus actividades, y también reconoce y recompensa las actividades de sus académicos en la docencia, la investigación e innovación y la vinculación con el mundo exterior.

Este procedimiento permite controlar y administrar todos los posgrados existentes manejados por la dirección de posgrados, llevando el seguimiento del estado del posgrado, responsables o coordinadores, fecha de creación, vigencia en el programa PNPC, nivel de PNPC. Para su aprobación se necesita evaluar la historia y los antecedentes del posgrado, llevando el control de la fecha de aprobación por consejo universitario, fecha de registro en la unidad de profesiones, adicionalmente los posgrados para su aprobación del plan de estudios y planta académica del núcleo académico básico, profesores de tiempo parcial y colaboradores de cada posgrado, evaluando que el plan de estudio y la planta académica cumpla con los objetivos por los cuales se creó el posgrado.



1.2.2 Procedimiento de seguimiento de estudiantes.



Figura 1.2.2 Procedimiento de seguimiento de estudiantes.

En la figura 1.2.2 se muestra el procedimiento de seguimiento a estudiantes que se encarga de definir los puntos necesarios para darle el seguimiento a todo el ciclo de los estudiantes desde su ingreso, permanencia y egresos de cada alumno.

Este procedimiento permite una selección de estudiantes rigurosa y objetiva. La calidad y el potencial de los estudiantes que ingresen reeditarán en altas tasas de graduación y en la calidad general del programa educativo.



La institución en el diseño de los procedimientos de ingreso toma en cuenta varios criterios de selección para evaluar el potencial de un estudiante de posgrado. Entre estos criterios pueden ser:

- Examen de admisión que permita evaluar conocimientos y habilidades de acuerdo con el perfil de ingreso;
- Entrevista con el estudiante;
- En el caso de programas de doctorado después de maestría, también se podría incorporar como requisito la elaboración de un anteproyecto de investigación;
- Carta de intención;
- Antecedentes académicos;
- Curso propedéutico;
- Promedio del ciclo anterior de 8; y
 - Matrícula de ingreso: Este parámetro sirve para determinar las Instituciones de Educación Superior (IES) de diversos tamaños, en el entendido de que la problemática para cada una es diferente.
 - Origen de los estudiantes: Este parámetro permitirá determinar si el programa tiene una demanda local, nacional o internacional. En este sentido se debe valorar el porcentaje de estudiantes inscritos que provienen de IES diferentes a la que se inscriben. De igual manera se debe considerar el porcentaje de estudiantes extranjeros.
 - Que los criterios de ingreso tengan valores preestablecidos y explícitos de acuerdo a la naturaleza del posgrado; estos pueden consistir en un puntaje ponderado, analizados y aplicados de manera objetiva por el cuerpo colegiado correspondiente.

Este procedimiento administrativo controla desde el registro de los aspirantes aplicando los criterios de evaluación. Se encargará de definir el ingreso de los estudiantes, la asignación de tutores y el trámite de las becas de los alumnos,



para posteriormente darle el seguimiento oportuno de la permanencia, además revisa la productividad y eficiencia terminal de los estudiantes, así como la evaluación de los egresados para evaluar sí el posgrado está cumpliendo con los objetivos para el cual fue creado.

1.2.3 Procedimiento de Infraestructura de posgrados.



Figura 1.2.3 Procedimiento de Infraestructura de posgrados.

El procedimiento de infraestructura del posgrado se muestra en la figura 1.2.3 y permite el desarrollo del programa de posgrado, se constituye por un conjunto de espacios organizados con orden y relaciones definidas, el cual permite realizar la acción del posgrado y su proceso formativo.

Dichos espacios, están constituidos a su vez por tres ámbitos: físico, virtual y académico.



El espacio físico lo conforman: las aulas, los cubículos para profesores y estudiantes, el equipamiento, bibliotecas, laboratorios, y recursos materiales.

El espacio virtual se ha vuelto imprescindible, pues constituye una extensión de las capacidades del programa y permite, almacenar, procesar, compartir información, además de favorecer los procesos de aprendizaje, y auspiciar la comunicación y la formación de redes de colaboración, incluso con personas que se encuentran a distancia.

El espacio académico está integrado por profesores, estudiantes, personal de apoyo, asimismo incluye a los campos de práctica, profesores y tutores externos. Involucra la organización y el plan de estudios, a través de los cuales se despliegan las tareas de formación, investigación, vinculación con la sociedad y transferencia del conocimiento permitiendo desplegar, el ambiente intelectual, de trabajo en equipo y de colaboración.

En el procedimiento administrativo de infraestructura de posgrado distribuye la infraestructura existente para la dirección de posgrados, a tener el control de la cantidad de cubículos, aulas, laboratorios, equipo de cómputo para profesores y alumnos, bibliotecas y bibliotecas virtuales, este procedimiento se encargará de medir sí la infraestructura es suficiente para impartir con calidad el programa académico.



1.2.4 Procedimiento de Resultados y Vinculación.

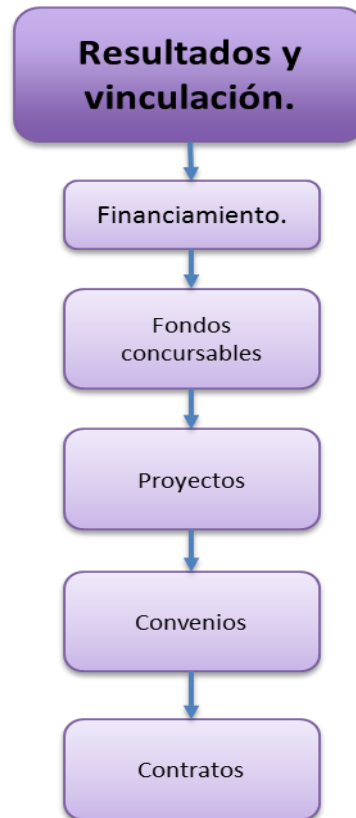


Figura 1.2.4 Procesos administrativos de resultados y vinculación.

El proceso de resultados y vinculación se muestra en la figura 1.2.4 y está constituido por los efectos del proceso formativo, incluyendo el aprendizaje y el dominio de las competencias. Los resultados de investigación o innovación, tienen relación con publicaciones, desarrollos tecnológicos, transferencia del conocimiento y contribución al desarrollo regional.



CAPÍTULO I



La incorporación al empleo tanto en sectores académicos como profesionales, el liderazgo ejercido por los egresados y su contribución al desarrollo social, la participación en sociedades académicas y profesionales, la participación en redes de conocimiento y en la formación de recursos humanos calificados en entornos no académicos, también hacen parte de esta categoría.

Conviene destacar que la calidad de los resultados, se encuentra determinada por los procesos bien desarrollados, porque el éxito se logra bajo diseño. Los resultados deben ser entendidos como indicadores de buenos procesos. Los resultados del posgrado se desarrollan en el mediano y largo plazo, por lo que un seguimiento sistemático de egresados, se convierte en pieza clave.

El programa de posgrado desarrolla las competencias de sus egresados y estos logran una visión panorámica del campo y sus métodos, siendo capaces de discutir y reflexionar sobre temas altamente relevantes. Los egresados y sus profesores demuestran satisfacción con los logros del programa. Se obtienen una pluralidad de productos de la investigación y desarrollo, como publicaciones académicas, informes técnicos, patentes, prototipos y procesos, de carácter innovador, que demuestran originalidad y pensamiento creativo que reestructura la visión de los problemas en el campo.

Los egresados se gradúan en tiempo y con evaluaciones de calidad, e ingresan a sociedades académicas y profesionales de carácter selectivo, recibiendo con frecuencia reconocimientos a sus aportaciones. Se incorporan con facilidad al empleo por su capacidad y formación. El posgrado como organización ejerce un liderazgo intelectual y es reconocido como una escuela de pensamiento a nivel nacional e internacional, se ha constituido en un elemento central para el



desarrollo regional o nacional y contribuye de manera sustantiva a la innovación de las actividades sociales y económicas. Ha construido redes de colaboración con sus egresados que demuestran capacidad de liderazgo y trabajo en equipo, por lo que se constituyen en un factor de cambio y desarrollo.

El procedimiento administrativo de resultado y vinculación del posgrado permite medir los resultados esperados, medir los montos de financiamiento, controlar el apoyo proporcionado por cada organismo nacional e internacional.

1.2.5 Procedimiento de Currículum Vitae Universitario (CVU).

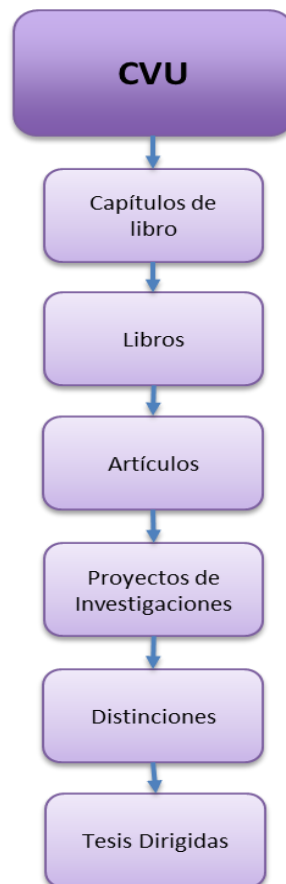


Figura 1.2.5.1 Procesos administrativos de CVU.



En la figura 1.2.5.1 se muestra el procedimiento de CVU, se enfoca en definir la productividad académica de los alumnos y profesores: Este procedimiento tiene como finalidad controlar y administrar la productividad en investigación o innovación, tesis dirigidas, artículos, publicaciones, patentes o desarrollos tecnológicos y libros publicados.

1.3 Normatividad de la institución a la que se debe apegar el proyecto.

La finalidad de este proyecto es implementar un Sistema de Productividad del Posgrado de una Universidad, empleando herramientas de última generación, que permita a la institución el aseguramiento de la calidad de los posgrados actualmente reconocidos por el Programa Nacional de Posgrado de Calidad y los que se encuentran en proceso de evaluación, con base a las normas del programa PNPC se apegará a las siguientes normas:



Tabla (1.1) Principales rasgos de los posgrados de Investigación²

Criterio	Rasgo	Referente
Admisión de estudiantes	El programa de posgrado cuenta con un proceso de admisión riguroso que asegure que los aspirantes tienen los conocimientos previos necesarios.	Los procedimientos y criterios de admisión se ajustan a sus objetivos y son públicos, accesibles y garantizan la formación adecuada de ingreso. En el programa existen estudiantes de otras instituciones del ámbito nacional e internacional.
Perfil de Egreso	El programa tiene como propósito la formación de investigadores críticos y creativos a través de investigaciones originales.	Las metas y los objetivos del programa son coherentes con el perfil de egreso y las líneas de generación y aplicación del conocimiento que se abordan en el programa y son de actualidad y comparables internacionalmente.

² CONACYT (2016). Marco de referencia. <http://www.conacyt.mx/index.php/el-conacyt/convocatorias-y-resultados-conacyt/convocatorias-pnpc/convocatorias-cerradas-pnpc/9005-marco-de-referencia-modalidad-escolarizada/file>



CAPÍTULO I



Plan de estudios	El programa de posgrado cuenta con un plan de estudios bien estructurado, acorde con el perfil de egreso.	Síntesis del plan de estudios, en la cual se definen detalles tales como la organización de los contenidos en sus dimensiones de verticalidad y horizontalidad, el número de cursos por periodo lectivo (año escolar, semestre, cuatrimestre, trimestre, etcétera), los cursos obligatorios y optativos, el número de cursos teóricos y prácticos con relación a las necesidades formativas del programa de posgrado, los créditos o asignaturas, la seriación y los prerrequisitos.
Estudiantes de dedicación exclusiva	El programa cuenta con estudiantes con una dedicación exclusiva para culminar los estudios, incluida la tesis doctoral, en el tiempo establecido en el plan de estudios.	Los estudiantes tienen dedicación completa al programa y a las actividades afines a él, y cuentan con financiamiento (becas, contratos, etcétera) para concluir los estudios en el tiempo previsto en el plan de estudios.



ENTORNO DEL PROBLEMA



Infraestructura	El programa garantiza la disponibilidad de la infraestructura física y los recursos necesarios para el desarrollo de la investigación a realizar por el estudiante, con estándares internacionales.	El programa cuenta con una infraestructura científica y recursos adecuados para su desarrollo, de acuerdo con el diseño de formación previsto. Además tiene una estrategia para la mejora de la infraestructura científica y los recursos.
Núcleo académico básico	El programa de posgrado se desarrolla en el seno del núcleo académico básico.	El núcleo académico básico responsable del programa tiene una productividad académica reconocida y un interés científico común, sobre la base de lo cual comparten líneas de generación o aplicación del conocimiento, claramente planteadas y están activos científicamente y tienen una producción de calidad y suficiente en número, considerando los objetivos del doctorado.
Dirección de tesis	Los directores de tesis orientan a los estudiantes, cooperan con ellos y siguen de cerca su actividad académica para lograr resultados de investigación.	El núcleo académico básico realiza una atención personalizada y un seguimiento periódico eficaz del trabajo de los estudiantes. Los estudiantes participan en las reuniones, los seminarios o actividades similares organizadas en el núcleo académico.



CAPÍTULO I



Movilidad de estudiantes y profesores	El programa de posgrado promueve la movilidad de estudiantes y profesores así como la cooperación entre grupos de investigación afines de diferentes instituciones y países.	El Programa muestra evidencia de que sus estudiantes y profesores han participado en acciones de movilidad en los últimos 5 años. Existen informes o memorias de proyectos realizados, tesis codirigidas (realizadas o en proceso de elaboración), asesorías o publicaciones, en el marco de los convenios o acuerdos de cooperación.
Tesis de doctorado	La tesis de doctorado es un trabajo original de investigación.	Las tesis doctorales dan lugar a una o más publicaciones en revistas científicas internacionales indexadas, libros, etcétera. La preparación y defensa de la tesis doctoral debe reunir tres condiciones: <ul style="list-style-type: none">• Una solución original a un problema de investigación.• Que el doctorando reúna un conocimiento suficiente del área del programa de posgrado.• Que el doctorando sea capaz de desarrollar investigación independiente en su área científica.



ENTORNO DEL PROBLEMA



Tesis de Maestría y Especialidad	En general las tesis de maestría y especialidad son el inicio metodológico de la investigación.	Las tesis de maestría y especialidad deberán mostrar evidencias que los estudiantes son aptos para continuar sus estudios de doctorado
Contribución al conocimiento	Resultados y productos del programa de posgrado.	Medida en que los resultados de la investigación o el desarrollo tecnológico o la innovación contribuyen a la generación y aplicación del conocimiento y atiende los problemas y oportunidades del programa, o bien, se transforman en aplicaciones y se transfieren a actores pertinentes para su implantación o explotación.
Pertinencia del Programa	Desempeño de los egresados en una actividad laboral afín a su formación.	Para el caso de los doctorados un indicador es el contar con el reconocimiento académico de los egresados en el Sistema Nacional de Investigadores.
Satisfacción de los egresados	Valor aportado por el programa a sus egresados y reconocido por éstos (encuestas de satisfacción de los egresados).	Contribución de los egresados al conocimiento y reconocimiento por los empleadores y la sociedad.



La estructura de los posgrados estará definida también con base a la ley universitaria de la universidad:

ARTÍCULO 84. En la Universidad, los Planes de Estudio se estructurarán, administrarán y desarrollarán bajo el modelo denominado currículum flexible. Consecuentemente, se considera a un Plan de Estudios oficial y flexible, cuando reúne las siguientes características:

- Es previamente evaluado y aceptado por el o los Consejo(s) de Unidad(es) responsable(s) y por el o los respectivos Consejos Académicos Colegiales involucrados;
- Una vez evaluado y aceptado es aprobado por el H. Consejo Universitario;
- Es administrado y ejecutado por una o más Unidades Académicas, pertenecientes a una o más Redes o Colegios de la Universidad;
- Según el nivel o modalidad educativa, el plan ofertará un conjunto relacionado de cursos o materias o dejará el espacio curricular para que el educando, en función de sus intereses, perfil profesional buscado, necesidades de especialización o requerimientos de formación integral, bajo determinados criterios y respetando los procedimientos previamente establecidos.



1.4 Alcance y acotamiento del problema.

El Sistema de Productividad del Posgrado manejará toda la información de forma segura en una plataforma web, accediendo de manera expedita y preventiva a los procesos académicos tales como ingreso, trayectoria escolar, eficiencia terminal, seguimiento de egresados y producción científica que permitan a la institución tomar medidas puntuales y a tiempo para garantizar su permanencia en el PNPB.

A través de una plataforma automatizada que alerte y dé evidencia de los procesos académicos para garantizar el seguimiento de los posgrados.

Se implementará un sistema Web empleando herramientas JEE 6, Spring MVC 3, JSF 2.2 y JBoss AS 7 que son software de distribución libre; para el almacenamiento de la información se utilizará Oracle Database 11g, del cual la Universidad posee el licenciamiento. El sistema web deberá contar con los siguientes módulos:

- **Estructura del posgrado.**

- o Consulta, alta, edición de los posgrados, donde el director del posgrado podrá dar de alta los nuevos posgrados o los posgrados actuales, el Director de posgrados podrá subir información de la clave del posgrado, coordinador del posgrado, Unidad académica Sede, fecha de creación del posgrado, estado del posgrado (creación, restructuración o vigente), nivel en el PNPB, fecha de obtención en el PNPB, grado Académico (Posgrado, Maestría o Especialidad) y fecha de registro ante profesiones.



- **Ingreso.**

- Selección de los criterios de evaluación por posgrado, el coordinador seleccionará los criterios de evaluación al posgrado que se les aplicará a los aspirantes.
- Consulta alta y edición para la asignación de tutores a los nuevos estudiantes, el coordinador podrá conocer cuándo es el ingreso de un alumno al posgrado, asignarle un tutor, y el tutor podrá registrar las reuniones de tutorías para dar seguimiento a los alumnos.

- **Permanencia – Egreso.**

- Consulta de seguimiento a materias, reuniones de tutores y movilidad de los alumnos, donde los alumnos y tutores podrán consultar y registrar información de las reuniones de tutoría.
- Consulta, alta y edición de seguimiento al protocolo, la dirección (seguimiento), y revisión de la tesis. Para este proceso, el coordinador del posgrado será responsable de validar el protocolo y asignar los directores y revisores de la tesis; una vez autorizado el protocolo el sistema deberá permitir cargar los avances de la tesis y los comentarios por el director y el revisor así como el porcentaje de avance en la elaboración de la tesis.

- **Infraestructura.**

- Consulta, alta y edición de la cantidad de aulas, cubículos, laboratorios equipo de cómputo, bibliotecas físicas y virtuales que cuenta cada posgrado, el coordinador del posgrado será responsable de capturar la cantidad de cada rubro y una evidencia en formato PDF de todo el detalle del inventario.

- **Resultados y vinculación.**

- Consulta, alta y edición de los proyectos desarrollados, contratos y convenios de los posgrados, el coordinador y director de posgrado podrán consultar los proyectos desarrollados por investigadores y estudiantes de la Universidad.



- **Investigación (CVU)**

- o Consulta, alta y edición de libros publicados, artículos, tesis, docencia, distinciones, investigación y desarrollo tecnológico, por alumno o profesor; cada profesor y alumno podrá cargar su información y el coordinador o director de posgrado deberá poder consultarla.

- **Seguridad y administración**

- o Manejo de privilegios para los usuarios y roles del sistema. Este deberá de permitir administrar los mismos, los usuarios se podrán auto-registrar siempre y cuando cuenten con un correo institucional. El usuario se definirá de la siguiente manera: para los profesores deberá ser su clave de empleado y para los alumnos su matrícula.

El sistema Web permitirá el acceso de manera exclusiva al personal que labora en la Universidad y a los estudiantes de la misma, por lo que es necesario crear una base de datos en la que se encuentre el registro de toda la información de los módulos previamente mencionados, todo esto será almacenado en Oracle 11g, que es un producto con el que la institución ya cuenta.

El diseño de este sistema web y la base de datos permitirán a la Universidad contar con un sistema de difusión educativa más amplia de manera online, disponible en todo momento. La forma en que se relaciona este subsistema con el sistema integral de la Universidad, se muestra en la figura 1.4.1.



Fig. 1.4.1 Sistema Integral de la Universidad.



ENTORNO DEL PROBLEMA





CAPÍTULO II

MARCO TEÓRICO

2.1 Características, ventajas y desventajas de la metodología CASE y bases de datos relacionales.

2.1.1 Bases de datos relacionales.

Una **base de datos** está formada por un conjunto de datos que se encuentran en dispositivos de almacenamiento que permiten el acceso directo a los datos, y un conjunto de programas, conocidos como **DBMS (DataBase Management System, Sistema Gestor de Base de Datos, SGBD)** efectúan la administración del conjunto de datos.

La finalidad de una base de datos es servir a una o varias aplicaciones de la mejor manera posible. Un sistema de base de datos se puede definir como:

- Conjunto de datos almacenados físicamente en la base de datos.
- Hardware, los dispositivos de almacenamiento físico.
- Software, programas para manejar la base de datos, o DBMS.
- **Database Administrator (DBA, Administrador de Base de Datos, ABD)**, quien se encarga del control y administración de la base de datos.
- Usuarios finales, quienes manipulan e interpretan la información.



También es importante mencionar que los datos almacenados deben cumplir lo siguiente:

- Debe implementarse un modelo que describa la estructura e interrelaciones de los datos, con base a las reglas de operación de la Universidad.
- Debe controlarse la redundancia de datos, aunque por cuestiones de eficiencia se presente redundancia.
- Deben servir a toda la organización.
- La definición y descripción del conjunto de datos contenidos en la base debe ser homogénea e integrada con los mismos datos.
- Se debe realizar la actualización y recuperación de los datos mediante los procesos del SGBD, con el fin de mantener la integridad, seguridad y confidencialidad de la base.

Las **bases de datos relacionales** tienen los siguientes objetivos:

- Independencia física, que implica que la representación lógica de los datos no debe verse afectada por la forma en que se almacenan estos.
- Independencia lógica, que se consigue al no modificar las aplicaciones que aprovechan la base de datos cuando se modifican elementos de la misma base de datos. Las modificaciones en el nivel lógico son necesarias siempre que la estructura lógica de la base de datos se altere.
- Flexibilidad, al ofrecer distintas vistas de la base de datos en función de los usuarios y aplicaciones.
- Sencillez.



Características de las bases de datos relacionales.

- Uso de tablas.
- Filas (tuplas).
- Atributos (campos).
- Presentación simplificada, respecto a las bases de datos de red.
- Independencia física.
- Independencia lógica.
- Flexibilidad.
- Uniformidad.
- Sencillez.

Ventajas de las bases de datos relacionales.

- Reduce la duplicación en el ingreso de datos.
- Búsquedas más rápidas.
- Creación de informes y formularios selectivos a través de consultas.
- Favorece la normalización.
- Permite la integridad referencial.
- Independencia de los datos, con respecto a los tratamientos que reciben.
- Mejor disponibilidad de los datos.
- Mayor eficiencia en la recopilación.
- Mayor coherencia.
- Mayor valor informativo.
- Acceso más rápido y sencillo de los usuarios finales.
- Más facilidades para compartir los datos por el conjunto de los usuarios.
- Mayor flexibilidad para atender a demandas cambiantes.



Desventajas de las bases de datos relacionales.

- Deficientes con datos gráficos.
- Instalación compleja.
- Exceso de tablas.
- Difícil comprensión de interrelación de los elementos.
- En algunas aplicaciones aumenta el tráfico en la red.
- Si está centralizada corre el riesgo de alguna caída abrupta.

Este modelo se basa en el uso de tablas, que se representan gráficamente como una estructura rectangular formada por filas y columnas, o tuplas y atributos (ver Fig. 2.1.1.1). Cada columna almacena información sobre una propiedad determinada de la tabla; mientras que cada fila posee una ocurrencia de la instancia o relación representada por la tabla.

NOMBRE

atributo 1	atributo 2	atributo 3	atributo n	
valor 1,1	valor 1,2	valor 1,3	valor 1,n	← tupla 1
valor 2,1	valor 2,2	valor 2,3	valor 2,n	← tupla 2
.....
valor m,1	valor m,2	valor m,3	valor m,n	← tupla m

Fig. 2.1.1.1 Representación de una tabla en el modelo relacional

Estas tablas pueden ser persistentes, que solo pueden ser borradas por los usuarios, y temporales (en memoria), que se eliminan automáticamente por el sistema. Ambos tipos de tablas se subdividen en base, vistas, o instantáneas.



Las tablas base son independientes, y se crean indicando su estructura y sus ejemplares. Las tablas vistas solamente almacenan una definición de consulta; si los datos de las tablas base cambian, los de la vista que utiliza esos datos también cambian. Las tablas instantáneas son vistas que almacenan los datos que muestra, además de la consulta que dio lugar a esa vista. El resultado de las tablas instantáneas se actualiza por el sistema cada cierto tiempo.

El uso de dominios en este modelo, permite especificar los posibles valores válidos para un atributo. Cada dominio incorpora su nombre y una definición del mismo.

Las claves son atributos que identifican a las tuplas de la tabla. Se le llama clave candidata a aquella que identifica unívocamente cada tupla de la tabla. Una clave primaria es una clave candidata que se escoge como identificador de las tuplas. Las claves alternativas son cualquier clave candidata que no sea primaria. Y al atributo de una tabla relacionado con una clave de otra tabla se le llama clave externa o secundaria.

Los valores nulos indican contenidos de atributos que no tienen ningún valor. En las bases de datos relacionales se admite utilizar valores nulos en todo tipo de operaciones, además de los valores falso o verdadero. Las restricciones son condiciones de cumplimiento obligado por los datos de la base. Son restricciones inherentes aquellas que no son determinadas por los usuarios, sino por el hecho de que la base de datos sea relacional. Las restricciones semánticas son las que incorpora el usuario a los datos, de forma personal.



Cardinalidad.

Es el número de ocurrencias de una entidad en una relación. Para un conjunto entidades A y B, la cardinalidad puede ser:

- Una a una. Una y sola una entidad de A esta asociada a una entidad de B y viceversa.
- Una a muchas. Una entidad de A esta asociada a una o varias entidades en B, en cambio una entidad en B solo puede estar asociada con una sola entidad de A.
- Muchas a muchas. Una entidad en A esta asociada a cualquier cantidad de entidades en B; y una entidad en B está asociada a cualquier cantidad de entidades en A.

Una vez creado el modelo de datos, se cuenta con una estructura no normalizada motivo por el cual se pueden presentar varias anomalías o desventajas, se recomienda que antes de comenzar con la manipulación de los datos se realice una normalización que consiste en un proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas; que además son más simples y más estables; son más fáciles de mantener. Este proceso sigue una serie de reglas que sirven de apoyo a los diseñadores de bases de datos en el desarrollo de un modelo relacional que minimice los problemas de lógica debido a que mejora la manera de definir las diferentes tablas y claves que se involucran en el diseño y funcionalidad de la base de datos. Las reglas a seguir se conocen como **formas normales**, y se enlistan a continuación^{1, 2}:

¹ Aguilar G., Concha E., Corona N., Macario Alejandro & Reyes J (2012). Sistema de punto de venta para una empresa fotográfica.

² Gómez, M. (2013). Notas del curso bases de datos. Universidad Autónoma Metropolitana.



- **Primera forma normal.** Una relación R se encuentra en primera forma normal si y solo si por cada renglón columna tiene valores atómicos, la tabla contiene una llave primaria y si no posee ciclos repetitivos.

En otras palabras, una tabla se encuentra en primera forma normal si y sólo si los valores que componen el atributo de una tupla son atómicos. Los atributos atómicos, son valores únicos y lo más pequeño posible, es decir, indivisibles. Además tiene una clave primaria, atributos no nulos y no tiene tuplas repetidas.

- **Segunda forma normal.** Una tabla esta en segunda forma normal si y solo si esta en primera forma normal y los atributos no primos dependen funcionalmente de la llave primaria; es decir no existen dependencias parciales.

Dicho de otra forma, una tabla está en la segunda forma normal cuando está en la primera forma normal y además cada campo secundario (aquel que no pertenece a la clave principal) depende de la clave principal en su totalidad y no de una parte de ella. Esto se cumple cuando ninguna clave primaria es compuesta, es decir, esta forma normal sólo se considera si la clave principal de la tabla es compuesta.

- **Tercera forma normal.** Una relación R esta en tercera forma normal si y solo si esta en segunda forma normal y todos sus atributos no llave, dependen transitivamente de su llave principal.

O lo que es lo mismo, una tabla está en la tercera forma normal cuando está en la segunda forma normal y además cada campo que no sea llave primaria solo depende de la llave primaria o de las claves secundarias de la tabla y no depende de otro campo de tal forma que no existen atributos no primarios que son transitivamente dependientes de cada posible clave de la tabla.



En resumen, para que una tabla cumpla con la tercera forma normal, un atributo secundario sólo se debe conocer a través de la clave principal o claves secundarias de la tabla y no por medio de otro atributo no primario.

- **Forma normal de Boyce-Codd³.** Un esquema de relación R está en forma normal de Boyce-Codd si, para todas las dependencias funcionales de la forma $A \rightarrow B$, donde A y B son subconjuntos de R, se cumplen las siguientes condiciones:
 - $A \rightarrow B$ es una dependencia funcional (DF) trivial (B es un subconjunto de A). Entonces se mantiene la condición de forma normal Boyce-Codd, debido a que sólo una DF no trivial puede violar esta condición.
 - A es una superclave del esquema R. Si se tiene $A \rightarrow B$, pero no $B \rightarrow A$, entonces A es la superclave y cada dependencia no trivial contiene a A en la izquierda, por lo tanto no hay violación a la condición de la forma normal de Boyce-Codd.

Toda dependencia funcional trivial tiene una clave candidata como su determinante.

- **Cuarta forma normal.** Una tabla esta en cuarta forma normal si esta en tercera forma normal y en la forma normal Boyce-Codd y no contiene dependencias multivaluadas.
- **Quinta forma normal.** Una tabla esta en quinta forma normal si no se puede dividir y volver a reconstruir sin pérdida de información y también está en cuarta forma normal⁴.

³ Date, C. (2001). Introducción a los sistemas de bases de datos. Pearson Educación.

⁴ Centeno, E. (2008). Formas normales: 1FN, 2FN, 3FN, 4FN, 5FN.
http://istprpecenteno.blogspot.mx/2008_06_01_archive.html



Para finalizar este punto, en la tabla 2.1.1.1 se muestra una comparativa entre el modelo de base de datos relacionales y el modelo de base de datos orientado a objetos.

	Ventajas	Desventajas
Modelos Relacionales	<ul style="list-style-type: none">- Ofrecen sistemas simples y eficaces para representar y manipular los datos.- Provee herramientas que garantizan evitar la duplicidad de registros.- Garantiza la integridad referencial.- Al eliminar un registro elimina todos los registros relacionados dependientes (independencia lógica).- Favorece la normalización por ser más comprensible y aplicable.- Integridad de datos.- Uniformidad en cuanto a la manipulación de la BD por parte de los usuarios.- Sencillez, para que sea fácil de comprender y utilizar por parte del usuario final.	<ul style="list-style-type: none">- Presentan deficiencias con datos gráficos, multimedia, CAD, y sistemas de información geográfica.- No se manipulan de forma manejable los bloques de texto como tipo de dato.- Las bases de datos orientadas a objetos (BDOO), se propusieron con el objetivo de satisfacer las necesidades de las aplicaciones anteriores y así complementar, pero no sustituir, a las bases de datos relacionales.- Pueden promover problemas de islas de información. Como el modelo relacional es tan fácil de utilizar, a muchas personas se les hace fácil crear sus propios subconjuntos y aplicaciones de BD.



<p>Modelos Orientados a Objetos</p>	<ul style="list-style-type: none"> - Manipula datos complejos de una manera rápida. - Mayor capacidad de modelado: ya que un objeto puede encapsular un estado como un comportamiento, puede almacenar todas las relaciones que tenga, y pueden agruparse para formar objetos complejos. - Ampliabilidad: dado que se pueden construir datos a partir de los ya existentes, y se pueden reutilizar las clases. - Lenguaje de consulta más expresivo. - Mayores prestaciones, puesto que proporciona mejoras significativas de rendimiento. - Adecuación a las aplicaciones avanzadas de BD, ya que estos modelos resultan efectivos para sistemas multimedia, CAD, CASE, OIS. 	<ul style="list-style-type: none"> - La depuración de código orientado a objetos es más compleja que la depuración de código estructurado. - En caso de producirse un error, se debe recorrer todo el árbol de herencia para encontrarlo. - Imposibilidad de usar procedimientos almacenados, ya que los objetos solo pueden ser consultados en el cliente. - Inmadurez en el mercado. Se carece de experiencia por ser modelos relativamente nuevos. - Carecen de una base teórica, puesto que no tienen un modelo de datos que esté universalmente aceptado. - La optimización de consultas compromete el encapsulamiento. Esto se debe a que se requiere una comprensión de la implementación de los objetos, para poder acceder a la base de datos de manera eficiente.
--	---	---

Tabla 2.1.1.1 Comparativa de modelos de bases de datos⁵.

2.1.2 Metodología CASE.

Debido a que el desarrollo de sistemas de información es una actividad compleja, ésta puede dividirse para su estudio en las siguientes etapas⁶:

⁵ Lorza, I. (2013). Cuadro comparativo de los modelos de base de datos. <http://es.slideshare.net/alorza1/ventajas-y-desventajas-de-los-modelos-de-bd>

⁶ Sommerville, I. (1992). Software Engineering. AddisonWesley.



- Definición y análisis de los requerimientos del usuario.
- Diseño del sistema y de la base de datos.
- Implementación y prueba de módulos.
- Integración y pruebas del sistema.
- Operación y mantenimiento.

Estas etapas son muy elaboradas, por lo que se han desarrollado distintos métodos para realizarlas de forma estructurada. La metodología **CASE** (**Computer Aided System Engineering**, Ingeniería de Software Asistida por Computadora, ISAC) plantea una secuencia de etapas que es similar a la citada anteriormente, pero más detallada, proporcionando para cada etapa su descripción, definición de objetivos y metas, productos de la etapa, factores críticos de éxito, y la lista de tareas que conviene realizar.

Características.

- Topdown
- Secuencial
- Detallada
- Automatizada
- Descriptiva

Ventajas.

- Mejora de la calidad y aumento de la productividad de los desarrollos
- Facilidad para la revisión de aplicaciones
- Soporte para el desarrollo de prototipos de sistemas
- Generación de código ordenado
- Satisfacción de los requerimientos del usuario
- Soporte para el proceso de desarrollo



Desventajas.

- Incompatibilidad de los procesos del cliente en el uso los métodos estructurados
- Carencia de estandarización para el soporte de la metodología
- Conflictos en el uso de los diagramas
- Diagramas no utilizados
- Función y alcance limitados

La metodología CASE está basada en un análisis y desarrollo del tipo descendente (**topdown**), donde el ciclo de vida de un sistema está compuesto por las siguientes etapas (ver Fig. 2.1.2.1):

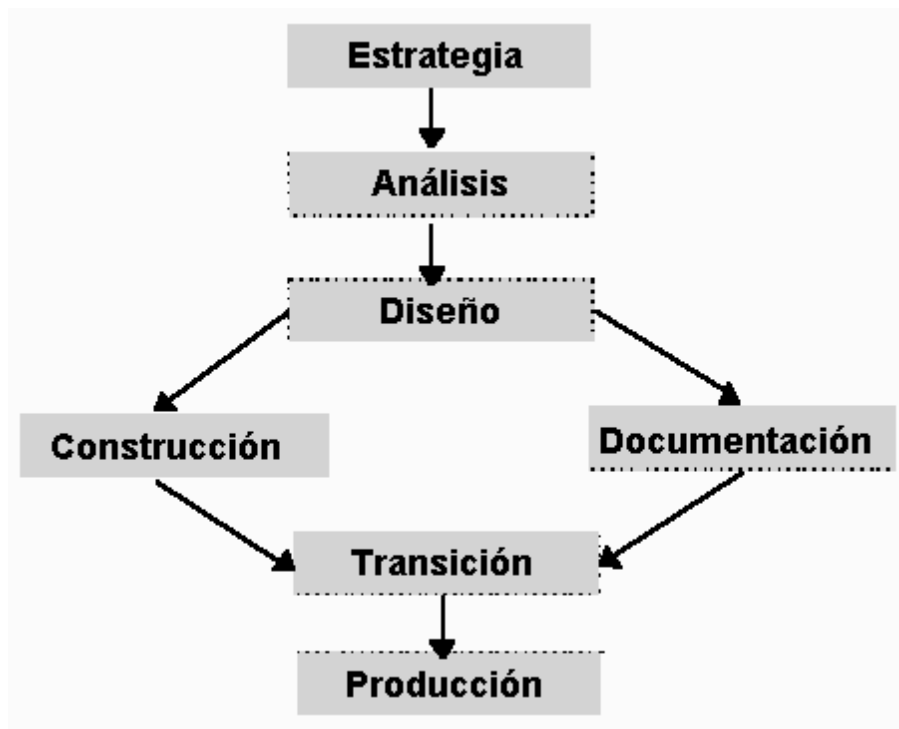


Fig. 2.1.2.1 Etapas de la metodología CASE



La **estrategia** tiene la finalidad de lograr un entendimiento claro de las necesidades de la organización y del ambiente en que operará el sistema a implantar. Es aquí cuando se analizan las funciones que lleva a cabo la organización y sus necesidades de información a todos los niveles. Se realizan entrevistas con la dirección y los responsables de los departamentos, y partiendo de esta información se realiza un primer modelado de los requerimientos del sistema de información acorde a las necesidades de la organización. Después, y para definir una primera versión de la arquitectura del sistema, también se consideran las tecnologías disponibles en ese momento y los sistemas de información operantes. Como resultado de esta etapa, se obtienen modelos de la empresa, recomendaciones, y un plan acordado de desarrollo de los sistemas de información.

Durante el **análisis**, y con el fin de refinar los modelos de la etapa anterior, se realiza otra serie de entrevistas a un nivel operativo y técnico, realizando un análisis detallado de sus requerimientos específicos. A partir de esto, se genera el modelado del sistema. Los resultados de esta etapa son el modelo de entidad-relación (explotación total de los datos); el modelo funcional (modelado de los servicios); y se definen las restricciones que tendrá el sistema y la estrategia a seguir en la etapa de transición.

En el **diseño** se pasa del diseño conceptual al diseño final que servirá para la implantación. En esta etapa el modelo de entidad-relación se transforma en un diseño lógico, empleando un modelo relacional; mientras que el modelo funcional se transforma en módulos y manuales de procedimientos. Los resultados de esta etapa son la arquitectura del sistema, el diseño de la base de datos, la especificación de los programas, y la especificación de los manuales de procedimientos.



En la etapa de **construcción** se codifican y prueban los programas nuevos. Los resultados son los programas probados y la base de datos afinada. La **documentación** comprende la elaboración de los manuales, a partir de las especificaciones de diseño, de los programas hechos, y del análisis de las condiciones de los usuarios y operadores de los sistemas.

Durante la **transición**, se alimentan las nuevas bases de datos, se capacitan a los usuarios, y se realizan pruebas. Todo esto encaminado a la implementación, y con una interrupción mínima de la organización. El resultado es un reporte de pruebas satisfactorias.

En la **producción** se asegura que el sistema funcione correctamente en la totalidad de los casos, y con intervención mínima de los administradores del sistema. Para conseguirlo, se llevan a cabo nuevas pruebas, se reevalúan los resultados y se refina el sistema, introduciendo los cambios sin afectar a los usuarios, consiguiendo la máxima confianza de éstos. El resultado, un sistema listo para su operación.

Diagramas utilizados en la metodología CASE.

- Diagramas de **contexto**.

Es el nivel cero de un diagrama de flujo de datos, en donde una sola sección o burbuja puede representar todo el sistema, muestra a través del flujo de datos las interacciones existentes entre agentes externos y el sistema, se representa como un único proceso de entradas y salidas.



En este tipo de diagramas no se representan todos los flujos de datos del sistema, se representa una visión general, siguiendo:

- Una representación de los flujos de datos que tienen que ver con el objetivo principal del sistema.
- El uso de flujos de datos compuestos que representen características similares.
- Diagramas de **casos de uso**.

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

- Diagramas **entidad relación**.

El modelo de datos entidad relación está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos. La entidad representa una "cosa" u "objeto" del mundo real con existencia independiente, es decir, se diferencia únicamente de otro objeto o cosa, incluso siendo del mismo tipo,



o una misma entidad. A las características que definen o identifican a una entidad se les llama atributos de esa entidad.

- Diagrama de **clases**.

Las clases se describen por medio del diagrama de clases. La notación para una clase es una caja rectangular, que contiene el nombre de la clase. La notación general para el objeto se extiende mediante el nombre de la clase subrayado seguido del nombre del objeto. Por lo general, se utilizan más los diagramas de clases que los diagramas de objetos, ya que los diagramas de clases son más generales y corresponden a varios diagramas de objetos.⁷

2.2 Características, ventajas y desventajas de la arquitectura cliente-servidor.

De acuerdo al Software Engineering Institute (SEI), la **arquitectura de software** se refiere a "las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos."⁸ Los elementos pueden ser entidades que existen en tiempo de ejecución (objetos, hilos), entidades lógicas que existen en tiempo de desarrollo (clases, componentes) y entidades físicas (nodos, directorios). Por otro lado, las relaciones entre elementos dependen de propiedades visibles (o públicas) de los elementos, quedando ocultos los detalles de implementación. Finalmente, cada conjunto de elementos relacionados de un tipo particular corresponde a una estructura distinta, de ahí que la arquitectura está compuesta por distintas estructuras.

⁷ Weitzenfeld, A. (2005). Ingeniería de Software Orientada a Objetos con UML, Java e Internet.

⁸ Bass L., Clements P., Kazman R. (2003). Software Architecture in Practice.



La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de éste para satisfacer lo que se conoce como los atributos de calidad del sistema. Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema, o bien la modificabilidad, que tiene que ver con qué tan simple resulta introducir cambios en el sistema.

La **arquitectura cliente-servidor** es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. El cliente realiza una petición al servidor y este le responde⁹.

Características:

- La arquitectura distingue dos elementos básicos: un sistema cliente y un sistema servidor.
- El sistema cliente genera solicitudes que son enviadas al servidor y espera la respuesta de este.
- El sistema servidor recibe las peticiones del cliente y es capaz de procesarlas para entregar una respuesta.
- Centralización de la gestión de la información.
- La capacidad de proceso está repartida entre los clientes y los servidores.
- Separación de responsabilidades.
- Cada plataforma puede ser escalable independientemente.
- Un servidor da servicio a múltiples clientes en forma concurrente.

⁹ Ferrer, J. (2014). Implantación de aplicaciones web en entornos de internet, intranet y extranet.



Ventajas:

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado.
- Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Fácil mantenimiento.
- Facilita la integración entre sistemas diferentes y el intercambio de información a diferentes niveles.
- Los clientes pueden ser sistemas de recursos limitados.

Desventajas:

- La congestión del tráfico (a mayor número de clientes, más problemas para el servidor).
- El software y el hardware de un servidor son generalmente muy determinantes.
- Normalmente se necesita software y hardware específico, para satisfacer el trabajo.



Dentro de la arquitectura cliente-servidor, se han ideado modelos de programación basados en capas o planos. La programación por capas tiene el objetivo primordial de lograr una separación de la lógica de negocios de la lógica de diseño.

Una de las clasificaciones mejor conocidas de las arquitecturas cliente-servidor se basa en la idea de planos (tier), la cual es una variación sobre la división o clasificación por tamaño de componentes. Esto se debe a que se trata de definir el modo en que las prestaciones funcionales de la aplicación serán asignadas, y en qué proporción, tanto al cliente como al servidor. Dichas prestaciones se deben agrupar entre los tres componentes clásicos para cliente-servidor: interfaz de usuario, lógica de negocios y los datos compartidos, cada uno de los cuales corresponde a un plano.

Dentro de esta categoría tenemos las aplicaciones en dos planos (dos capas), tres planos (tres capas) y multi-planos (n-capas). Este término ha sido utilizado indistintamente para referirse tanto a la parte lógica (software) como a la parte física (hardware) de una aplicación.

La parte a la que se refiere la programación en n-capas se da lugar en la parte lógica de la aplicación, este segmento es el más generalizado y el que más se ajusta a los enfoques modernos, dado que se fundamenta en los componentes lógicos de la estructura Cliente/Servidor y en la madurez y popularidad de la computación distribuida. Por ejemplo, esto permite hablar de servidores de aplicación distribuidos a lo largo de una red, y no tiene mucho sentido identificar a un equipo de hardware como servidor, sino más bien entenderlo como una plataforma física sobre la cual pueden operar uno o más servidores de aplicaciones.



Los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas cliente-servidor:

- El proceso cliente es quien inicia el diálogo. También es conocido como Front-End.
- El proceso servidor es el que pasivamente espera a que lleguen peticiones de servicio y se le conoce con el término de Back-End.
- El Middleware corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes.

Modelo cliente-servidor de dos capas.

Tradicionalmente la arquitectura cliente-servidor está basada en un modelo de computación de dos capas. Este modelo consiste de un cliente y un servidor de base de datos (ver Fig. 2.2.1). El procesamiento de tareas y la lógica de la aplicación son compartidas entre el servidor de base de datos y el cliente.

En este modelo a los clientes se les llama clientes pesados, en donde reside mucho del poder de procesamiento y de la lógica de la aplicación. Esto hace que el mantenimiento del cliente sea costoso.

Adicionalmente, los clientes pueden estar operando en diferentes plataformas, dando como resultado una distribución compleja de versiones de aplicaciones específicas de las plataformas.

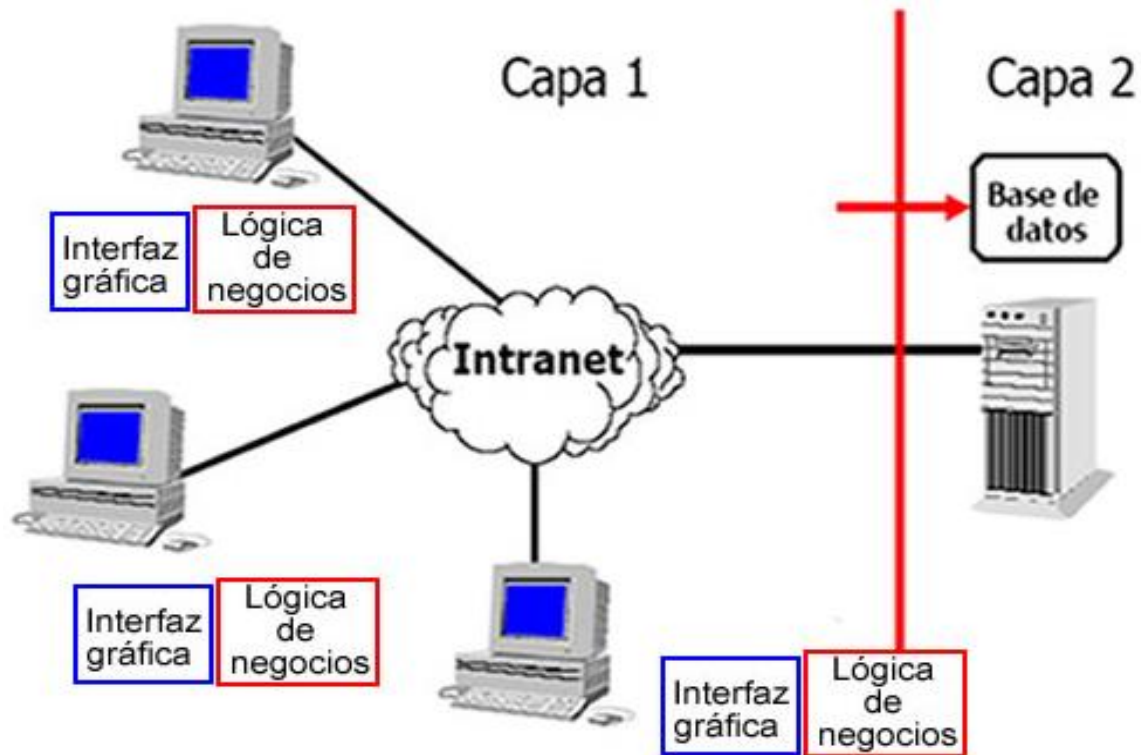


Fig. 2.2.1 Diagrama de modelo cliente-servidor de dos capas.

Las desventajas del modelo de dos capas son:

- El nivel de las aplicaciones se recargan, entremezclando aspectos típicos del manejo de la interfaz con las reglas del negocio.
- Las reglas del negocio quedan dispersas entre el nivel de aplicación y los procedimientos de la base de datos. El nivel de aplicación puede ser demasiado pesado para el cliente.

También puede existir en este modelo un cliente delgado y un servidor robusto; esto es lo que generalmente se utiliza en la práctica.



Modelo cliente-servidor de tres capas.

El modelo de computación de tres capas es un modelo evolucionado del modelo de dos capas. En este modelo, existe una capa intermedia entre el cliente y el servidor de base de datos. Esta capa consiste de un servidor de aplicaciones que contiene el grueso de la lógica de la aplicación (ver Fig. 2.2.2). Los clientes en este modelo son clientes livianos o clientes ligeros. Con esta arquitectura la lógica de la aplicación reside en una sola capa que puede ser fácilmente mantenida. El diseño arquitectónico de la capa media puede también ser optimizada en funciones del servidor puesto que éste no tiene que contener u hospedar la base de datos.

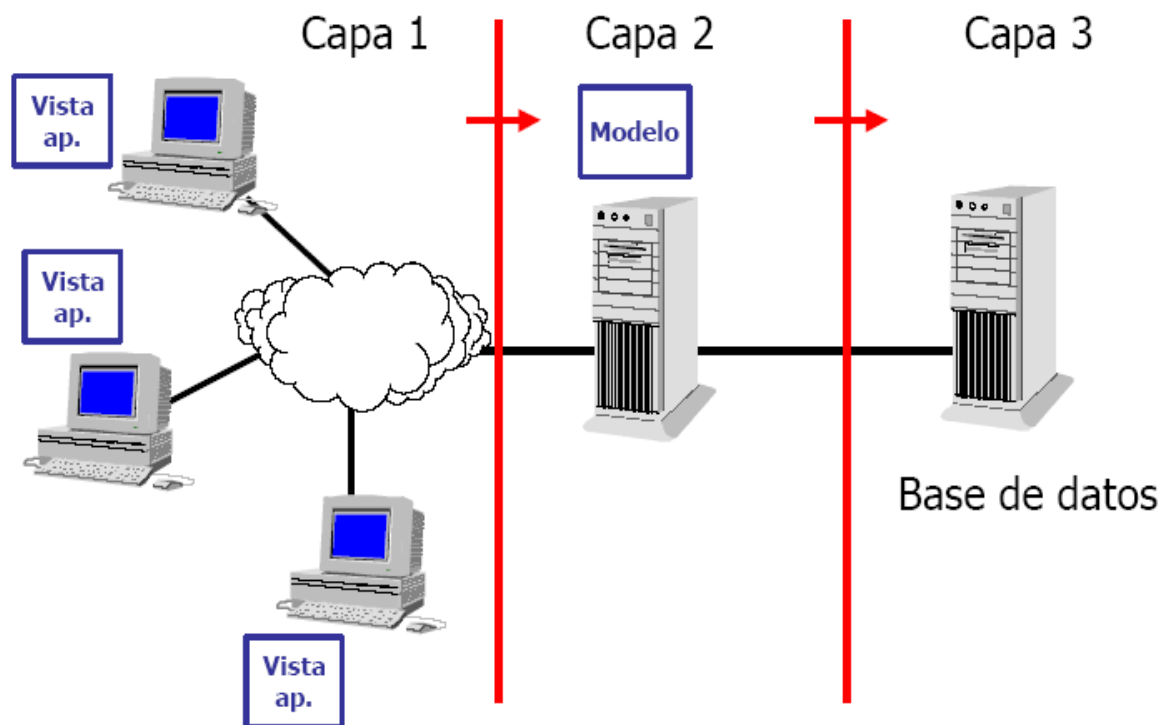


Fig. 2.2.2 Diagrama de modelo cliente-servidor de tres capas.



En esta arquitectura de tres capas, el software del cliente (capa cliente) es ligeramente pesado, suficiente para ser descargado bajo demanda y lo suficientemente pequeño como para presentar la interfaz del usuario. El grueso de la lógica de la aplicación está implementada ya sea en la capa media (servidor de aplicaciones) o está almacenada en la base de datos. Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central de bases de datos. El servidor local tiene un comportamiento dual, dado que actúa como cliente o servidor en función de la dirección de la comunicación. Algo muy importante y que es la mayor ventaja de esta arquitectura es que ahora únicamente se cambia la regla en el servidor de aplicación y ésta actuará en todos los clientes, cosa que no sucedía con la arquitectura en dos capas ya que si alguna regla se cambia, es necesario ir a cada cliente a realizar el cambio. La mayor desventaja que tiene esta arquitectura es que no se puede realizar un repositorio de datos distribuido, un ejemplo de este tipo de repositorio es el de los bancos que necesitan de un repositorio de datos distribuido para poder tener sus sucursales en otros países y ciudades.

Modelo cliente-servidor de n-capas.

El modelo n-tier (n-capas) de informática distribuida ha emergido como la arquitectura predominante para la construcción de aplicaciones multiplataforma en la mayor parte de las empresas. Como tecnología, las arquitecturas de n-capas proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios constantes.

Todas las aplicaciones basadas en n-capas permitirán trabajar con clientes ligeros, tal como navegadores de Internet, WebTV, Teléfonos Inteligentes, PDAs



(Personal Digital Assistants o Asistentes Personales Digitales) y muchos otros dispositivos preparados para conectarse a Internet.

De este modo, las arquitecturas de n-capas se están posicionando rápidamente como la piedra angular de los desarrollos de aplicaciones empresariales y las compañías están adoptando esta estrategia a una velocidad de vértigo como mecanismo de posicionamiento en la economía emergente que tiene su base en la red (lo que se ha venido a denominar "Nueva Economía").

Las arquitecturas basadas en n-capas permiten a los componentes de negocio correr en una LAN, WAN o Internet. Esto significa que cualquiera con un ordenador y conexión a la Red (por ejemplo, un miembro del equipo de fuerza de ventas) posee toda la funcionalidad que tendría si se encontrase delante de su sistema de escritorio.

Realmente, los componentes distribuidos de una arquitectura de n-capas es una tecnología esencial para crear la siguiente generación de aplicaciones e-business, aplicaciones que son altamente escalables, fiables y que proporcionan un alto rendimiento y una integración sin fisuras con los sistemas de back-end heredados. Las arquitecturas empresariales de n-capas se están convirtiendo en la nueva base para el desarrollo de aplicaciones de misión crítica y ofrecen la única arquitectura funcional para la siguiente generación de soluciones informáticas distribuidas basadas en Internet. Los sistemas distribuidos de n-capas proporcionan un conjunto de avances tecnológicos sin precedentes, como pooling de conexiones, multiplexado de conexiones, balanceo de carga dinámico y rendimientos excelentes en hardware trabajando en cluster. Las aplicaciones industriales basadas en n-capas pueden ayudar a las compañías a desarrollar un



nuevo núcleo de habilidades en prácticamente todo, desde la gestión del conocimiento hasta los sistemas relacionados con comercio electrónico.

A diferencia de lo que se pudiera pensar, el desarrollo en n-capas no es un producto o un estándar, es un concepto estratégico que ayuda a la construcción y despliegue lógico de un sistema distribuido. Los sistemas de n-capas subdivididos ayudan a facilitar el desarrollo rápido de aplicaciones y su posterior despliegue, con beneficios incrementales fruto de los esfuerzos del desarrollo en paralelo coordinado y del outsourcing inteligente, resultando un enorme decremento del tiempo de desarrollo y de sus costes.

Una arquitectura basada en clientes ligeros desplaza la capa de presentación de la aplicación en el lado del cliente, mientras que la lógica de negocio y los datos residen en el middleware y los servidores de back-end. El diseño para clientes ligeros minimiza los problemas de despliegue de las aplicaciones, mientras que maximiza la accesibilidad a la misma desde una amplia variedad de plataformas heterogéneas. Los frameworks basados en n-capas se crean para obtener las ventajas de los estándares abiertos de la industria que permiten a las aplicaciones resultantes operar en entornos distribuidos multiplataforma.

Utilizando estos potentes estándares abiertos se permite a los integradores de sistemas asegurar la efectividad y consistencia de la comunicación y mensajería a todos los niveles en aplicaciones cruzadas Interdepartamentales e Interempresariales.



El diseño de aplicaciones basado en n-capas considera a la red como un pool de servicios distribuidos, un concepto mucho más ambicioso que el simple acceso de un cliente a un servidor. La separación de la presentación, lógica de negocio y datos es realizada en un número indefinido de capas lógicas, permitiendo a cada capa ser desarrollada, mejorada, gestionada y desplegada de forma independiente. Esta es precisamente la base para el modelo de informática de red en n-capas. Las plataformas multicapa funcionan consistentemente a lo largo de un variado conjunto de hardware, permitiendo escalar las operaciones del negocio desde un simple portátil, hasta un datacenter, desde el dispositivo más simple hasta el más complejo de los mainframes.

Las arquitecturas basadas en n-capas son el siguiente paso lógico en un proceso de evolución, el cual, está basado en las arquitecturas convencionales cliente-servidor (2 y 3 capas) más la convergencia de dos tecnologías tremendamente potentes:

- Desarrollo de aplicaciones basadas en componentes - relacionado directamente con la Programación Orientada a Objetos (Lenguajes y Técnicas).
- Internet - primer ejemplo de un sistema complejo de n-capas cliente-servidor.

Los sistemas de n-capas utilizan técnicas de desarrollo basadas en componentes combinados con los estándares abiertos de Internet, para crear aplicaciones multiplataforma muy potentes con bajos costes, fáciles de mantener y con gran efectividad. Lo que realmente es nuevo en el modelo de n-capas es la posibilidad de distribuir objetos independientes sobre el número de capas que sean



necesarias y enlazarlas dinámicamente, cuando sea necesario, para proporcionar una flexibilidad ilimitada a la aplicación.

N-Tier forma parte también de un revolucionario proceso, actualmente en desarrollo, basado en la aplicación de estas nuevas tecnologías (componentes y estándares de Internet). Estas tecnologías son los bloques para crear software de negocio y sistemas de información adaptables que ayuden a las empresas a integrar todos sus sistemas de tecnologías de la información, así como las inversiones realizadas en éstos, mientras que obtienen una ventaja clara en el uso de Internet.

El surgimiento de la tecnología de componentes distribuidos es la clave de las arquitecturas de n-capas. Estos sistemas de computación utilizan un número variable de componentes individuales que se comunican entre ellos utilizando estándares predefinidos y frameworks de comunicación como:

- **CORBA** (Common Object Request Broker Architecture) del Object Management Group (OMG).
- **DNA** (Distributed interNet Architecture) de Microsoft (incluye COM/DCOM y COM+ además de MTS, MSMQ, etc.).
- **EJB** (Enterprise Java Beans) de Sun Microsystems.
- **XML** (eXtensible Markup Language) del World Wide Web Consortium (W3C).



Estas y otras tecnologías en rápida evolución proporcionan la infraestructura necesaria y la fontanería relacionada que permite a las compañías operar en un entorno complejo, multiplataforma y con capacidades de computación distribuida, tanto interna como externamente según se requiera en cada caso.

El desarrollo de aplicaciones en n-capas es un proceso iterativo de división del problema en piezas manejables denominadas componentes. Estos componentes, o "Componentes de Negocio - Business Objects" son "modelos software" basados típicamente en la "vista" de un objeto real, evento o proceso de negocio. Los componentes software individuales pueden formar parte y adaptarse tanto de estructuras independientes como de sistemas colaborativos.

El diseño de aplicaciones en n-capas es ideal para la creación de sistemas adaptables, donde cada componente puede ser utilizado y reutilizado en nuevas combinaciones para satisfacer requisitos de negocio dinámicos. Esto permite a los desarrolladores y a las nuevas aplicaciones reutilizar componentes existentes que modelan lógica de negocio sobradamente probada. En un entorno tremendamente cambiante como el actual, utilizar aplicaciones basadas en diseños de n-capas posibilita a las empresas ser más ágiles y adaptables en proporcionar valor a sus clientes. Los sistemas basados en n-capas tienen el potencial de reducir drásticamente tanto el time-to-market para las nuevas aplicaciones de negocio, como el coste total de mantenimiento, adaptando estos complejos y caros sistemas a las siempre cambiantes necesidades empresariales.

Los frameworks de n-capas ayudan a proveer una solución integral enlazando los procesos de negocio críticos a la vez que se conservan las inversiones existentes de la empresa en tecnologías de la información y recursos relacionados.



2.3 Características, ventajas y desventajas de JEE 6 y JBoss AS 7.

2.3.1 Introducción de JEE 6.

Las aplicaciones empresariales cada vez más tienen la necesidad de desarrollar aplicaciones distribuidas, transaccionales y portátiles que aprovechan la velocidad, la seguridad y la fiabilidad de la tecnología del lado del servidor. Las aplicaciones empresariales proporcionan la lógica de negocio para una empresa, así como la implementación de requerimientos no funcionales. Se gestionan de forma centralizada y, a menudo interactúan con otro software de la empresa. En el mundo de la tecnología de la información, las aplicaciones empresariales deben estar diseñados, construidos y producidos por menos costo, con mayor velocidad y con menos recursos.

La Plataforma Java Enterprise Edition (JEE) es una plataforma de código abierto basada en estándares abiertos, el desarrollo de aplicaciones empresariales Java nunca ha sido más fácil o más rápido. El objetivo de la plataforma Java EE es proporcionar a los desarrolladores un potente conjunto de APIs (Application Programming Interface), que les permita reducir la complejidad de la aplicación y el tiempo de desarrollo desde el punto de vista no funcional.



La plataforma Java EE utiliza un modelo de programación simplificado. Archivos descriptores de despliegue XML son opcionales. En su lugar, un desarrollador puede simplemente introducir la información como una anotación directamente en un archivo de código fuente de Java, y el servidor Java EE configurará el componente en la implementación y ejecución. Estas anotaciones se utilizan generalmente para ser asociados en un programa de datos que de otro modo se suministra en un descriptor de despliegue. Con anotaciones, se pone la información de las especificaciones en el código junto al elemento del programa afectado.

La plataforma Java EE se desarrolla a través de la comunidad Java Community Process (JCP), que es responsable de todas las tecnologías Java. Grupos de expertos, integrado por las partes interesadas, han creado un gran número de especificaciones Java Specification Requests (JSR) para definir las diversas tecnologías Java EE. El trabajo de la Comunidad Java bajo el programa JCP ayuda a garantizar la compatibilidad de la norma de estabilidad y multi-plataforma de tecnología Java.

Java EE promueve la creación de aplicaciones por medio de capas, las cuales son grupos lógicos y físicos que cumplen un propósito específico y permiten especializar las diferentes tareas que una aplicación debe soportar, esta separación es gracias al manejo de inyección de dependencia que permite aplicar todos los recursos a las necesidades de los componentes, ocultando efectivamente la creación y la búsqueda de recursos de código de la aplicación. La inyección de dependencias se puede utilizar en contenedores EJB, contenedores Web y clientes de la aplicación. La inyección de dependencias permite que el contenedor Java EE asigne automáticamente referencias a otros componentes o recursos requeridos, utilizando anotaciones.



Dichas arquitecturas son implementadas por medio de los servicios que proporcionan los servidores de aplicaciones, a continuación en la figura 2.3.1.1 se muestra dichas arquitecturas.

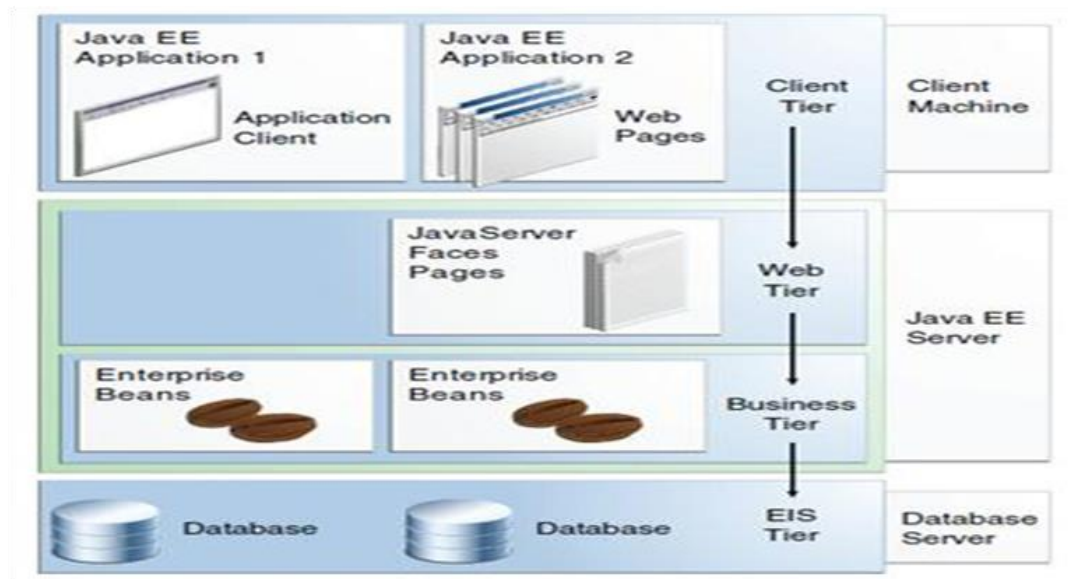


Figura 2.3.1.1 Modelo arquitectónico multicapa Java EE

La plataforma JAVA se forma de APIs estandarizadas como un subconjunto de tecnologías orientadas a distintos tipos de aplicaciones. La plataforma JAVA tiene 3 tipos de ediciones:

- Java SE (Standard Edition) aplicaciones de propósito general.
- Java EE (Enterprise Edition) para desarrollo de aplicaciones empresariales.
- Java ME (Micro Edition) para la programación de dispositivos móviles.



Características de Java EE 6.

Algunas características que pueden utilizar las aplicaciones empresariales por medio de Java EE son las siguientes^{10 11}:

- Inyección de dependencias: Esta característica permite la capacidad de inyectar componentes en una aplicación de un modo seguro, incluyendo la capacidad de elegir en tiempo de despliegue, la capacidad de inyectar desde una interface sobre una particular implementación, esta característica permite contar con componentes altamente des-acoplados y nos permitirá tener mejor modularidad en los desarrollos.
- Multiplataforma: Se puede ejecutar y desplegar en cualquier plataforma y sistema operativo, Java utiliza su propia máquina virtual para su ejecución esto es lo que permite ser multiplataforma.
- Aplicación distribuida: Una de las características de JEE es que utiliza un modelo de aplicación de varios niveles distribuida para las aplicaciones empresariales. La lógica de la aplicación se divide en componentes de acuerdo con la función, y los componentes de las aplicaciones que componen una aplicación Java EE están instalados en varias máquinas, dependiendo del nivel en el entorno Java EE de varios niveles a la que pertenece el componente de aplicación.

¹⁰ JAVA (2013). The Java EE 6 Tutorial. <http://docs.oracle.com/javaee/6/tutorial/doc/bnaaw.html>

¹¹ JAVA (2008). Community Development of Java Technology. <http://jcp.org/en/jsr/detail?id=316>



- Seguridad: Aunque otros modelos de aplicaciones empresariales requieren medidas de seguridad específicas de la plataforma en cada aplicación, el entorno de seguridad de Java EE permite las restricciones de seguridad que se definirán durante el despliegue. La plataforma Java EE hace que las aplicaciones sean más portátiles para una amplia variedad de implementaciones de seguridad protegiendo a los desarrolladores de aplicaciones con la complejidad de la implementación de elementos de seguridad. La plataforma Java EE proporciona reglas de control de acceso estándar declarativas que están definidos por el desarrollador e interpretadas cuando la aplicación se implementa en el servidor. Java EE también proporciona mecanismos de acceso estándar para que los desarrolladores de aplicaciones no tengan que poner en práctica estos mecanismos en sus aplicaciones. La misma aplicación funciona en una variedad de entornos de seguridad sin cambiar el código fuente.
- Persistencia: Java contiene una API que permite administrar la persistencia de objetos, proporciona a los desarrolladores un mecanismo de mapeo objeto / relacional para la gestión de datos relacionales en aplicaciones Java permitiendo mejorar los tiempos de desarrollo y rendimiento de aplicaciones, persistiendo ciertos objetos en memoria para su posterior sincronización con la base de datos, para que puedan ser consultados de forma más rápida así como el acceso a los datos de una forma sencilla.
- Modularidad: Se pueden desarrollar componentes altamente desacoplados que permite mejorar la administración y la independencia en los despliegues de cada uno.
- Base de datos: Ofrece un modelo estándar con un conjunto de API's y interfaces para que los fabricantes de diferentes proveedores de base de datos desarrollen sus propios conectores, con el fin de que los proveedores de base de datos cumplan con la necesidad de sus cliente en conectar sus aplicaciones JEE con los diferentes manejadores de base de datos.



- Encapsulamiento: Esta característica protege a los datos asociados a un objeto para que no puedan modificar por quien no tenga derecho a acceder a ellos. Permite manejar los objetos como unidades básicas, dejando oculta su estructura interna.
- Multitarea: J2EE permite la creación de procesos que se ejecuten de forma simultánea independientemente del sistema operativo.
- Datos compartidos.-J2EE permite compartir la información de los objetos por medio de la localidad de memoria o bien comunicarse mediante mensajes.
- Plataforma de código fuente.

Ventajas de Java EE 6.

Algunas ventajas que se pretende implementar en las aplicaciones empresariales por medio de Java EE son las siguientes:

- Disponibilidad y portabilidad: Debe permitir múltiples plataformas y sistemas operativos.
- Fiabilidad: Cumple con las expectativas de negocio y tecnológicas.
- Seguridad: Debe controlar el acceso, la autorización y el transporte.
- Robustez: Es tolerante a fallos.
- Escalabilidad: Tiene que facilitar la ampliación y modificación necesarias.
- Alta disponibilidad: Es fácil su adquisición, formación y actualización.
- Fácil mantenimiento: Es fácil mantener el sistema mediante la actualización de sus componentes.
- Reusabilidad: Nos soporta reutilizar el código o componentes ya implementados.



- **Multiplataforma:** Se puede ejecutar y desplegar en cualquier plataforma y sistema operativo, Java utiliza su propia máquina virtual para su ejecución esto es lo que permite ser multiplataforma.
- Cuenta con una Herramienta de monitoreo robusta para monitorear el rendimiento de la JVM.
- Es código abierto, no se requiere pagar licenciamiento.
- Desarrollo de aplicaciones simplificado, en lugar de utilizar archivos descriptores XML, un desarrollador puede simplemente introducir la información como una anotación directamente en un archivo de código fuente de Java, y el servidor Java EE configurará el componente en la implementación y ejecución.

Desventajas de Java EE 6.

- Un mal uso de la memoria por parte de los desarrolladores puede ocasionar que el componente de la JVM llamado “Garbage collector” consuma mucho tiempo de CPU ocasionando serios problemas de rendimiento
- Dado que las especificaciones se han convertido en un gran número, hay una gran curva de aprendizaje asociada con JEE.
- Disponibilidad de gran número librerías o bibliotecas conduce al mal uso de los desarrolladores y por lo tanto provoca que el sistema no sea interoperable a través de varios servidores de aplicaciones.
- Una mala configuración en los parámetros iniciales de la memoria de la máquina virtual de Java puede traer problemas de rendimiento considerable.

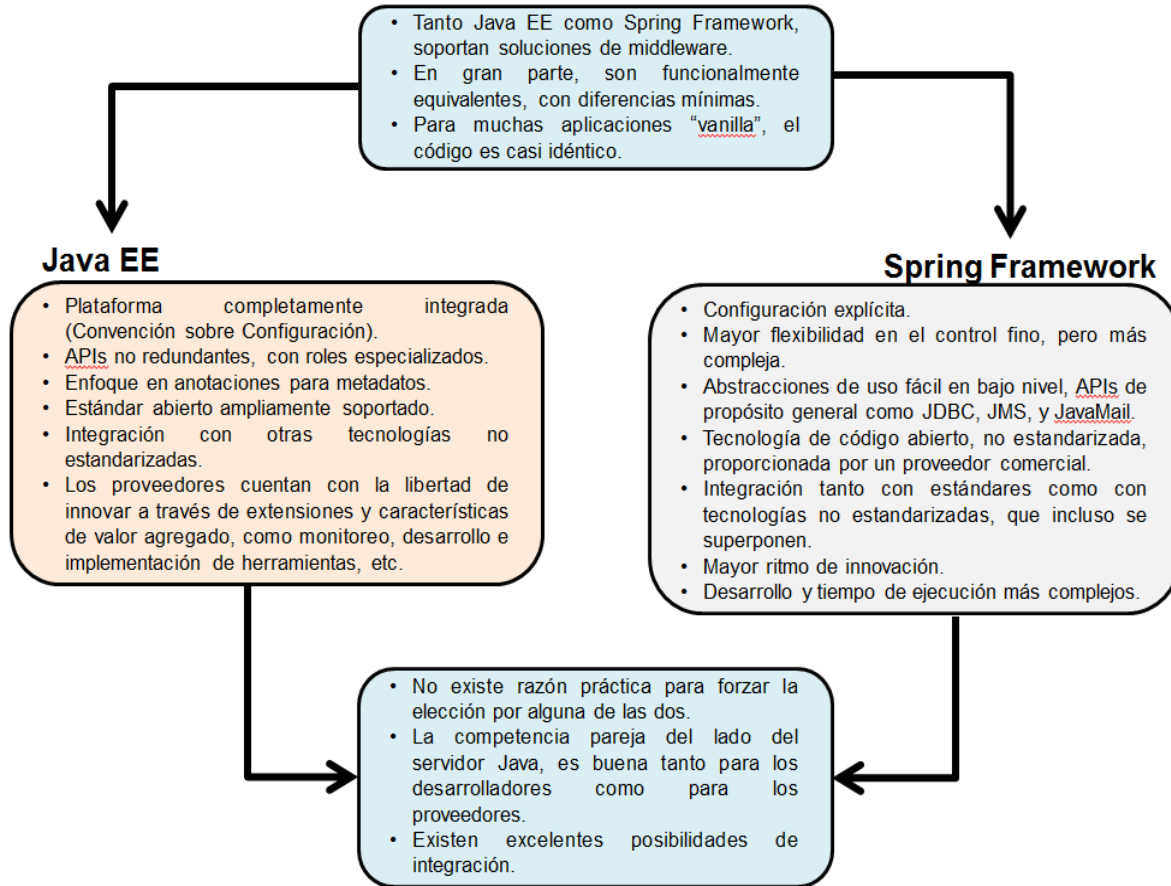


Fig. 2.3.1.2 Breve comparativa entre Java EE y Spring Framework¹².

2.3.2 Introducción de JBOSS AS 7.

Un servidor de aplicaciones es una plataforma de middleware para el desarrollo y despliegue de software basado en componentes. Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones, tareas relacionadas con el mantenimiento de la seguridad y del estado, acceso a datos y persistencia entre otros.

¹² Kelapure, R. (2011). Java EE and Spring Framework Shootout. <http://www.slideshare.net/kelapure/java-e-evsspringshootout>



Un servidor de aplicaciones puede verse como el corazón de un sistema distribuido, JBoss en particular ha sido desde sus inicios un servidor de aplicaciones extensible y modular.

JBoss AS es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., empresa fundada por Marc Fleury y que desarrolló inicialmente JBoss, fue adquirido por Red Hat en abril del 2006. El proyecto está apoyado por una red mundial de colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios. JBoss implementa todo el paquete de servicios de JEE.

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado JEE desde la versión 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business.

Combinando una arquitectura orientada a servicios revolucionaria con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware para desarrolladores, vendedores independientes de software y, también, para grandes empresas.

JBoss Application Server 7 es la última versión, es una rápida y poderosa implementación de la especificación de Java Enterprise Edition 6. En esta nueva



versión se hicieron grandes cambios en la arquitectura, principalmente en el kernel, permitiendo habilitar servicios bajo demanda cuando una aplicación lo requiere haciendo a JBoss AS mucho más modular.

JBoss As 7 es una plataforma de middleware basada en estándares abiertos, y compatible con Java EE. Integra JBoss Application Server 7 con alto nivel de disponibilidad, servicios de mensajería, almacenamiento en caché distribuido, y otras tecnologías para crear una plataforma estable, escalable y rápido. Además, también incluye APIs y los marcos de desarrollo que se pueden utilizar para desarrollar segura, potente, y escalables aplicaciones Java EE rápidamente.

Características de JBOSS AS 7.

JBOSS AS 7 tiene varias características en la que se incluye las siguientes¹³:

- JBoss Enterprise Application Plataforma 6 es una aplicación certificada de la Java Enterprise Edition 6.
- Permite utilizar un dominio administrado permitiendo una gestión centralizada de múltiples instancias de servidor desplegadas en diferentes equipos físicos.
- Configuraciones, despliegues, fijaciones de conector, módulos, extensiones y las propiedades del sistema pueden ser todo gestionado por un grupo de servidores.
- La administración del servidor de aplicaciones puede ser por medio de la consola de administración o administración por CLI que son interfaces para la gestión de dominio o instancia independiente.

¹³ Chaudhary N., Dickenson R., Gilda S. (2013). Jboss Enterprise Application Platform 6 Administration and Configuration Guide. https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6/pdf/Administration_and_Configuration_Guide/JBoss_Enterprise_Application_Platform-6-Administration_and_Configuration_Guide-en-US.pdf



- La seguridad de aplicaciones, incluyendo dominios de seguridad, se administra de forma centralizada.
- El diseño del directorio de JBoss AS 7 se ha simplificado. El directorio /modules/ es el directorio que contiene ahora los módulos de servidor de aplicaciones, en lugar de utilizar el directorio /lib. Los directorio /domain y /standalone contienen los artefactos y archivos de configuración de dominio y las implementaciones independientes.
- El mecanismo de carga de clase se ha hecho completamente modular, de modo que los módulos se cargan bajo demanda. Esta proporciona beneficios de rendimiento y seguridad.
- La administración del pool de conexiones hacia diferentes fuentes de datos se hace más eficiente. Los controladores de base de datos se pueden implementar al igual que otros servicios. Además, las fuentes de datos se crean y administran directamente en la consola de administración o CLI.
- Aplicación JBoss 7 arranques y paradas muy rápidamente, lo que es especialmente beneficioso para los desarrolladores. Se utiliza menos recursos y es extremadamente eficiente en su uso de los recursos del sistema.
- Soporta configuración en modo en alta disponibilidad, lo que lo hace más robusto y escalable.

Ventajas de JBoss AS 7.

- JBoss AS 7 escala a los niveles requeridos por las más exigentes aplicaciones. El modelo de conexión híbrida puede manejar cargas de clientes de 10,000 conexiones simultáneas.
- Producto de licencia de código abierto sin costo adicional.
- Certificado y soportado por Red Hat.



- Fácil de administrar, se cuenta con consola web de administración e interfaces de comandos con CLI.
- Soporte de alta disponibilidad, lo que lo hace más escalable y robusto el servidor de aplicaciones.
- Multiplataforma, puede correr sobre cualquier plataforma, debido a las características de la máquina virtual de java.
- Cumple con los estándares del mercado.
- Confiable a nivel de empresa, ya que está respaldado por Red Hat y se puede comprar soporte directamente con Red Hat.
- Modulable y cumple con una arquitectura orientada a servicios.
- Ayuda profesional 24x7 de la fuente y con costo directamente con Red Hat.
- Disponibilidad y portabilidad: permite múltiples plataformas y sistemas operativos.
- Cumple con las expectativas de negocio y tecnológicas.
- Robusto, ya que es tolerante a fallos.
- Fácil mantenimiento: fácil mantenimiento del servidor de aplicaciones mediante la consola de administración o interfaces de línea de comando CLI.

Desventajas de JBoss AS 7.

- Al inicio no se contaba con mucha información de guía de instalación y administración, si no se tenía el conocimiento adecuado podría ser un problema severo.
- Un cambio de versión, genera un impacto alto en el código fuente de las aplicaciones.
- Cambios continuos de versiones, ocasionan problema de soporte por parte de Red Hat.



CAPÍTULO II



Servidor de Aplicaciones	Proveedor	Licencia	Perfil	Disponibilidad de soporte por parte del Proveedor	Versión Java
GlassFish Server 3.01	Oracle	OTN / Comercial	FP	Sí	HotSpot 6/7
GlassFish Server Open Source Edition 3.x	Oracle	GPL + CDDL	FP	No	HotSpot 6/7
WebSphere Application Server 8.x	IBM	Comercial (prod+dev)	FP	Sí	IBM JVM 7
WebSphere Application Server Community Edition 3.0	IBM	IBM International License Agreement for Non-Warranted Programs	FP	Sí	IBM JVM 7
JEUS 7	TMAX	Comercial	FP	Sí	HotSpot 6
Interstage Application Server powered by Windows Azure	Fujitsu	Comercial	FP	Sí	HotSpot 6
Interstage Application Server v10.1	Fujitsu	Comercial	FP	Sí	HotSpot 6
Geronimo 3.0-beta-1	Apache	Apache 2.0	FP	No	HotSpot 6
WebLogic Server 12.1.1	Oracle	Comercial / Free for Dev	FP	Sí	HotSpot 6/7
uCosminexus Application Server v9.0	Hitachi	Comercial	FP	Sí	?
JBoss Application Server 7.x	RedHat	LGPL	FP	No	HotSpot 6
JBoss Enterprise Application Platform 6	RedHat	LGPL / Comercial	FP	Sí	HotSpot 6
Resin 4.0.17	Caucho	GPL "Resin Open Source" version "Resin Professional" Comercial	WP	Sí	HotSpot 6
TomEE 1.0	Apache	Apache 2.0	WP	No	HotSpot 6/7
JOnAS 5.3.0-M8-SNAPSHOT	OW2	LGPL	WP	No	HotSpot 6/7
NetWeaver Cloud	SAP	Comercial	WP	Sí	SAP Java Server VM 1.6

Tabla 2.3.2.1 Breve comparativa entre servidores de aplicaciones compatibles con Java EE 6¹⁴.

¹⁴ Eisele, M. (2013). Selecting your Java EE 6 Application Server. <http://blog.eisele.net/2013/01/selecting-your-java-ee-6-application.html>



2.4 Características, ventajas y desventajas de Oracle Database Server.

Introducción Oracle Database Server 11g.

La base de datos Oracle 11g presenta capacidades de programación de alto nivel que permite el uso versátil de la información y facilita el trabajo con los datos desde entornos sencillos, pasando por los entornos cliente-servidor hasta llegar a los entornos distribuidos¹⁵.

Oracle Base de Datos 11g ofrece un rendimiento y una escalabilidad excepcionales en servidores Windows, Linux y UNIX, y aporta un rápido rendimiento de la inversión porque permite pasar de un solo servidor a Grid Computing sin modificar ni una sola línea de código.

Oracle Base de Datos 11g automatiza las tareas de administración y ofrece las mejores funciones de seguridad y de cumplimiento de las normativas, por lo que consigue resultados óptimos. Gracias a Real Application Clusters, se obtienen los mayores niveles de disponibilidad. Como ofrece distintas ediciones y unos costos operativos más bajos que IBM DB2 y Microsoft SQL Server, es la opción ideal para empresas en expansión. Existen diferentes ediciones que se ajustan a las diferentes necesidades de las organizaciones.

¹⁵ Pérez, M. (2013). Programación orientada a objetos con ORACLE PL/SQL



Todas las ediciones de Oracle Database 11g parten de la misma arquitectura confiable de motor de base de datos y son totalmente compatibles entre ellas. Están disponibles para una variedad de sistemas operativos e incluyen una serie de herramientas en común para el desarrollo de aplicaciones e interfaces de programación. Con Oracle Database 11g, se puede comenzar instalando Standard Edition One y, a medida que la empresa crezca, actualizar fácilmente a Standard Edition o Enterprise Edition de acuerdo con lo que se ajuste mejor a sus necesidades. Una de las ventajas de los productos de Oracle es que la actualización es muy sencilla: lo único que tiene que hacer es instalar el software de la siguiente edición y, sin cambiar nada en su base de datos o aplicaciones, ya tiene a su disposición el rendimiento, la escalabilidad, la confiabilidad y la seguridad que caracterizan a Oracle en entornos fáciles de administrar, a continuación se mencionan las características principales.

Características de Oracle.

Oracle Database tiene varias características¹⁶ en la que se incluye las siguientes:

- **Oracle Database 11g Standard Edition One** ofrece innovación en facilidad de uso, capacidad y relación precio/rendimiento para aplicaciones de grupos de trabajo, departamentales y web en servidores únicos con un máximo de dos **sockets**.
- **Oracle Database 11g Standard Edition** se puede usar en un solo servidor o en servidores agrupados con una capacidad máxima de cuatro sockets en total. En el paquete normal, se incluye Oracle Real Application Clusters (Oracle RAC) sin costo adicional.

¹⁶ Teaching soft group. (2012). Programación en Oracle 11g. SQL, SQL*Plus y PL/SQL.



- **Oracle Database 11g Enterprise Edition** se puede usar en un solo servidor o en servidores agrupados con una cantidad ilimitada de sockets. Con esta versión, es posible gestionar la información de manera eficiente, confiable y segura en aplicaciones transaccionales indispensables, almacenes de datos con alto tráfico de consultas y cargas mixtas de trabajo.
- **Oracle Active Data Guard**, una opción de Oracle Database 11g Enterprise Edition, mejora la calidad del servicio y la utilización de hardware redundante por medio de la descarga de actividades que ocupan muchos recursos. Las mueve de una base de datos de producción a una o más bases de datos de reserva sincronizadas. Con Oracle Active Data Guard, se obtiene acceso de sólo lectura a una base de datos física de reserva para realizar consultas, organizar información, solicitar informes, acceder a Internet, etc., al tiempo que se aplican constantemente los cambios provenientes de la base de datos de producción. Asimismo, Oracle Active Data Guard tiene la capacidad de abrir una base de datos de reserva para realizar pruebas y después revertir las modificaciones con una base de datos de reserva viable si fuese necesaria la recuperación en caso de desastre. Durante la etapa de pruebas, no se pierden los cambios realizados a la base de datos de producción. Con Oracle Active Data Guard, también es posible usar copias de seguridad incrementales rápidas cuando se descargan a una base de datos de reserva. También cuenta con los beneficios adicionales de la alta disponibilidad y la protección en caso de desastre o interrupción del servicio planificada o espontánea en el lugar de producción.



- **Oracle Advanced Security** proporciona funciones de cifrado transparente para los datos guardados en la base de datos y cifrado de redes para la información que circula en la red. También incluye una completa suite de servicios de autenticación de la base de datos de Oracle. El cifrado de redes se implementa por medio de algoritmos de cifrado y de integridad de los datos de uso habitual en el sector. En esta opción se incluyen distintas potencias de algoritmos y de cifrado para el momento de la implementación. Los sólidos servicios de autenticación sirven de apoyo a un amplio paquete de opciones de autenticación de terceros estándar en el sector. Las opciones de autenticación incluyen servicios de inicio único de sesión en la base de datos de Oracle mediante la interoperación con estructuras de autenticación existentes, y opciones de autenticación de dos factores, como las tarjetas inteligentes.
- La opción **Oracle OLAP** es un completo servidor de procesamiento analítico en línea (OLAP) incorporado a la base de datos de Oracle. Con esta opción, se pueden mejorar las herramientas y aplicaciones de inteligencia empresarial basadas en SQL mejorando el rendimiento de las consultas y aportándoles contenidos analíticos. Como solución OLAP estrechamente incorporada a la base de datos de Oracle, la opción Oracle OLAP ofrece una gestión centralizada de datos y reglas de negocio en una plataforma segura, de arquitectura escalable y pensada para empresas.
- La opción **Oracle Partitioning** mejora el entorno de gestión de datos para procesamiento de transacciones en línea y aplicaciones del almacén de datos y de almacenes de datos especializados incrementando la capacidad de administración, la disponibilidad y el rendimiento de grandes tablas e índices de bases de datos subyacentes. Oracle Partitioning permite la división de tablas grandes en partes más pequeñas de administración individual sin perder la visión integral de los datos a nivel de la aplicación. Admite una variedad muy amplia de métodos de particionamiento, entre los que se incluyen la posibilidad de dividir tablas muy grandes (y sus



respectivos índices) y convertirlas en unidades más pequeñas y manejables, con lo que se adopta el enfoque "divide y vencerás" para la gestión de bases de datos muy voluminosas. El particionamiento también contribuye a un mejor rendimiento, ya que el optimizador se encargará de limpiar las consultas para poder usar solamente las particiones pertinentes de una tabla o de un índice durante una búsqueda.

- **Oracle Real Application Clusters (RAC)** aprovecha el poder de procesamiento de varios servidores interconectados en un clúster. Al permitir que los distintos servidores del grupo accedan a una sola base de datos, se reduce el riesgo de fallas del servidor tanto para las aplicaciones como para los usuarios. También ofrece rendimiento ampliable a demanda y a un bajo costo, y es un componente vital de la computación en mallas que posibilita el acceso simultáneo de varios servidores a una sola base de datos. Oracle Database 11g también incluye Automated Storage Management (ASM) y Oracle Clusterware. Si se combina el uso de ASM con el de Oracle Clusterware, se virtualiza el almacenamiento, los servidores de base de datos, los servidores de aplicaciones, la gestión holística y todos los otros aspectos relacionados con la implementación y la administración de un entorno informático virtualizado.

Ventajas de la base de datos Oracle.

- Oracle es el motor de base de datos objeto-relacional más usado a nivel mundial.
- Ahorra una cantidad considerable de tiempo y esfuerzo a través del ciclo de vida de administración.
- Alta disponibilidad para una gran variedad de bases de datos de aplicaciones OLTP, personalizadas y OLAP.



- Reduce el tiempo de inactividad planificado y no planificado mediante el monitoreo automático y la solicitud de servicios a Oracle Support.
- Puede ejecutarse en todas las plataformas, desde una Pc hasta un supercomputador.
- Protege las bases de datos ante fallas del servidor y de almacenamiento con Oracle Real Application Clusters y Automatic Storage Management.
- Reduce el costo con flexibilidad para así ajustar el gasto en software al crecimiento de la empresa sin necesidad de actualizar el hardware.
- Manejo de un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con **triggers** (disparadores) y procedimientos almacenados, con una integridad referencial declarativa bastante potente.
- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.
- Existe incluso una versión personal para Windows x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.
- Base de datos altamente segura que permite a las empresa cumplir con el reglamento regulatorio.



Desventajas de Oracle.

- Las versiones más recientes de Oracle son la 12c, 11g, 10g, 9i, 8, desde el lanzamiento original de la 8 se sucedieron varias versiones con correcciones, hasta alcanzar la estabilidad. El motivo de tantos fallos fueron los problemas en el manejo del sistema de almacenamiento, esto es una desventaja para usar Oracle ya que cuestiones de soporte ocasiona actualizaciones continuas a los productos.
- El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de personal Oracle son excesivamente caras.
- Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
- Se requiere el personal altamente capacitado para la administración e implementación de Oracle, una base de datos mal implementada o mal administrada puede ser un fuerte problema en cuestiones de rendimiento.
- También es elevado el costo de la información, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.



CAPÍTULO II



	Creador	Fecha de la primera versión pública	Última versión estable	Licencia de software
Adaptive Server Anywhere	Sybase/Anywhere	1992	10.0	Propietario
Adaptive Server Enterprise	Sybase Inc	1987	15.0	Propietario
ANTs Data Server	ANTs Software	1999	3.6	Propietario
DB2	IBM	1982	9	Propietario
Firebird	Firebird Foundation	25 de julio de 2000	2.5	Licencia Pública InterBase
Informix	Informix Software	1985	10.0	Propietario
HSQLDB	Hsqldb.Org	2001	1.9	Licencia BSD
Ingres	Berkeley University, Computer Associates	1980	2006	CA-TOSL
InterBase	Borland	1985	7.5.1	Propietario
SapDB	SAP AG	?	7.4	GPL con drivers LGPL
MaxDB	MySQL AB, SAP AG	?	7.7	GPL o propietario
Microsoft SQL Server	Microsoft	1989	2014	Propietario
MySQL	MySQL AB	Noviembre de 1996	5.0	GPL o propietario
Oracle	Oracle Corporation	1977	11g Release 2	Propietario
PostgreSQL	PostgreSQL Global Development Group	Junio de 1989	9.0	Licencia BSD
SmallSQL	SmallSQL ↗	16 de abril de 2005	0.12	LGPL
SQLite	D. Richard Hipp	17 de agosto de 2000	3.6.16	Dominio público
Microsoft Access	Microsoft	1992	2013	Propietario
	Creador	Fecha de la primera versión pública	Última versión estable	Licencia de software

Fig. 2.4.1 Información general de sistemas gestores de bases de datos¹⁷.

¹⁷ Wikipedia. (2016). Anexo: Comparación de sistemas administradores de bases de datos relacionales. https://es.wikipedia.org/wiki/Anexo:Comparación_de_sistemas_administradores_de_bases_de_datos_relacionales



2.5 Características, ventajas y desventajas de SPRING MVC 3 y JSF 2.1.

2.5.1 Introducción de SPRING MVC 3.

En 2005, un año antes de la salida de Java EE 5, Rod Johnson funda la empresa SpringSource. Más adelante, Spring Framework incorpora las anotaciones a su inyector de dependencias, pero lo hace mucho tiempo después; recién en la versión 2.5 podemos hacer un uso completo de ellas. Aunque EJB 3.0 y JPA presentan una mejora considerable, aún siguen teniendo desventaja frente a Spring en funcionalidades como Orientación a Aspectos e Inyección de Dependencias. Spring 3.0 fue liberada en diciembre de 2009, y Spring 3.1 dos años más tarde.

Spring¹⁸ es un framework de código abierto para la plataforma JEE, que permite el desarrollo rápido de aplicaciones. Spring MVC (Modelo-Vista-Controlador) es el componente web del framework que proporciona una funcionalidad completa para la elaboración de aplicaciones web. La arquitectura de Spring MVC se ha diseñado de tal manera que cada componente y funcionalidad es altamente configurable. También se puede integrar fácilmente con otros frameworks como Struts y JSF (Java Server Faces).

Las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen varias extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar de que no impone ningún modelo de programación en particular, este framework se ha vuelto

¹⁸ Johnson R., Hoeller J., Donald K. (2004-2012). Spring Framework Reference Documentation.



popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (Enterprise JavaBean).

Spring MVC está diseñado en torno a un DispatcherServlet que envía solicitudes a los controladores, el controlador predeterminado se basa en una anotación @Controller y @RequestMapping, que ofrecen una amplia gama de métodos de control flexibles, hacia la capa del negocio.

La clase DispatcherServlet está en la capa de Front y es responsable de delegar y coordinar el control entre varias interfaces en la fase de ejecución durante una petición HTTP. Las interfaces más importantes definidas en Spring MVC, y sus responsabilidades, son las siguientes:

- HandlerMapping: Permite manejar peticiones de entrada.
- HandlerAdapter: Ejecución de objetos que permiten manejar las peticiones entrantes.
- Controller: Está entre el modelo y la vista, y permite manejar peticiones entrantes y redirigirlas a la respuesta adecuada.
- ViewResolver: Selecciona una vista basada en un nombre lógico de la vista.
- HandlerInterceptor: Intercepta las peticiones entrantes, es comparable pero no igual a los filtros de Servlet.
- LocaleResolver: Resuelve y opcionalmente salva la región y hora de un usuario individual.
- MultipartResolver: Facilita trabajar con ficheros de subida con peticiones de entrada.

Estas interfaces ofrecidas son potentes, pues permiten configurar un conjunto de



interfaces juntas y ofrecer funcionalidades más robustas. Sin embargo, los desarrolladores son libres de escribir otras implementaciones.

Características Spring 3.

Algunas características que pueden utilizar las aplicaciones empresariales por medio de Spring MVC 3 son las siguientes:

- Spring 3 Framework soporta la plataforma JEE 5 y 6.
- Es código abierto.
- Soporta configuración basada en anotaciones.
- Utiliza un archivo XML DispatcherServlet como controlador.
- Spring 3.0 framework soporta servicios REST (servicios sin estados).
- Anotaciones para formato, validación y conversión de datos.
- Spring 3 soporta integración con JPA 2 y EJB.
- Simplifica la programación orientada a aspectos.
- Simplifica el acceso a datos o Beans.
- Integración con JSF.
- Soporte acceso a componentes remotos.
- Manejo simplificado de excepciones.

Ventajas de Spring MVC 3.

Entre las principales ventajas que presenta Spring MVC 3, tenemos las siguientes:

- Spring MVC ofrece una división limpia entre las capas Controlador, Modelo (JavaBeans) y Vista.
- Spring MVC es muy flexible, ya que implementa toda su estructura mediante interfaces.



- Spring MVC provee interceptores y controladores que permiten interpretar y adaptar el comportamiento común en el manejo de múltiples peticiones.
- Se enlaza fácilmente con los Beans de negocio.
- Integración fácil con JSF.
- Soporte acceso a componentes remotos.
- Soporta configuración basada en anotaciones.
- Reutilización de código. Utilización de los objetos existentes como comandos o formularios, en lugar de la duplicarlos para extender una clase particular del Framework.
- Soporta internacionalización.
- Una biblioteca de etiquetas JSP que proporciona soporte para funciones como el enlace de datos con el servidor que hace la escritura en las páginas sea mucho más fácil.
- Manejo simplificado de excepciones.
- Programación modular.
- Altamente configurable.

Desventajas de Spring MVC 3.

Algunas desventajas que se pretende implementar en las aplicaciones empresariales por medio de Spring MVC 3 son las siguientes:

- Alta configuración de archivos XML que pueden ocasionar problemas de mantenimiento en la aplicación.
- Uso considerable de memoria por el contenedor de Spring.

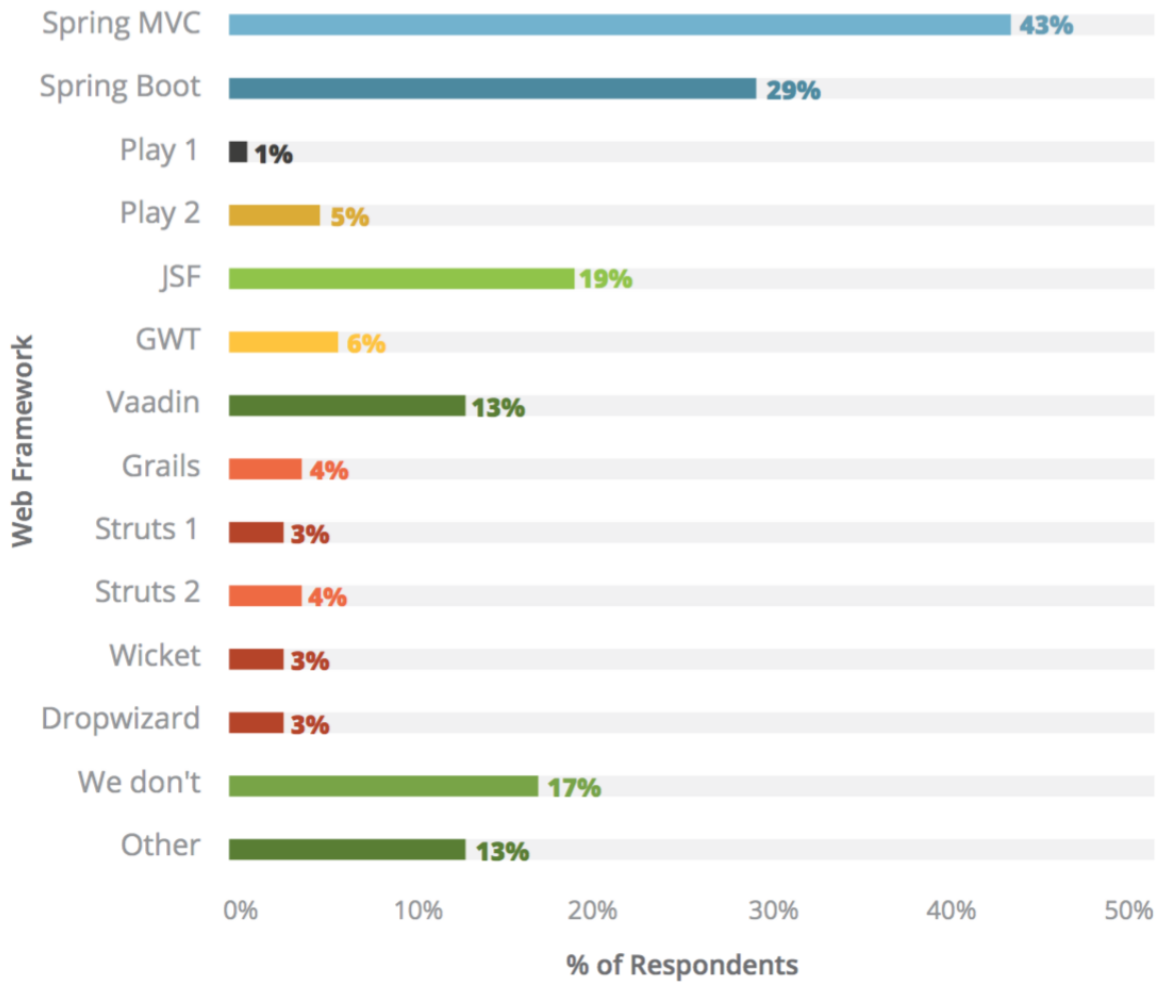


Fig. 2.5.1.1 Grado de utilización de Web Frameworks (abril 2016)¹⁹.

¹⁹ Rebellabs. (2016). Java Tools and Technologies Landscape Report 2016. <http://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016/>



2.5.2 Introducción de JSF 2.2.

La tecnología JSF nació en el año 2004, pero no fue hasta 2006 cuando comenzó a ser popular al integrarse con Java EE 5. Actualmente, la última versión disponible de JSF es la versión 2.2, la cual ha cambiado considerablemente desde 2006. JSF difiere en gran medida de los desarrollos con PHP o JSP. Para empezar se suele separar en dos grupos de archivos, los que contienen la presentación de la página propiamente dicha (xhtml), y los que contienen la lógica de presentación (Java). En los archivos xhtml (*.xhtml) se usan componentes de JSF que gestionan todos los formularios, campos, tablas de datos, así como los estilos y Javascript para los mismos. Mientras tanto, en los archivos java (*.java) configurados, se realizan las peticiones a base de datos, el procesado de la información o el registro de acciones, entre otros.

La tecnología Java Server Faces²⁰ es un framework de interfaz de componentes de usuarios del lado del servidor para aplicaciones web basadas en el desarrollo de interfaces de usuario en aplicaciones Java EE. Los principales componentes de la tecnología JSF son:

- Una API para representar componentes de Interfaz de Usuario (UI) y gestionar su estado:
 - Manejar eventos, validar en el servidor y conversión de datos.
 - Definir la navegación de páginas.
 - Soporte de internacionalización y accesibilidad.
- Librerías de etiquetas JSP personalizadas para expresar componentes en una página JSP y enlazar los componentes a objetos del servidor.

²⁰ Leonard, A. (2010). JSF 2.0 CookBook.



Para el desarrollo de este proyecto se ha optado por la tecnología Java, concretamente, por la tecnología Java Server Faces (JSF) como núcleo principal para desarrollo de la capa de presentación. Esta tecnología ofrece componentes extendidos (respecto a HTML estándar) muy usados en el mundo de la programación web, muchos de estos componentes son muy robustos e interactúan bien unos con otros, dando la capacidad de generar una página web completa con pocas líneas de código.

JSF utiliza un modelo de programación bien definido y las librerías de etiquetas facilitan la construcción y mantenimiento de las aplicaciones web con Interfaces de Usuario (UI) de servidor. Con un mínimo esfuerzo se podría:

- Poner componentes en una página mediante etiquetas de componentes.
- Enlazar eventos generados por componentes con código de la aplicación en el servidor.
- Relacionar componentes UI en una página con datos del servidor.
- Construir una UI con componentes reutilizables y extensibles.
- Salvar y restaurar el estado de la UI más allá de la vida de las peticiones.

JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías.



Características de JSF 2.2.

JSF 2.2 tiene varias características en la que se incluye las siguientes:

- Utiliza páginas JSP para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios HTML.
- Asocia a cada vista con formularios un conjunto de objetos java manejados por el controlador que facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.
- Introduce una serie de etapas en el procesamiento de la petición, como por ejemplo la de validación, reconstrucción de la vista, recuperación de los valores de los elementos, etc.
- Utiliza un sencillo fichero de configuración para el controlador en formato xml.
- Es extensible, pudiendo crearse nuevos elementos de la interfaz o modificar los ya existentes.
- Forma parte del estándar JEE. En efecto, hay muchas alternativas para crear la capa de presentación y control de una aplicación web Java, como Struts y otros Frameworks, pero solo JSP forma parte del estándar.
- Permite definir un esquema de navegación de las páginas.
- Soporta internacionalización.
- Bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.



Ventajas de JSF 2.2.

- El código JSF con el que creamos las vistas (etiquetas JSP) es muy parecido al HTML estándar. Lo pueden utilizar fácilmente desarrolladores y diseñadores web.
- JSF se integra dentro de la página JSP y se encarga de la recopilación y generación de los valores de los elementos de la página.
- JSF tiene un conjunto de validadores a nivel de la vista que permite reducir el tiempo de desarrollo en validaciones de campos.
- JSF tienen un conjunto de mensajes de error que permite fácilmente con etiquetas mostrar en pantalla los mensajes de error.
- Soporta internacionalización.
- JSF permite introducir JavaScript en la página, para acelerar la respuesta de la interfaz en el cliente (navegador del usuario).
- JSF es extensible, por lo que se pueden desarrollar nuevos componentes a medida.
- Debido al amplio repertorio de implementaciones o de componentes, JSF permite optimizar mucho el código.
- Se divide físicamente la parte de presentación y lógica para la presentación.
- Permite utilizar Ajax de manera integrada creando navegaciones muy flexibles.
- Ofrece gran cantidad de componentes Opensource.
- Los APIs de la tecnología JavaServer Faces se han creado directamente sobre el API JavaServlet. Esto permite hacer algunas cosas interesantes como usar otra tecnología de presentación junto a JSP como crear componentes propios personalizados.
- Ofrece una rápida adaptación para nuevos desarrolladores.
- Proporciona una rica arquitectura para manejar el estado de los componentes, procesar datos, validar la entrada del usuario y manejar eventos.



- Fácil mantenimiento a las aplicaciones.

Desventajas de JSF 2.2.

- Su naturaleza como estándar hace que la evolución de JSF no sea tan rápida como pueda ser la de otros entornos como WebWork, Wicket y Spring.
- Requiere de buen aprendizaje. La integración con otros lenguajes como JavaScript o JQuery complica más la tecnología.

	JSF	JSP
Componentes.	Ofrece un componente de interfaces de usuario (UI) por eventos.	Usa JSTL (JSP Standard Tag Library) que no son interfaces de usuario.
Manejo de Independencia.	Un conjunto de librerías que ofrece manejo de independencias.	No usa.
Manejo de errores y validación.	Utiliza un Framework de validación y validadores predefinidos.	No usa.
Scripting.	Se puede adjuntar scripts dentro de los eventos, sus componentes son accesibles por medio de scripts.	Embebe Java dentro del JSP.
Flujo de páginas.	Manejo de navegación dentro del archivo faces-config.xml.	No usa.

Tabla 2.5.2.1 Breve comparativa entre JSF y JSP.



MARCO TEÓRICO





CAPÍTULO III

DESCRIPCIÓN DEL SISTEMA WEB

3.1 Análisis del problema.

3.1.1 Situación actual.

El estado en el que se encuentra la administración del posgrado en la universidad se representa en la figura 3.1.1.1. Actualmente los alumnos de posgrado realizan sus trámites de forma manual, entregando la documentación impresa. De igual forma, la interacción con sus profesores y asesores es de forma directa, controlando de forma manual los avances alcanzados, por ejemplo, los de sus trabajos de tesis. Las observaciones hechas por sus asesores solamente son del conocimiento de ambas partes. La trayectoria de los alumnos (curriculum vitae) solo es almacenada de forma documental, lo que hace su consulta lenta y tediosa.

De forma similar, los profesores solo pueden consultar la trayectoria de los alumnos, y los avances académicos de estos, de forma directa. No existe base de datos alguna con tal información. Por supuesto, tampoco los profesores cuentan con un medio para proporcionar información sobre su trayectoria profesional.



DESCRIPCIÓN DEL SISTEMA WEB

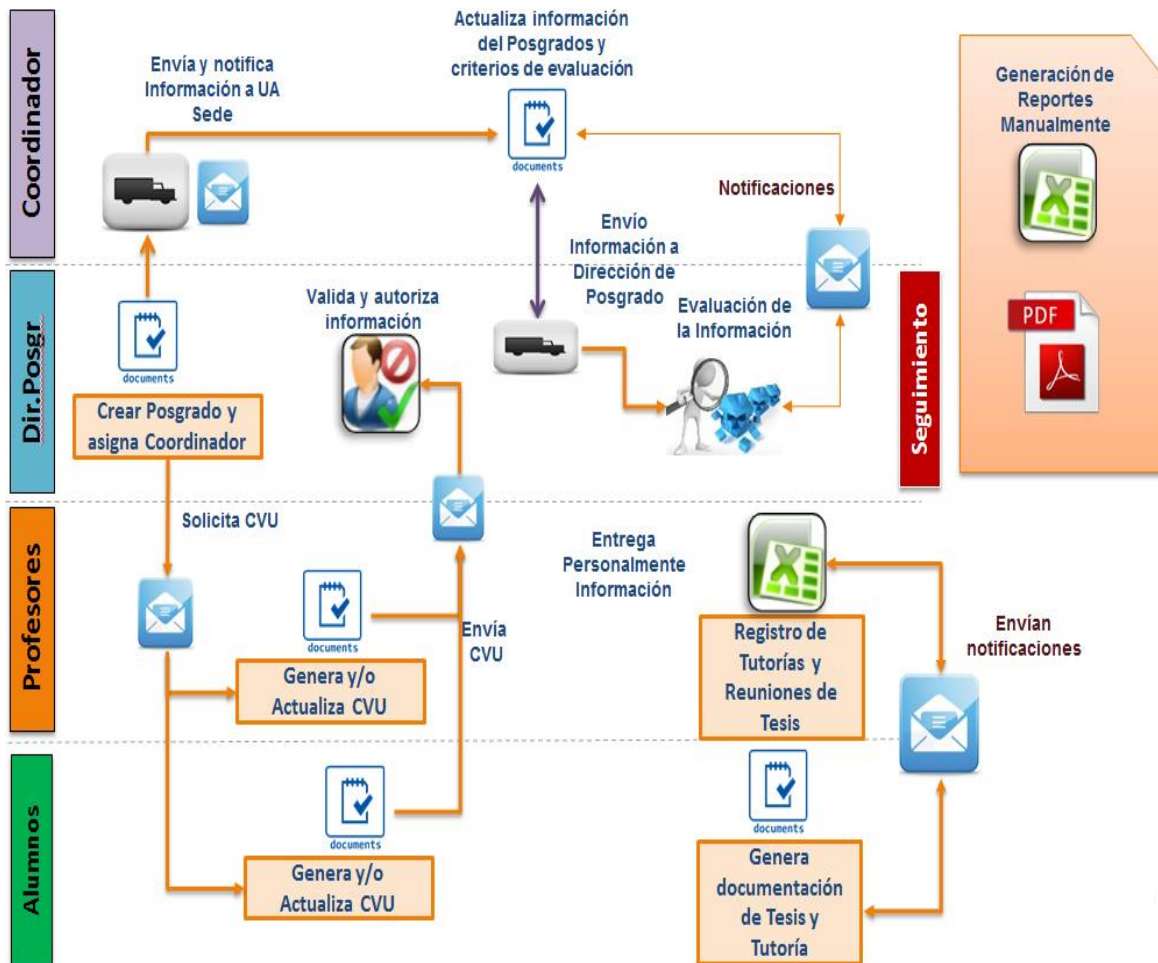


Fig. 3.1.1.1 Situación actual de la gestión del posgrado en la universidad

Los coordinadores de cada uno de los posgrados tampoco cuentan con información sobre las trayectorias académicas y profesionales de alumnos y profesores, a menos que la soliciten personalmente. Cuando es necesario abrir un nuevo programa, el proceso se realiza de forma manual, lo que en ocasiones genera retrasos en los trámites ante el CONACyT y la SEP, lo que redundaría en problemas administrativos para los alumnos que ingresan a esos programas.



Desde la dirección del posgrado de la universidad, es imposible seguir la trayectoria de sus egresados. Mucho menos cuenta con mecanismos ágiles que les permita conocer los resultados de cada uno de los programas que imparten, lo que repercute en una baja eficiencia de tales programas, y no les permite planificar adecuadamente métodos que contribuyan a elevar la productividad académica, o la apertura de nuevos programas.

En resumen, la universidad actualmente no cuenta con un mecanismo tecnológico que permita mejorar la calidad de la educación y el seguimiento, en particular a los posgrados. Para lograr este objetivo se requiere una herramienta tecnológica que ayude a cumplir el seguimiento de los posgrados de los estudiantes, así como de los egresados y profesores de los posgrados de la institución educativa, proporcionando mecanismos de consulta y administración de la información más rápida y eficaz.

3.1.2 Solución propuesta.

Dada la situación actual que presenta la universidad, el objetivo es analizar y diseñar una herramienta tecnológica que ayude a cumplir con el seguimiento de los posgrados de la institución educativa, proporcionando mecanismos más rápidos y eficaces para la consulta y administración de la información.

Para conseguir el objetivo citado, en la figura 3.1.2.1 se presentan los módulos que integran la propuesta de la solución.

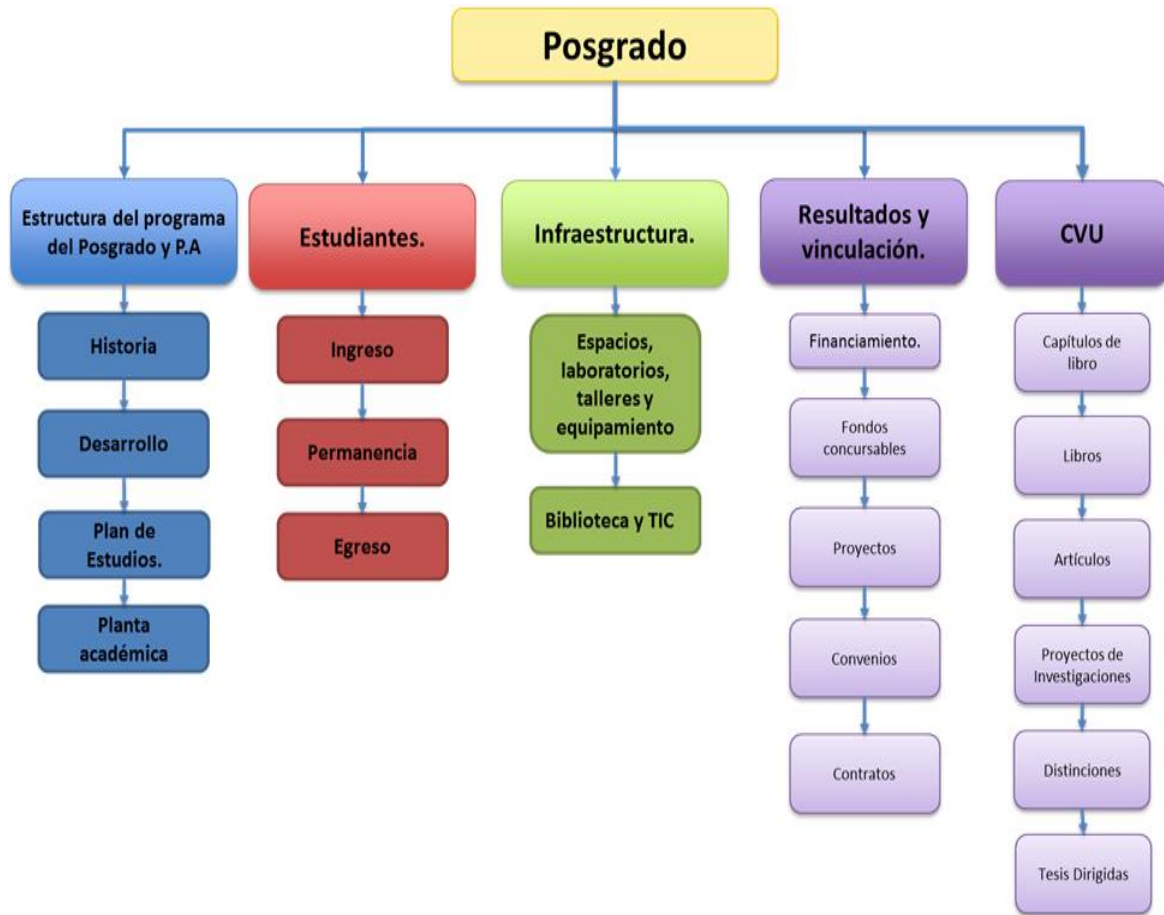


Fig. 3.1.2.1 Módulos de la solución

Los **beneficios** de la solución presentada son los siguientes:

- **Información centralizada.** Todos los documentos se encontrarán en un solo lugar y se almacenarán de manera segura.
- **Consulta fácil y rápida.** Reducirá tiempos de generación de documentos; no se requiere que alguien tenga la información concentrada en un equipo específico. La información estará disponible siempre para cualquier usuario del sistema.



- **Datos actuales.** Al usar el sistema de manera oficial, no se manejará información fuera del sistema, por lo tanto cualquier cambio se actualizará en el momento en que se requiere.
- **Reportes.** Los reportes se obtendrán de manera rápida y en el momento, no se tendrá que esperar a que alguien actualice la información.

La **metodología** que se utilizará para llevar a cabo las soluciones CASE. Como se describió en el Capítulo II, se compone de las siguientes etapas:

- **Análisis.**
 - Definición de documentación.
 - Fechas de inicio y entrega.
 - Sesiones de consultoría.
 - Entender y documentar los requerimientos.
 - Preparar el plan de trabajo del proyecto.
 - Analizar y valorar los riesgos del proyecto.
 - Documento de visión y alcance.
- **Diseño.**
 - Creación de documentos de diseño técnico y funcional.
 - Generación de prototipos.
 - Definir equipo de trabajo, asignación de actividades, roles y responsabilidades.
 - Asegurar una fácil implementación para futuras mejoras de la aplicación.



- **Desarrollo.**
 - Instalar entorno técnico/operativo de la solución.
 - Desarrollar la solución.
 - Realizar pruebas con la solución.
 - Obtener la aceptación.

- **Implementación.**
 - Instalar la solución en servidores productivos.
 - Probar el correcto funcionamiento de la solución.
 - Capacitar a los usuarios finales.
 - Obtener la aceptación y validar los requerimientos y fecha de arranque (**Go-Live**).
 - Documentación de solución, memorias técnicas, código fuente, etc.

- **Operación.**
 - Determinar si las metas y objetivos del proyecto se alcanzaron.
 - Proporcionar soporte remoto a los usuarios de la solución.
 - Monitorear la solución.
 - Generar la documentación de cierre.

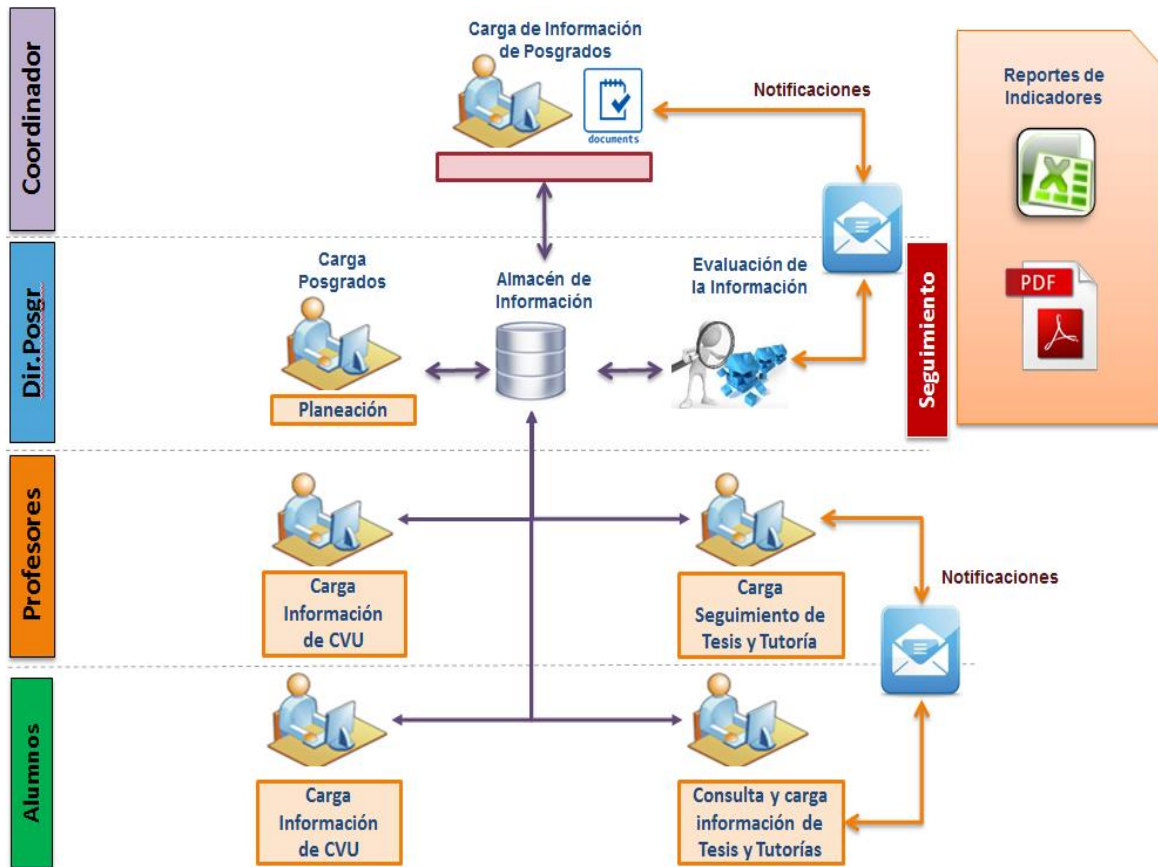


Fig. 3.1.2.2 Gestión del posgrado de la universidad con la solución aplicada

La figura 3.1.2.2 resume la gestión del posgrado de la universidad cuando la solución presentada esté en operación. Básicamente el sistema de gestión del posgrado permitirá realizar lo siguiente:



- **Estructura del posgrado.**
 - Consulta, alta, edición de los posgrados, donde el director del posgrado podrá dar de alta los nuevos posgrados o los posgrados actuales, el director de posgrados podrá subir información de la clave del posgrado, coordinador del posgrado, Unidad académica Sede, fecha de creación del posgrado, estado del posgrado (creación, reestructuración y vigente), nivel en el PNPB, fecha de obtención en el PNPB, grado Académico (Posgrado, Maestría o Especialidad) y fecha de registro ante profesiones.
- **Ingreso.**
 - Selección de los criterios de evaluación por posgrado, el coordinador seleccionará los criterios de evaluación al posgrado que se les aplicará a los aspirantes.
 - Consulta alta y edición para la asignación de tutores a los nuevos estudiantes, el coordinador podrá conocer cuándo es el ingreso de un alumno al posgrado, asignarle un tutor, y el tutor podrá ir registrando las reuniones de tutorías para irle dando seguimiento a los alumnos.
- **Permanencia – Egreso.**
 - Consulta de seguimiento a materias, reuniones de tutores y movilidad de los alumnos, donde los alumnos y tutores podrán consultar y registrar información de las reuniones de tutoría.
 - Consulta, alta y edición de seguimiento al protocolo, la dirección (seguimiento), y revisión de la tesis. Para este proceso, el coordinador del posgrado será responsable de validar el protocolo y asignar los directores y revisores de la tesis; una vez autorizado el protocolo el sistema deberá permitir cargar los avances de la tesis y



los comentarios por el director y el revisor así como el porcentaje de avance en la elaboración de la tesis.

- **Infraestructura.**

- Consulta, alta y edición de la cantidad de aulas, cubículos, laboratorios equipo de cómputo, bibliotecas físicas y virtuales que cuenta cada posgrado, el coordinador del posgrado será responsable de capturar la cantidad de cada rubro y una evidencia en formato PDF de todo el detalle del inventario.

- **Resultados y vinculación.**

- Consulta, alta y edición de los proyectos desarrollados, contratos y convenios de los posgrados, el coordinador y director de posgrado podrán consultar los proyectos desarrollados por investigadores y estudiantes de la Universidad.

- **Investigación (CVU).**

- Consulta, alta y edición de libros publicados, artículos, tesis, docencia, distinciones, investigación y desarrollo tecnológico, por alumno o profesor; cada profesor y alumno podrá cargar su información y el coordinador o director de posgrado deberá poder consultarla.

- **Seguridad y administración.**

- Manejo de privilegios para los usuarios y roles del sistema. Este deberá permitir administrar los mismos, los usuarios se podrán auto-registrar siempre y cuando cuenten con un correo institucional. El usuario se definirá de la siguiente manera: para los profesores deberá ser su clave de empleado y para los alumnos su matrícula.



3.2 Requerimientos generales y particulares del usuario.

El Instituto de Ingeniería Eléctrica y Electrónica (IEEE, por sus siglas en inglés) define un requerimiento como¹: Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes del sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Los requerimientos pueden dividirse en funcionales y no funcionales.

Los requerimientos funcionales definen las tareas que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

En el caso del Sistema de Productividad del Posgrado de una Universidad se identificaron los siguientes requerimientos:

¹ IEEE (2010). Standards Glossary.
https://www.ieee.org/education_careers/education/standards/standards_glossary.html



- Dividir la aplicación en módulos bien delimitados.
 - Estructura de posgrado
 - Investigación
 - Resultados y vinculación
 - Permanencia de estudiantes
 - Ingreso
 - Infraestructura
- Cada módulo podrá trabajar de manera independiente pero la información debe ser compartida con el resto de módulos, en caso de ser necesario.
- Cada usuario tendrá un perfil que le brindará permisos para la consulta o modificación de información.

Estructura de posgrado.

- Se debe permitir mostrar todos los posgrados existentes registrados por la dirección de posgrados.
- Se mostrará una pantalla para los datos generales del posgrado con un formulario de captura, que permita agregar o editar los datos contenidos en estos.
- Se mostrará una pantalla para los datos de la historia del posgrado con un formulario de captura que permita agregar o editar los datos contenidos en estos.
- Se mostrará una pantalla para los datos del desarrollo del posgrado con un formulario de captura que permita agregar o editar los datos contenidos en estos.
- Se mostrará una pantalla para los datos que estructuran el plan de estudios del posgrado, con un conjunto de pestañas las cuales contendrán los formularios de captura, estos permitirán agregar o editar los datos contenidos en estos.



- Se mostrará una pantalla para cargar la planta académica en la que se mostrará una lista de los profesores que estén dados de alta en la universidad.
- Se imprimirá un reporte completo de la estructura del posgrado en formato PDF.

Investigación.

- El sistema permitirá el registro y control de diversos proyectos de investigación de la universidad, categorizados en distintos rubros:
 - Libros.
 - Investigación y consultoría.
 - Redes y grupos de investigación.
 - Estancias de investigación.
 - Capítulos de libros.
 - Artículos publicados.
 - Distinciones.
 - Desarrollo tecnológico.
 - Docencia.
 - Tesis.
 - Verano de investigación.
- El sistema permitirá la consulta en forma de listado de cada categoría.
- Cuando el usuario lo solicite, el usuario podrá mostrar el detalle de cada registro contenido en el listado general.
- El sistema dispondrá de reportes por categoría.

Resultados y vinculación.

- El sistema permitirá la consulta de la productividad de cada dirección de posgrado. Se podrá generar un reporte con la información mencionada.



CAPÍTULO III



- Se mostrará una pantalla para consultar la información del total de proyectos que están dados de alta en el posgrado. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para consultar la información del total de convenios que están dados de alta en el posgrado. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para consultar la información del total de contratos que están dados de alta en el posgrado. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para consultar la información del total de proyectos que están dados de alta en los posgrados. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para agregar comentarios y porcentajes de ponderación para cada uno de los proyectos de los posgrados.
- Se mostrará una pantalla para consultar la información del total de convenios que están dados de alta en los posgrados. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para agregar comentarios y porcentajes de ponderación para cada uno de los convenios de los posgrados.
- Se mostrará una pantalla para consultar la información del total de contratos que están dados de alta en los posgrados. Esta información podrá ser extraída en forma de reporte.
- Se permitirá mostrar la trascendencia, cobertura y evolución existente por la dirección de posgrados.
- Se mostrará una pantalla para consultar la información del alcance del posgrado que está dada de alta en la universidad. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para consultar la información del índice estudiante/profesor del posgrado que está dada de alta en la universidad. Esta información podrá ser extraída en forma de reporte.



- Se mostrará una pantalla para consultar la información del total de artículos de revistas que están dados de alta en la universidad. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para consultar la información del total de libros que están dados de alta en la universidad. Esta información podrá ser extraída en forma de reporte.
- Se mostrará una pantalla para consultar la información del total de capítulos de libros que están dados de alta en la universidad. Esta información podrá ser extraída en forma de reporte.

Permanencia de estudiantes.

- Se manejará un usuario “tutor” que dará seguimiento a la permanencia de los estudiantes dentro del posgrado.
- El tutor podrá consultar las materias de cada alumno de manera individual.
- El tutor podrá consultar los avances de tesis de cada alumno.
- El tutor podrá consultar los eventos académicos de los alumnos (movilidad).
- El tutor podrá consultar todas las reuniones de tutorías existentes y se mostrarán opciones adicionales para la administración de las mismas.
- El tutor podrá agregar o editar reuniones con cada alumno.
- El sistema podrá registrar avances en tesis de cada alumno. Cuando se registre un avance en una tesis, el sistema notificará vía correo al alumno de dicho cambio. Cuando el avance registrado sea del 100%, el sistema enviará al alumno un mensaje vía correo con la terminación del tema. Si el avance registrado sea menor al 100%, el sistema permitirá que el alumno agregue nueva información para ser evaluada por el asesor.
- El sistema aceptará archivos PDF que contengan la información relativa a las tesis y sus avances de cada alumno. Cuando un alumno adjunte avances de su tesis, el sistema notificará vía correo al asesor correspondiente.



- Cada avance de tesis ingresado en el sistema podrá ser aceptado o rechazado por el asesor. Cuando el asesor acepte o rechace un avance, el sistema notificará vía correo al alumno responsable.
- Cada tesis registrada en el sistema podrá ser liberada cuando se considere terminada. El sistema permitirá la liberación de tesis por un revisor que considere que el desarrollo esté completo.
- Cuando una tesis se libera el sistema notificará al alumno y solicitará una fecha de defensa de la tesis. No se contempla una comunicación con un sistema externo para la validación de la defensa de la tesis.
- El sistema permitirá consultar un listado de los alumnos de cada posgrado.

Ingreso.

- Se mostrará una pantalla para capturar los datos de la convocatoria. Se permitirá adjuntar archivos que respalden la información ingresada
- El sistema ejecutará tareas calendarizadas para la actualización de listados de alumnos inscritos a los posgrados.
- Se mostrará una pantalla donde se listarán por coordinación todos los aspirantes que pagaron trámite para ingreso al posgrado.
- Se mostrará una pantalla para la documentación del proceso de admisión con un formulario de captura que permita agregar o editar los datos contenidos en estos.
- El sistema será capaz de evaluar a un aspirante y ponderar si cumple con las condiciones de ingreso al posgrado para ser aceptado o rechazado.
- El sistema deberá enviar notificaciones a los actores cuando un aspirante es aceptado en el posgrado.
- El sistema será capaz de enviar información relevante a los alumnos, vía correo electrónico, sobre convocatorias de becas otorgadas por el CONACyT.



- Se mostrará una pantalla para la captura de información adicional requerida por la dirección de posgrados.
- El sistema deberá comunicarse con otros sistemas de control de pagos para validar que el aspirante ha cubierto la cuota correspondiente a la inscripción.
- Se mostrará el módulo de Convocatoria, el cual permitirá cargar la documentación en PDF y datos relevantes como fecha aplicación de los exámenes o entrevistas.
- Se mostrará un formulario donde se capturarán los datos correspondientes a los criterios de evaluación y ponderación.

Infraestructura.

- Se debe permitir mostrar la infraestructura existente por la dirección de posgrados.
- Se mostrará una pantalla con un formulario de captura para agregar los datos de inventario de cubículos.
- Se mostrará una pantalla con un formulario de captura para agregar los datos de inventario de aulas.
- Se mostrará una pantalla con un formulario de captura para agregar los datos de inventario de laboratorios.
- Se mostrará una pantalla con un formulario de captura para agregar los datos de inventario de equipo de cómputo para profesores y alumnos.
- Se mostrará una pantalla con un formulario de captura para agregar los datos de inventario de bibliotecas.
- Se mostrará una pantalla con un formulario de captura para agregar los datos de inventario de biblioteca virtual.
- Se mostrará una pantalla para consultar la información del inventario de cubículos dados de alta en el posgrado.



- Se mostrará una pantalla para consultar la información del inventario de aulas dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de laboratorios dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario del equipo de cómputo para profesores y alumnos dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de bibliotecas dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de biblioteca virtual dados de alta en el posgrado.

3.3 Prototipos del Sistema.

El prototipo del sistema de productividad del posgrado será a través de un entorno web visible al usuario, el cual será diseñado para un fácil manejo e interacción. Desde aquí este usuario podrá realizar funciones como son: registrar toda la información de los posgrados, planes de estudios, historia, infraestructura, seguimiento a estudiantes y así como los reportes de los resultados del posgrado, adicionalmente todos los académicos podrán cargar todo su currículum vitae, que permitirá alimentar los reportes de la productividad académica de cada uno de los profesores o planta académica de cada uno de los posgrado.



Así pues, la interfaz deberá contener las entradas principales y funcionalidad pertinente que serán importantes para el sistema de productividad académica.

A continuación se muestra el diseño a nivel de bloques de la interfaz del usuario, vea figura 3.3.1:

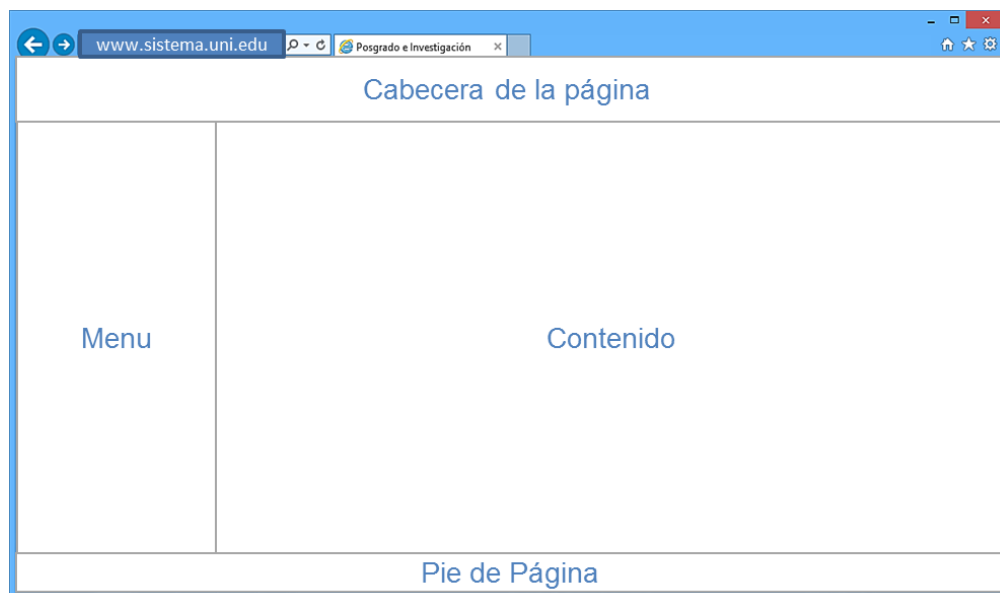


Figura 3.3.1 Diseño de bloques de la interface de usuario

En la parte superior se encuentra el bloque de cabecera de la página. El bloque de cabecera de la página (representado así en la figura 3.3.2), será el encabezado de la página, en él se colocará el logotipo, el nombre del sistema, nombre del módulo consultado y el nombre del usuario que esta autenticado en el sistema.

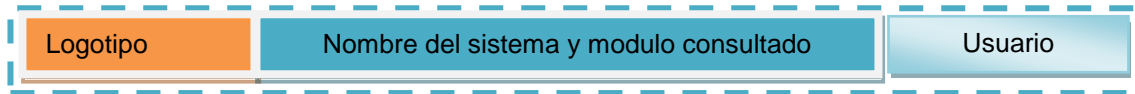


Figura 3.3.2 Bloques de cabecera de página

El bloque Menú es una barra lateral que se encontrará en la parte izquierda de la página. En el bloque Menú se colocarán diversos botones con acciones del sistema entre ellas se visualizara el menú Estructura de Posgrado, Permanencia, Infraestructura, Resultados y vinculación, y CVU. Ver figura 3.3.3.



Figura 3.3.3 Bloques de Menú



El bloque de pie de página es una barra horizontal que se encontrará en la parte inferior de la página. En el bloque de pie de página se colocarán domicilio de la empresa y copyright, vea figura 3.3.4.



Figura 3.3.4 Bloques de pie de página

El bloque de contenido principal se encuentra en el centro de la página, este estará formado por tres secciones principales, en la sección del buscador que contendrá al menos 3 criterios de búsqueda, en la sección de operaciones que contendrá las operaciones de consulta detalle, actualización, eliminación y alta de registros y la sección que contendrá la consulta listado. En el bloque de contenido principal se mostrarán los diferentes operaciones del sistema, vea figura 3.3.5.

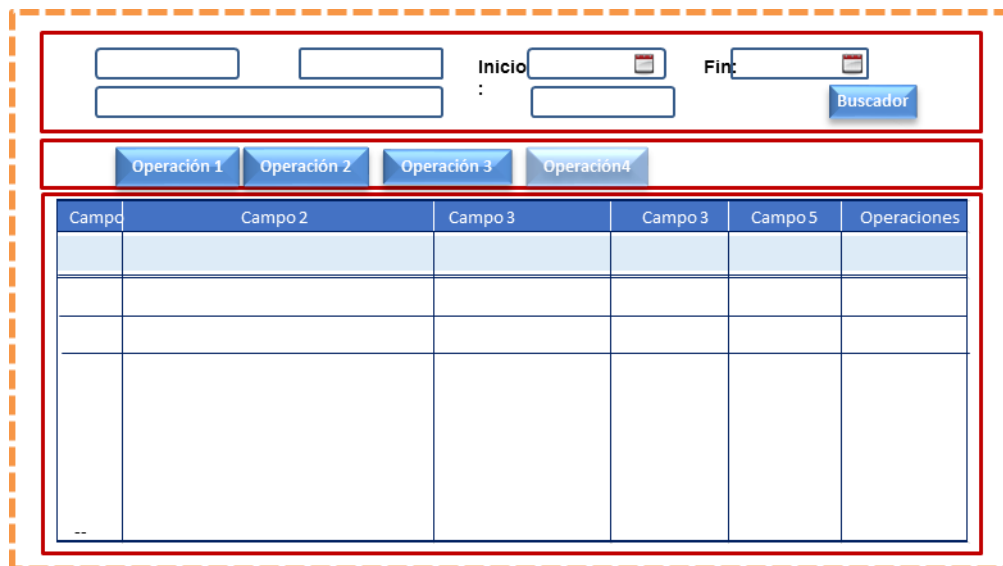


Figura 3.3.5 Bloques de contenido principal



El bloque de contenido de formularios contendrá los diferentes formularios de consulta detalle, alta, edición y eliminación de registros. En el bloque de contenido de formularios se mostrarán los campos del formulario del sistema, vea figura 3.3.6.

The image shows a web form with 17 fields, labeled Campo1 through Campo17. The fields are arranged in a grid-like structure. Campo1 is a radio button. Campo2 is a radio button with a green dot. Campo3, Campo4, and Campo5 are text input fields. Campo6 and Campo7 are text input fields. Campo8 and Campo9 are text input fields. Campo10 is a radio button with a green dot. Campo11 is a radio button with a green dot. Campo12 is a radio button with a green dot. Campo13 is a text input field with a calendar icon. Campo14 is a text input field. Campo15 is a text input field. Campo16 is a text input field. Campo17 is a text input field. A blue 'Guardar' button is located at the bottom right of the form.

Figura 3.3.6 Bloques de contenido principal

3.4 Arquitectura del sistema web.

La estructura de la información de un sistema Web es una pieza fundamental para su utilización ya que influirá directamente en la forma en que los usuarios acceden a los contenidos.



Mencionaremos cuatro tipos de estructuras de información.

Estructura jerárquica.

Las estructuras jerárquicas son aquellas que agrupan información en subsecciones que están en conexión directa a una sección “padre”. A su vez, estas subsecciones pueden contener otras.

Este tipo de estructura es fácilmente comprensible por los usuarios, ya que las estructuras jerárquicas se utilizan de una manera amplia en la vida cotidiana, lo que permite a los usuarios que generen una estructura mental del sitio donde se encuentran y de su posición actual dentro de la estructura del sistema.

Debido a las características de estas estructuras, los menús de navegación que mejor se adaptan son los sistemas de pestañas de varios niveles. Los menús desplegables también son ampliamente utilizados en este tipo de estructuras: menús desplegables de varios niveles, listas expandibles y finalmente los sistemas de menús pop-up interactivos.

Todos ellos respetan una de las características principales de las estructuras jerárquicas: presentan en un primer nivel las áreas de contenido más generales y relevantes y esconden las subdivisiones de estas áreas principales, que se irán presentando al usuario en sucesivas subdivisiones bajo petición de éste.



Estructura de hipertexto.

Este tipo de estructura consiste en organizar la información de una manera no lineal, es decir los bloques de información pueden estar enlazados utilizando cualquier tipo de asociación, ya sea jerárquica o no.

Por consiguiente, estos sistemas proporcionan gran flexibilidad ya que permiten establecer relaciones entre cualquiera de los elementos y áreas de información.

Si el modelo jerárquico es un sistema fácilmente entendible por los usuarios, ahora nos encontramos en el lado opuesto ya que la estructura de hipertexto plantea una complejidad conceptual que pueden confundir al usuario. El usuario puede sentirse perdido al no ser capaz de generar un modelo mental de la organización del sitio y al no poder encontrar la información que realmente le interesa.

Por este motivo, estas estructuras de la información suelen utilizarse como complemento a otro tipo de estructuras.

Los sistemas de navegación que mejor implementan este tipo de estructura son los sistemas de navegación ad hoc, que relacionan palabras, expresiones o frases con contenidos que amplían la información proporcionada por el autor.

Estas palabras, expresiones o frases suelen ser vínculos incrustados en el propio texto y por consiguiente no podemos tratarlos como menús de navegación.



El único sistema de menús de navegación que de alguna manera se podría ajustar a este tipo de estructura de la información serían las listas de opciones o las barras de imágenes. Estas podrían ser diferentes en cada uno de los bloques de información y, de esta manera, cada nodo de información tendría sus propios enlaces personalizados.

Estructura secuencial o lineal.

Es la forma más simple de organizar la información. La información que fluye de forma natural como una narrativa lineal o con un orden lógico es perfecta para este tipo de estructura.

La estructura secuencial puede ser cronológica, una serie de temas ordenados lógicamente de lo más general a lo más específico o incluso ordenado alfabéticamente.

Sin embargo, este tipo de organización sólo suele funcionar para sitios pequeños; a medida que la narrativa se torna más compleja, necesita una mayor estructuración para seguir siendo comprensible. La figura 3.4.1 es un ejemplo simple de esta estructura.

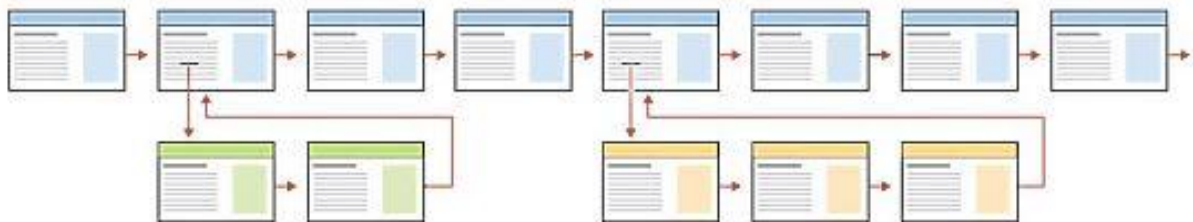


Fig. 3.4.1 Estructura secuencial



Estructura en malla.

Muchos manuales de procedimientos, listados de cursos universitarios, descripciones médicas de casos se adaptan perfectamente a este tipo de estructura.

Las mallas son una buena forma de asociar variables, en una serie de categorías estandarizadas como eventos, tecnología, cultura, etc.

Para tener éxito con este tipo de estructura es necesario que las unidades individuales de una malla, compartan una estructura de temas y subtemas absolutamente uniforme.

Sin embargo las mallas pueden ser difíciles de comprender si el usuario no consigue establecer la relación entre las diferentes categorías de información, y quizá más apropiadas para audiencias experimentadas que ya poseen un conocimiento básico del tema y de su organización.

Arquitectura del Sistema de Productividad del Posgrado.

Se trabajará bajo una estructura bien definida que permita identificar rápidamente los componentes de software implicados con el objetivo de simplificar el desarrollo y mantenimiento de las aplicaciones (figura 3.4.2). Para lograrlo es necesario adoptar una arquitectura de referencia que homologue el diseño de las aplicaciones y la forma en la que se estructuran sus componentes.

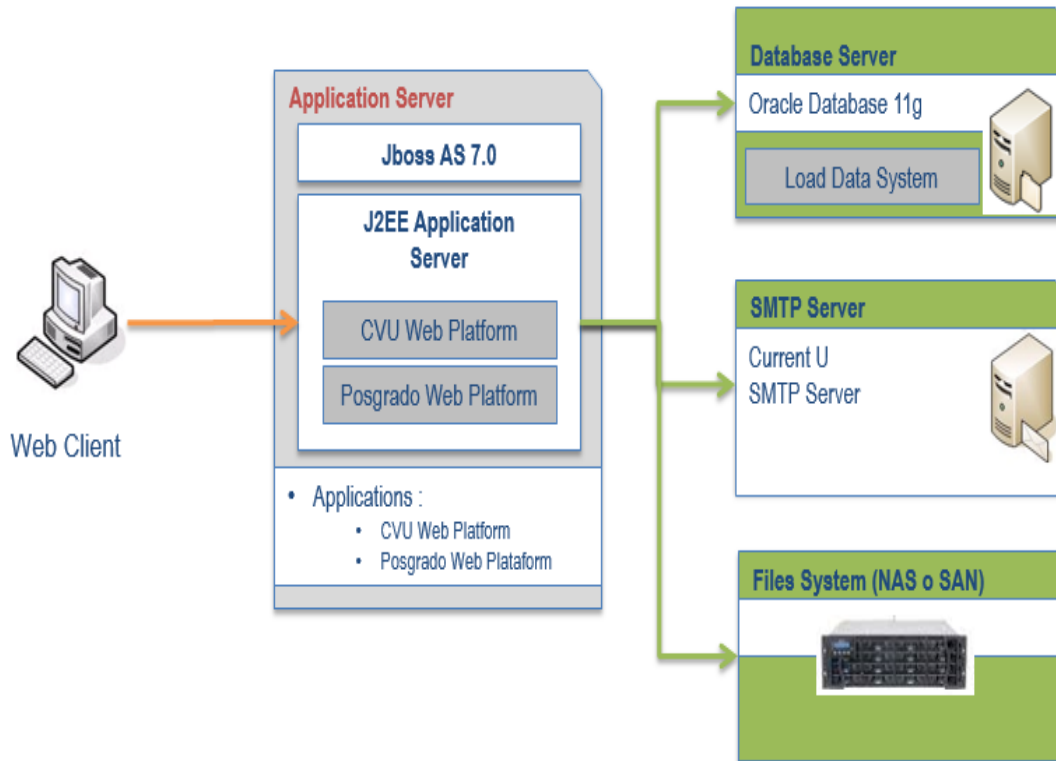


Figura 3.4.2 Capas de la arquitectura física

La arquitectura de referencia para el desarrollo de aplicaciones web, establece que los componentes deben separarse de acuerdo al siguiente modelo lógico (figura 3.4.3).

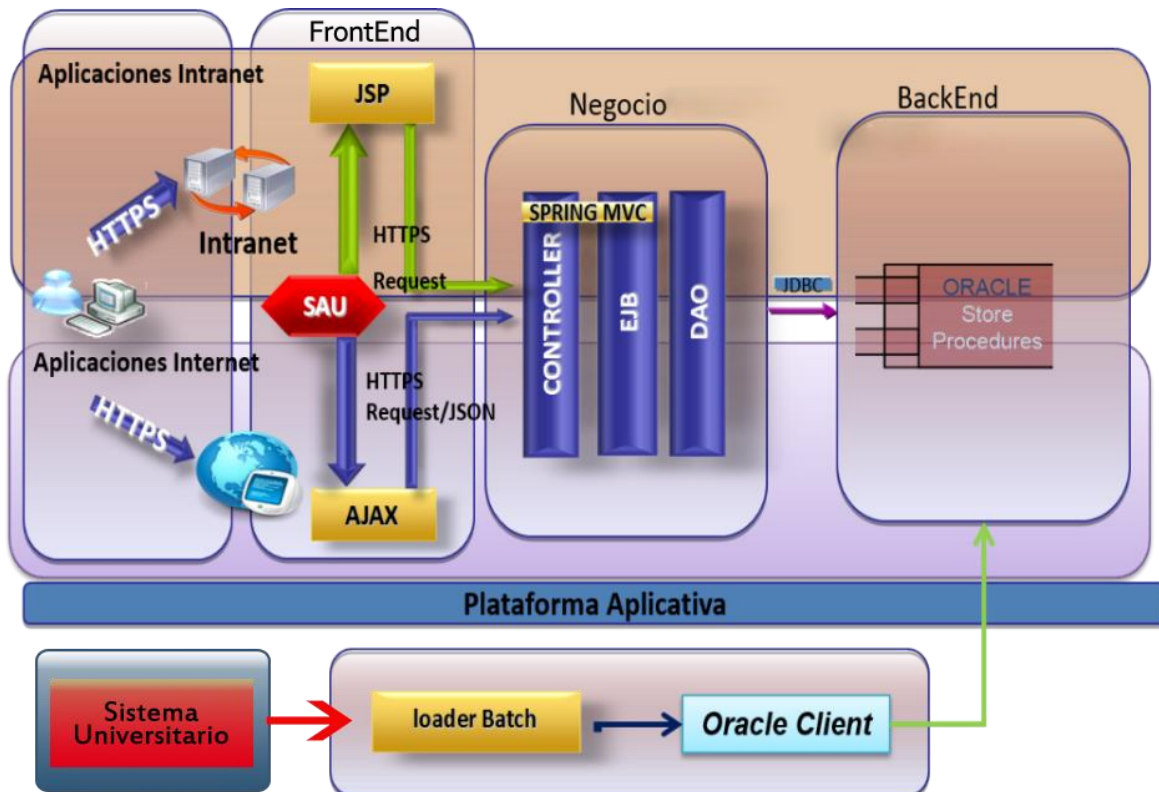


Figura 3.4.3 Capas de la arquitectura lógica

Capa cliente.

Se encarga de presentar y recopilar la información del usuario, se ejecuta en un navegador web mediante páginas HTML o Applets de Java.

Capa de presentación.

La función de la capa de presentación es recibir las peticiones provenientes de la capa cliente, solicitar el procesamiento de información a la capa de negocio y generar las respuestas correspondientes al usuario.



La capa de presentación se desarrollará a través de un módulo web, el cual representará a una aplicación de este tipo. Un módulo web se creará ensamblando servlets, archivos JSP (JavaServer Pages), y contenido estático como, por ejemplo, páginas HTML en una sola unidad desplegable. Los módulos Web se almacenarán en archivos WAR, que son ficheros archivadores Java estándar.

Las interfaces de usuario deberán diseñarse en base a la maqueta definida en el proyecto.

Cada módulo web deberá contener un descriptor denominado “web.xml”, que declarará el contenido del módulo. Este fichero contendrá información acerca de la estructura y las dependencias externas de los componentes Web del módulo y describirá como se tienen que utilizar los componentes durante la ejecución.

Se debe implementar el patrón MVC en esta capa, para ello se empleará el componente MVC de Spring Framework.

El controlador será el único componente que podrá mantener estado, es decir tener memoria a través de la sesión.



Capa de negocio.

Dentro de esta capa se realizará la implementación de los procesos que soportan al negocio. Se conformará de los siguientes componentes:

- Data Transfer Object (DTO)
- Business Objects
- Data Access Objects (DAO's)
- Capa de Integración
- Capa de Back-End

Características de estilo del sitio web².

- La navegación estará orientada a las necesidades de información y servicios de la institución.
- Se hará uso de un lenguaje simple y claro del contexto del contenido que permita al usuario la comprensión del mismo.
- Existirá estructura gramatical, ortografía y redacción correctas.
- El sitio dispondrá de una página principal (home page) de la cual se desprenderá un conjunto de páginas organizadas y aplicaciones interactivas.
- Contará con diversos enlaces que permitan el acceso a diversas partes del sitio.
- El contraste utilizado entre letra/tamaño/color y fondo permitirá una lectura de manera sencilla.
- Se tendrá una distribución clara y consistente que permitirá a los usuarios la identificación de las áreas que componen el sitio.
- Se logrará un equilibrio armónico y estético de los elementos en el espacio visual del sitio.

² Lynch, P. (2009). Web Style Guide: Basic Design Principles for Creating Web Sites.



DESCRIPCIÓN DEL SISTEMA WEB



- El sitio contará con un logotipo del programa universitario al que pertenece, así como el nombre (El logotipo es un elemento clave que nos ayuda a identificar la identidad de una institución, empresa u organización).
- Las imágenes utilizadas contarán con formatos ligeros en los que no pierden calidad.
- Los colores utilizados identificarán a la entidad.
- Se permitirán actualizaciones y mantenimiento de los contenidos.



CAPÍTULO IV

DISEÑO Y CONSTRUCCIÓN DEL SISTEMA WEB

4.1 Modelo esencial ambiental del sistema web.

La construcción de un modelo ambiental¹ de cualquier sistema web es uno de los primeros y más importantes pasos en la construcción del modelo con los requerimientos de un usuario.

Al desarrollar cualquier sistema se deben definir las interfaces entre el sistema y el universo, es decir el ambiente. Es necesario saber el tipo de datos que ingresarán desde el ambiente exterior y el tipo de información que saldrá al ambiente externo. Dentro del modelo ambiental se trata de identificar los acontecimientos que ocurren dentro de este, para los cuales debe ser capaz de responder el sistema, es decir se producen salidas como respuestas a algún acontecimiento o estímulo en el ambiente. Los sistemas que se construyen son racionales y tienen propósitos específicos, estos están definidos por los usuarios, y siempre se deben dejar áreas abiertas o dominio de cambios en las que se puedan integrar los aspectos de los que no está seguro el usuario o que no ha tomado en cuenta.

¹ Yourdon, E. (1994). Análisis Estructurado Moderno.



Se requiere invertir una gran parte de tiempo a este modelo, se aplican una serie de pasos iterativos, con datos que se van añadiendo y se renuevan, suele ser uno de los puntos que llegan a trabajar directivos, usuarios y administradores. Entre sus componentes encontramos el diagrama de contexto que a su vez contiene el flujo de control y datos, almacenamiento y una representación de todo el sistema, los casos de uso son una secuencia de interacciones entre un sistema y un usuario que utiliza sus servicios, así como el diagrama de entidad relación que consiste en un conjunto de objetos básicos llamados entidades y sus relaciones entre objetos, también es necesario definir el diccionario de datos que contendrá las características de los sitios donde se almacenan los datos del sistema incluyendo nombre, descripción, contenido y organización, así mismo se requiere del diagrama de clases que describirá a través de un diagrama la estructura estática que muestran las clases del sistema y sus interrelaciones.

4.1.1 Diagrama de contexto.

El diagrama de contexto es el nivel cero de un diagrama de flujo de datos, en donde una sola sección o burbuja puede representar todo el sistema, muestra a través del flujo de datos las interacciones existentes entre agentes externos y el sistema, se representa como un único proceso de entradas y salidas.

En este tipo de diagramas no se representan todos los flujos de datos del sistema, se representa una visión general, siguiendo:

- Una representación de los flujos de datos que tienen que ver con el objetivo principal del sistema.
- El uso de flujos de datos compuestos que representen características similares.



Figura 4.1.1.1 Diagrama de contexto.

En la Figura 4.1.1.1 se muestran las características del diagrama de contexto del Sistema de Productividad del Posgrado de la Universidad:

- Los terminadores, que se encuentran en la parte externa son las personas, organizaciones y sistemas con los que se comunica el sistema.
- Se encuentran representados los datos que el sistema recibe y que son procesados de alguna manera (Director, Coordinador, Investigación, Sistema de administración escolar).



- Los datos que son y serán producidos por el sistema y que se envían al exterior (Director, Coordinador, Investigación, Permanencia y egreso, Resultados y vinculación).
- El almacén de datos que el sistema comparte con los terminadores.
- La frontera entre el sistema y el resto del mundo.

4.1.2 Casos de uso.

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

En el caso del Sistema de Productividad del Posgrado de una Universidad, se identificaron los siguientes casos de uso:

Identificador	CVU-1.0
Nombre	CONSULTAR CVU



Actores: Estudiante/Profesor/Dirección de posgrado.

Descripción: Permite al usuario consultar la información del currículum vitae contenida dentro del sistema y a la cual tiene acceso de acuerdo a los privilegios otorgados. La información disponible en el sistema es: datos generales de los usuarios, libros, consultorías, redes y grupos de investigación, estancias, artículos, desarrollos tecnológicos y tesis.

Precondiciones: N/A

Postcondiciones: Se despliega la información solicitada por el usuario.

Flujo normal:

- El usuario ingresa al sistema (CVU-2.0). Si es un usuario válido el sistema muestra los menús con las opciones de consulta disponibles. Si el usuario no introduce credenciales válidas se mostrará un mensaje de error y el caso de uso termina. Si el usuario desea darse de alta se ejecuta el caso de uso CVU-2.2 CREAR USUARIO.
- El usuario elige una de las opciones a consultar: datos generales de los usuarios, libros, consultorías, redes y grupos de trabajo, estancias, artículos, distinciones, desarrollos tecnológicos, docencias, tesis. Si el usuario es administrador podrá elegir buscar usuarios por diferentes criterios para su consulta.
- Al seleccionar una opción de consulta, el sistema desplegará la información a la cual tenga permiso el usuario.
- Una vez desplegada la información, se le brindará al usuario la opción para generar el informe correspondiente. La generación del dicho informe consiste en la corrida de un archivo JASPER que a su vez exportará la información en un archivo PDF.



- Igualmente, el sistema dispondrá de una opción de modificación si es que el usuario tiene los privilegios que lo faculten para realizar la mencionada modificación. El sistema ejecutará el caso de uso CVU 1.1- MODIFICAR CVU.

Flujos alternos:

- Si el usuario no ingresa credenciales válidas el sistema no muestra ninguna información y termina el caso de uso.

Reglas de negocio:

- Los directores de posgrado tendrán acceso a toda la información contenida en base de datos.
- Los profesores tendrán acceso a su información más la relativa a los libros, artículos, desarrollos tecnológicos y a los datos generales de estudiantes que tengan bajo su docencia, dirección o redes y grupos de trabajo.
- Los estudiantes tendrán acceso a su información más la relativa a libros, artículos y distinciones cargados en sistema.

Notas: En la figura 4.1.2.1 se muestra un diagrama de los pasos a seguir en este caso de uso.

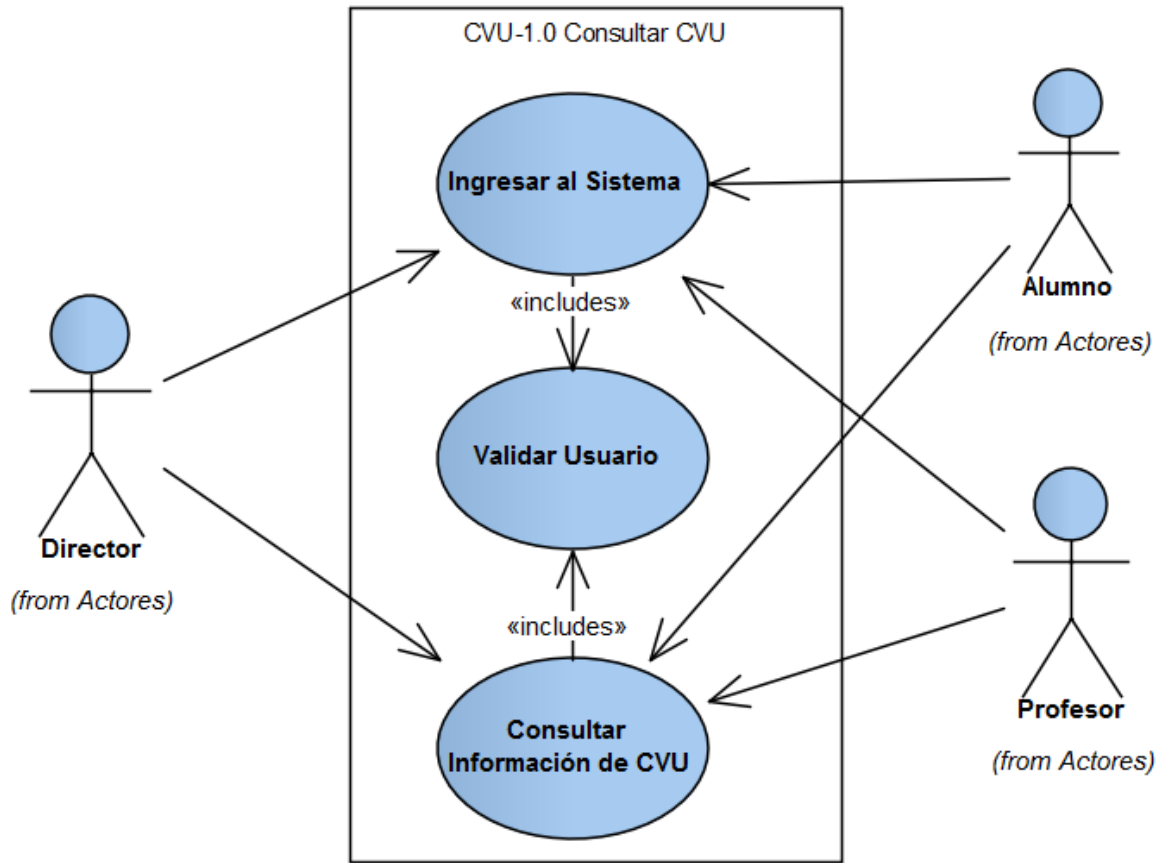


Fig. 4.1.2.1 Diagrama caso de uso CVU-1.0



Identificador	CVU-2.0
Nombre	AUTENTIFICAR USUARIO

Actores: Estudiante/Profesor/Dirección de posgrado.

Descripción: Permite al usuario identificarse ante el sistema para poder otorgarle permisos sobre la información almacenada en base de datos.

Precondiciones: El usuario ha sido cargado previamente en el sistema.

Postcondiciones: Carga los privilegios del usuario en sesión.

Flujo normal:

- El usuario ingresa su identificador y una contraseña.
- El sistema ejecuta el caso de uso CVU-2.1 VALIDAR USUARIO para determinar si la información ingresada pertenece o no a un usuario válido.

Flujos alternos:

- Si el usuario no ingresa un identificador preexistente en BD o si la contraseña ingresada no concuerda con la guardada en base el sistema negará el acceso al usuario y mostrará un mensaje de notificación sin especificar la razón de la declinación.
- El sistema dispondrá de una opción para “Alta de usuarios” en donde podrán registrarse nuevos identificadores de usuario. El sistema ejecuta el caso de uso CVU-2.2 CREAR USUARIO.
- Si es el primer acceso del usuario al sistema, se le pedirá que cambie su contraseña para poder continuar con su acceso.

Reglas de negocio:

- El sistema bloqueará una cuenta si se tuvieron 3 intentos fallidos de ingreso.
- Todas las contraseñas tendrán una validez de hasta 12 meses, después de concluido ese periodo si no ha sido actualizada la contraseña la cuenta será bloqueada por sistema.

Notas: En la figura 4.1.2.2 se muestra un diagrama de los pasos a seguir en este caso de uso.

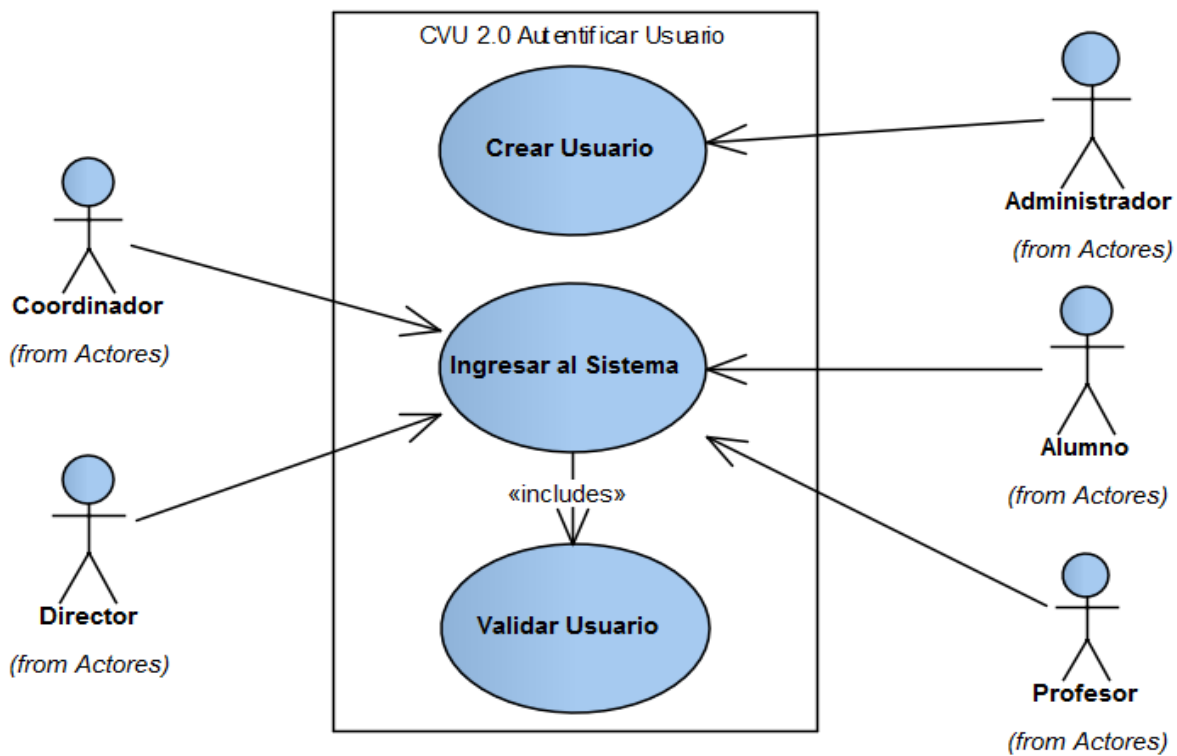


Fig. 4.1.2.2 Diagrama de caso de uso CVU-2.0



Identificador	CVU-2.1
Nombre	VALIDAR USUARIO

Actores: Estudiante/Profesor/Dirección de posgrado.

Descripción: El sistema determina si una dupla de identificador-contraseña pertenecen a un usuario válido o no.

Precondiciones: El usuario ha sido cargado previamente en el sistema.

Postcondiciones: Determina si la credenciales ingresadas son válidas o no.

Flujo normal:

- El sistema busca en base de datos la existencia del identificador ingresado. Si existe procede a validar la contraseña ingresada contra la guardada en BD. Si las contraseñas concuerdan el sistema marca el usuario como válido.

Flujos alternos:

- Si el identificador no existe en base de datos indica que no es válido y termina el caso de uso.
- Si la contraseña ingresada no concuerda con la guardada en base indica que no es un usuario válido y termina el caso de uso.

Reglas de negocio:

- Las contraseñas serán almacenadas ya encriptadas para mayor seguridad de las mismas.

Notas: En la figura 4.1.2.3 se muestra un diagrama de los pasos a seguir en este caso de uso.

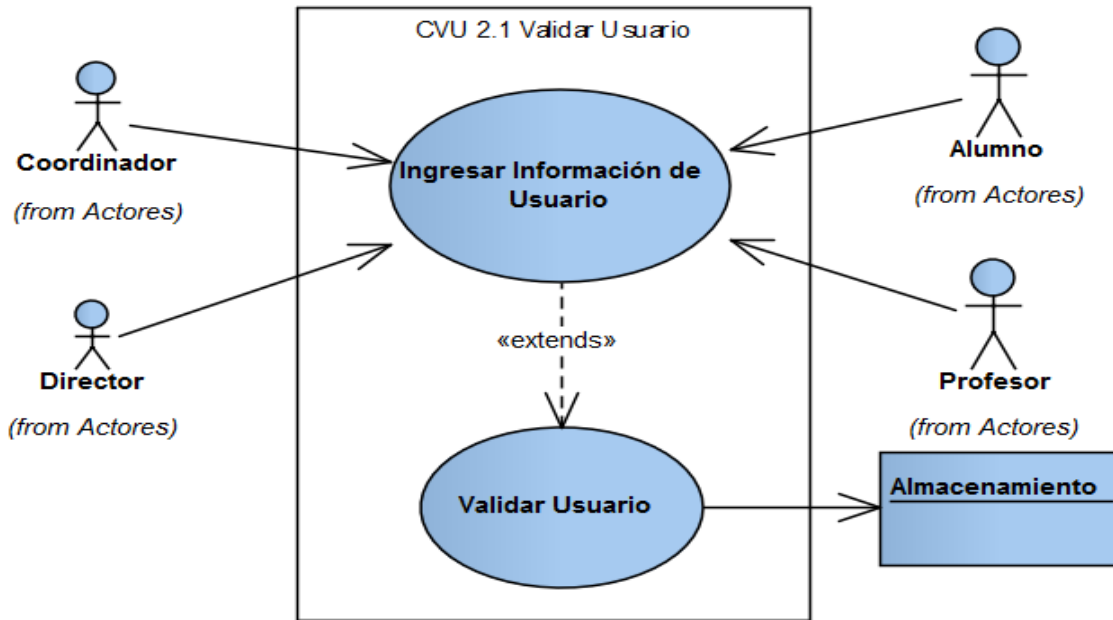


Fig. 4.1.2.3 Diagrama de caso de uso CVU-2.1

Identificador	CVU-2.2
Nombre	CREAR USUARIO

Actores: Estudiante/Profesor/Dirección de posgrado.

Descripción: El sistema ingresará la información de un nuevo usuario a la base de datos.

Precondiciones: El usuario no ha sido cargado al sistema.

Postcondiciones: La información del nuevo usuario es grabada en la base de datos.



Flujo normal:

- El sistema pedirá los datos del nuevo usuario:
 - Tipo.
 - Número de empleado (profesores y directores) o matrícula (estudiantes).
 - CURP.
 - Grado académico.
 - Correo electrónico.
- El sistema verifica que el número de empleado o matrícula del alumno no estén guardados previamente en el sistema. Si el usuario no se ha cargado previamente, el sistema genera un identificador y una contraseña para el nuevo usuario.
- El sistema envía por correo la información recién generada.
- Se despliega un mensaje con la notificación del alta exitosa.

Flujos alternos:

- Si la información del usuario ya existe en base de datos previamente, se negará la alta del usuario y se mostrará un mensaje de error con una notificación de rechazo.

Reglas de negocio:

- Los CURP no pueden repetirse para diferentes usuarios.
- Los números de empleado no pueden repetirse para diferentes usuarios.
- Los números de matrícula no pueden repetirse para diferentes usuarios.
- Los correos electrónicos no pueden repetirse para diferentes usuarios.

Notas: En la figura 4.1.2.4 se muestra un diagrama de los pasos a seguir en este caso de uso.

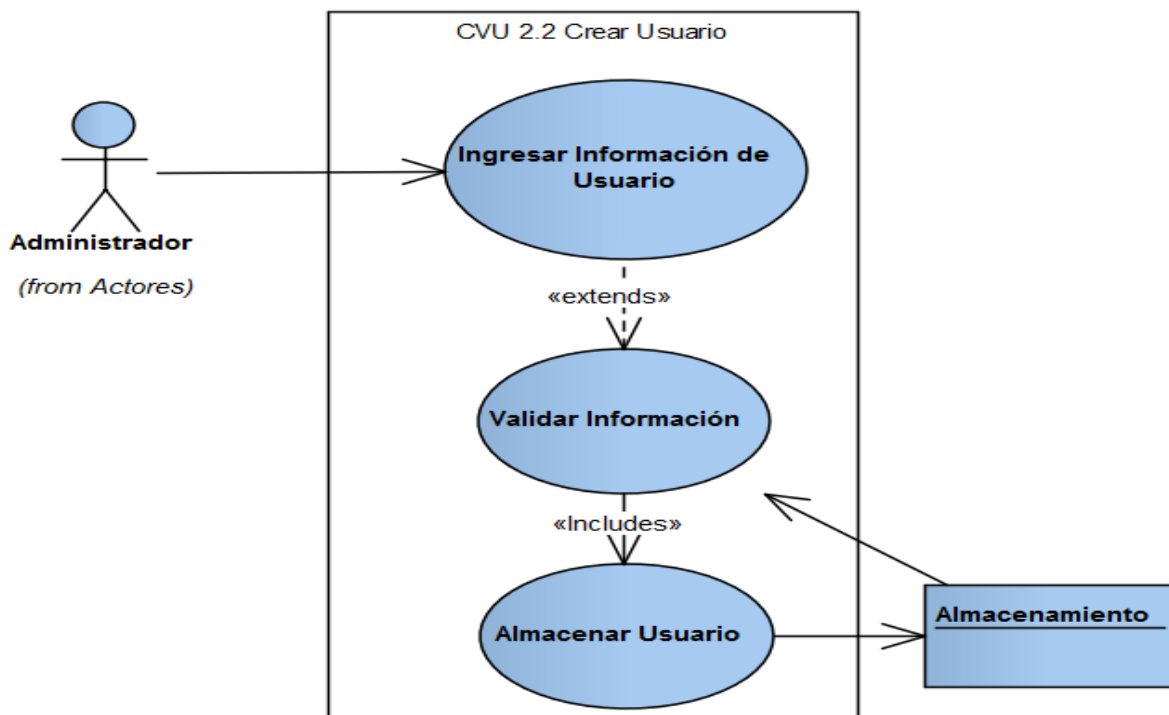


Fig. 4.1.2.4 Diagrama de caso de uso CVU-2.2

Identificador	CVU-1.1
Nombre	MODIFICAR CVU

Actores: Estudiante/Profesor/Dirección de posgrado.

Descripción: A través de esta opción el usuario podrá modificar su información del CVU.

Precondiciones: La información ha sido capturada en base de datos y el usuario ha seleccionado el registro que desea modificar.



Postcondiciones: El registro se guarda en base de datos con la nueva información ingresada.

Flujo normal:

- Una vez que el usuario ha seleccionado modificar su CVU, el sistema consulta la base de datos para extraer todos los detalles que pudieran no haberse cargado con anterioridad y muestra el formulario para la captura de la los datos de su CVU, datos generales del usuario, libros, consultorías, redes y grupos de investigación, artículos, libros y tesis.
- Cuando el usuario acciona el botón de “Guardar”, el sistema valida que la información cumpla con los requisitos de obligatoriedad, formato y reglas de negocio que correspondan antes de guardar la información en la base de datos.
- Si los nuevos datos cumplen con todas la reglas, el sistema los envía a la base y notifica al usuario del éxito en el guardado.

Flujos alternos:

- Si la información no ha cumplido con todos los requisitos, el sistema no modifica la base de datos y despliega un mensaje con el detalle de los errores detectados para que el usuario pueda corregirlos e intentar el guardado posteriormente.
- El sistema dispondrá de una opción para cancelar la modificación que llevará al usuario a la pantalla inmediata anterior y no guardará ninguna modificación en base de datos.

Reglas de negocio:

- Para cada opción, habrá campos que se considerarán obligatorios y que serán los mínimos necesarios para que el registro pueda ser modificado.
- Dependiendo del tipo de datos, cada campo deberá cumplir con los requisitos generales y particulares del sistema.



- Los usuarios tendrán información propia que podrán manipular e información general que sólo podrán modificar si tienen los permisos necesarios.

Notas: En la figura 4.1.2.5 se muestra un diagrama de los pasos a seguir en este caso de uso.

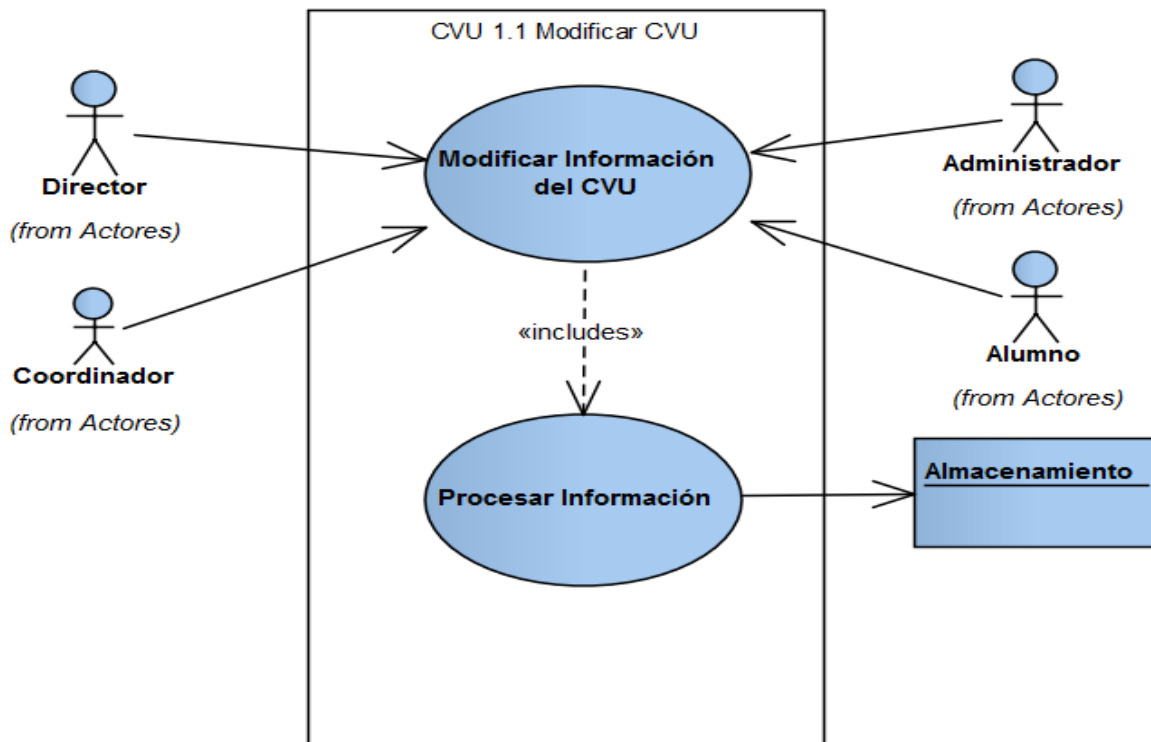


Fig. 4.1.2.5 Diagrama de caso de uso CVU-1.1



Identificador	CVU-1.2
Nombre	CREAR CVU

Actores: Estudiante/Profesor/Dirección de posgrado.

Descripción: A través de esta opción el usuario podrá ingresar nuevos registros o datos de su CVU, datos generales del usuario, libros, consultorías, redes y grupos de investigación, artículos, libros y tesis.

Precondiciones: N/A.

Postcondiciones: El registro se guarda en base de datos con la información ingresada por el usuario.

Flujo normal:

- El usuario acciona la opción “Agregar”.
- Se despliega el formulario correspondiente para ingresar los datos de su CVU, datos generales del usuario, libros, consultorías, redes y grupos de investigación, artículos, libros y tesis.
- Al terminar de llenar la información, el usuario acciona la opción “Guardar”.
- El sistema valida que los datos ingresados cumplan con los requisitos de obligatoriedad y formato correspondientes.
- El sistema valida que no haya duplicidad con la información previamente guardada en base.
- Se ingresa el nuevo registro a la base de datos.

Flujos alternos:

- Si la información no ha cumplido con todos los requisitos, el sistema no ingresa el nuevo registro a la base de datos y despliega un mensaje con el



detalle de los errores detectados para que el usuario pueda corregirlos e intentar el guardado posteriormente.

- El sistema dispondrá de una opción para cancelar la alta que llevará al usuario a la pantalla inmediata anterior y no guardará ninguna información en base de datos.
- Si se detecta duplicidad en la información, el sistema despliega un mensaje de error y no da de alta el nuevo registro.

Reglas de negocio:

- Para cada opción, habrá campos que se considerarán obligatorios y que serán los mínimos necesarios para que el registro pueda ser modificado.
- Dependiendo del tipo de datos, cada campo deberá cumplir con los requisitos generales y particulares del sistema.
- Los usuarios tendrán información propia que podrán manipular e información general que sólo podrán modificar si tienen los permisos necesarios.
- Cada información tendrá una o varias reglas para determinar la duplicidad que pueda impedir el alta de nuevo registros.

Notas: En la figura 4.1.2.6 se muestra un diagrama de los pasos a seguir en este caso de uso.

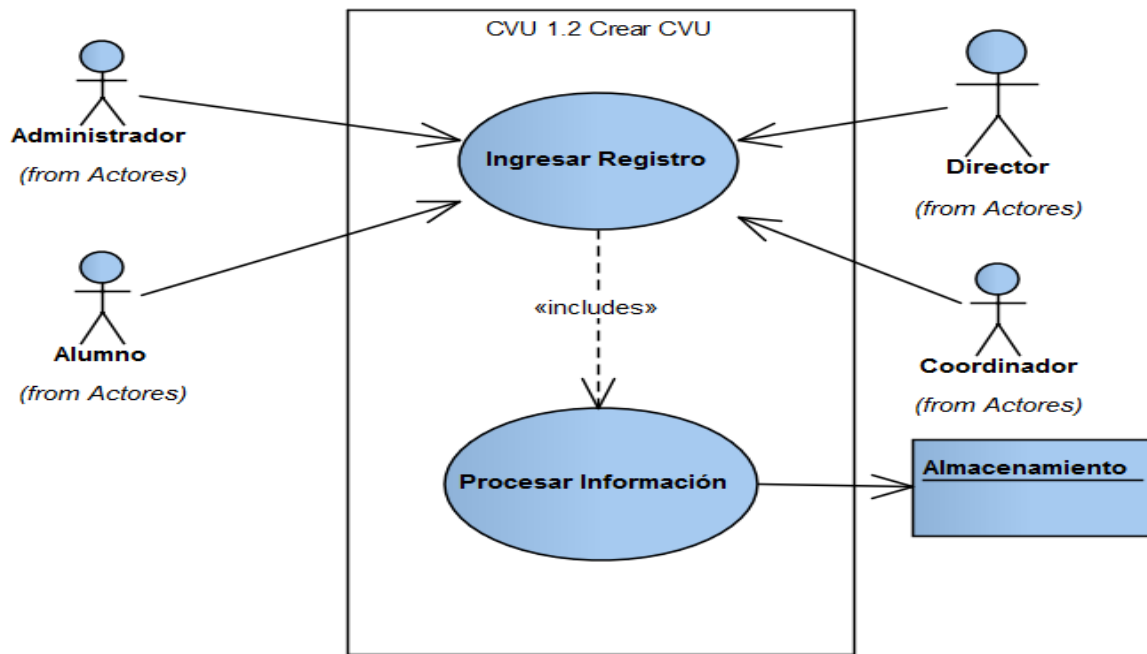


Fig. 4.1.2.6 Diagrama caso de uso CVU-1.2

4.1.3 Diagrama entidad relación.

El **modelo** de datos entidad relación está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de **relaciones** entre esos objetos. La **entidad** representa una “cosa” u “objeto” del mundo real con existencia independiente, es decir, se diferencia únicamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad. A las características que definen o identifican a una entidad se les llama **atributos** de esa entidad.



Se define una colección de tablas donde cada una tiene un número específico de columnas y un número arbitrario de filas. Cada elemento de la tabla guarda un valor de tipo primitivo, como enteros o cadenas. De esta manera, cada objeto se representa como una fila en una tabla y donde cada columna corresponde a un atributo distinto en el objeto. El lenguaje de consultas se basa en operaciones simples, donde la simplicidad del modelo es un beneficio pero también una limitación.²

Por motivos de legibilidad, los atributos suelen no aparecer representados en el diagrama entidad relación, sino descritos textualmente en otros documentos adjuntos. La relación describe cierta dependencia entre entidades o permite la asociación de las mismas.

A continuación se presentan dos de los principales diagramas entidad relación para el presente sistema.

En la figura 4.1.3.1 se muestra el diagrama entidad relación para el seguimiento a usuarios. Siendo concretos, en este diagrama se presentan las relaciones entre las entidades que se refieren al ingreso de datos del CVU (currículum vitae universitario). Se presentan las restricciones que se deben cumplir para poder ingresar nueva información respecto a un usuario y su currículum, como lo son sus datos generales, formación académica, participación en estancias de investigación, investigaciones desarrolladas, pertenencia a grupos de investigación, publicación de libros de su autoría, y distinciones recibidas.

² Weitzenfeld, A. (2005). Ingeniería de Software Orientada a Objetos con UML, Java e Internet.



El diagrama entidad relación extendido para el sistema que nos atañe, el del seguimiento al posgrado, se muestra en la figura 4.1.3.2. De igual forma, en esta figura se condensa de tal forma que solo se muestran los módulos más significativos del sistema web a implementar. Por mencionar algunos, se presenta la relación entre el seguimiento a posgrado y el seguimiento particular a los alumnos; a los laboratorios, aulas y bibliotecas con los que cuenta cada uno de los programas de posgrado de la universidad; a las opciones terminales de los posgrados; etc.



CAPÍTULO IV

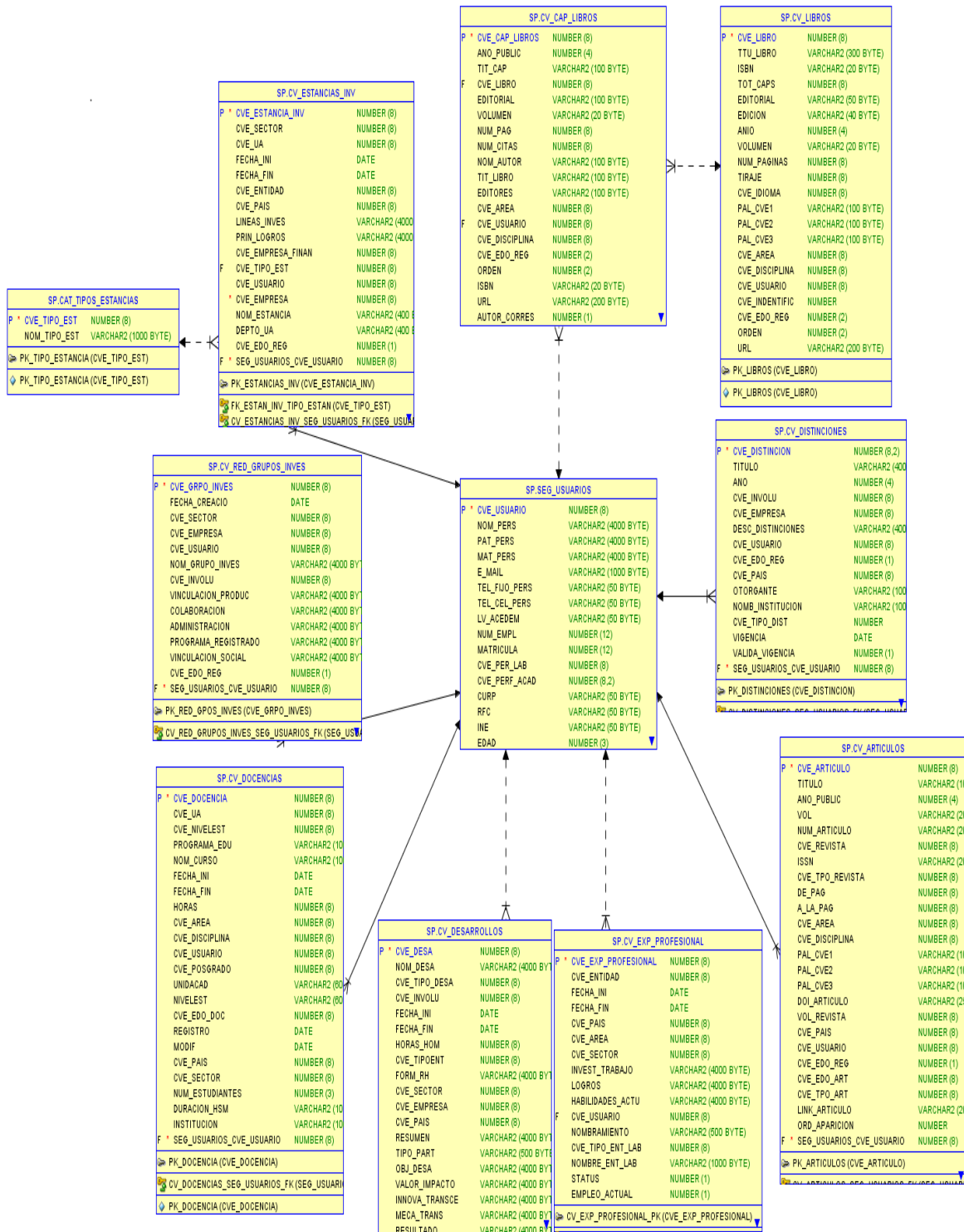


Fig. 4.1.3.1 Diagrama entidad relación para el seguimiento a usuarios



DISEÑO Y CONSTRUCCIÓN DEL SISTEMA WEB

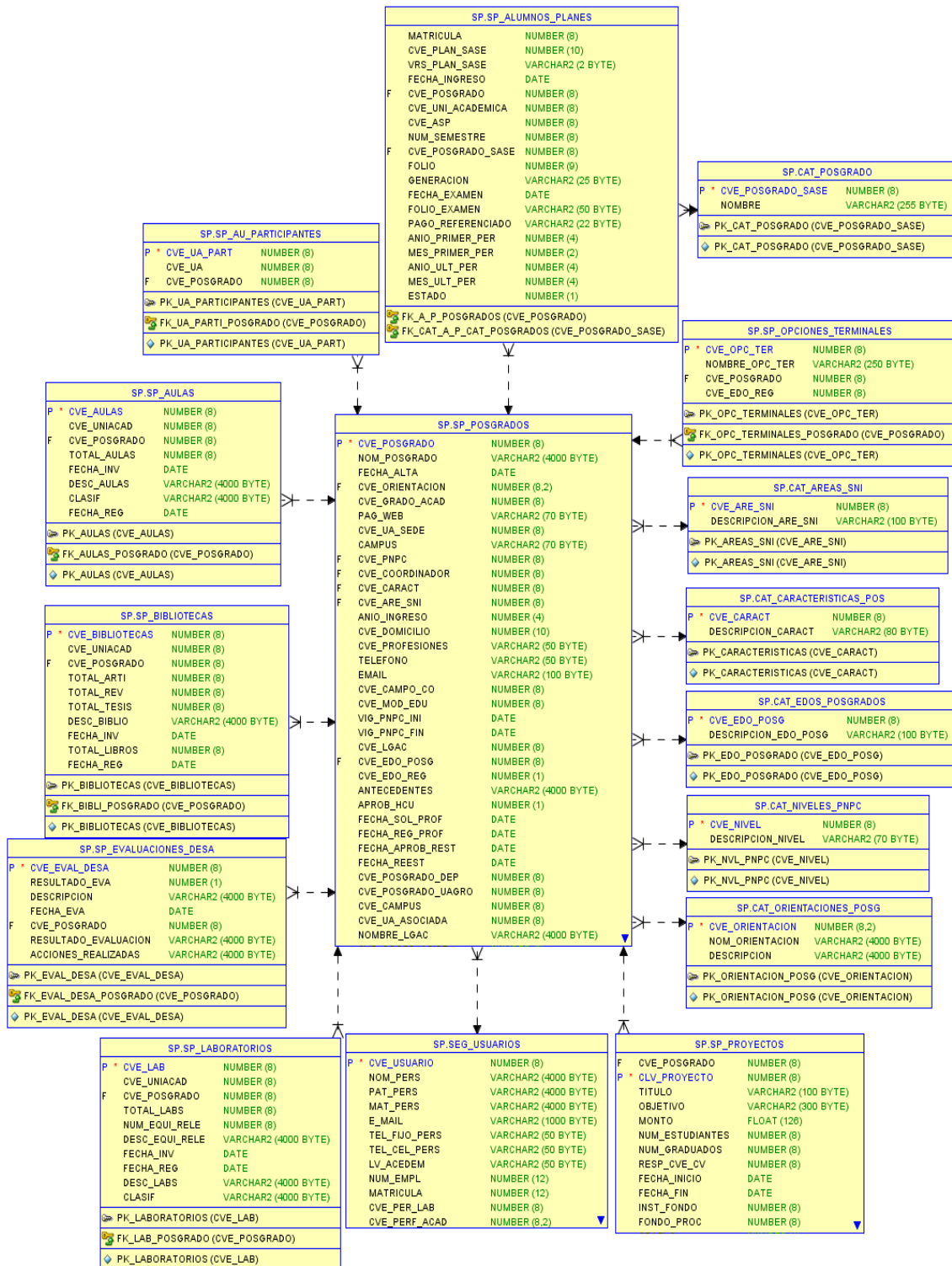


Fig. 4.1.3.2 Diagrama entidad relación para el seguimiento del posgrado



4.1.4 Diccionario de datos.

Un diccionario de datos es una forma de documentar una base de datos. En él se deposita la información como nombre, tipo y descripciones, de igual forma se encuentra información relacionada con los diferentes tipos de registros, privilegios de los usuarios y estadísticas.

Entre sus funciones se encuentran las siguientes:

- Describir las distintas características de cada uno de los módulos de la base de datos.
- Proveer una descripción de cada tabla y explicar el propósito de la misma.
- Muestra la estructura de cada tabla, incluyendo sus campos y sus tipos de datos, todos los índices y todas las vistas.
- Describir reglas, como pueden ser el uso de valores no nulos o valores únicos.

A continuación presentamos algunas de las tablas del diccionario de datos más representativas para el sistema web.



CVU.

- **Aplicaciones (APLICACIONES).** Tabla que almacena la información de los módulos o aplicaciones (tablas 4.1.4.1-4.1.4.3).

PK	Nombre	Tipo	No Nulo	Único	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_app	NUMBER	Verdadero	Falso	0	38	0		Clave de aplicación
Falso	nom_app	VARCHAR2	Falso	Falso	50	0	0		Nombre de Aplicación
Falso	dsc_app	VARCHAR2	Falso	Falso	50	0	0		Descripción de Aplicación

Tabla 4.1.4.1 Columnas

PK	Nombre	Tipo	No Nulo	Único	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_app	NUMBER	Verdadero	Falso	0	38	0		Clave de aplicación
Falso	nom_app	VARCHAR2	Falso	Falso	50	0	0		Nombre de Aplicación
Falso	dsc_app	VARCHAR2	Falso	Falso	50	0	0		Descripción de Aplicación

Tabla 4.1.4.2 Restricciones

Columnas	Asociación	Notas
	0..* COMPONENTES.FK_COMPONENTES_APLICACIONES 1 APLICACIONES.PK_APLICACIONES	

Tabla 4.1.4.3 Relaciones

- **Componentes (COMPONENTES).** Tabla que almacena el catálogo de componentes asociados a cada módulo o aplicación del sistema (tablas 4.1.4.4-4.1.4.6).



CAPÍTULO IV



PK	Nombre	Tipo	No Nulo	único	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_compte	NUMBER	Verdadero	Falso	0	38	0		Clave de componente
Falso	cve_app	NUMBER	Falso	Falso	0	38	0		Clave de aplicación o modulo
Falso	nom_compte	VARCHAR2	Falso	Falso	50	0	0		Nombre del componente
Falso	dsc_compte	VARCHAR2	Falso	Falso	50	0	0		Descripción del componente

Tabla 4.1.4.4 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
PK_COMPONENTES	Public	cve_compte		

Tabla 4.1.4.5 Restricciones

Columnas	Asociación	Notas
	0..* ROLES_PRIVILEGIOS.FK_ROLES_PRIVILEGIOS_COMPONENTES 1 COMPONENTES.PK_COMPONENTES	
(cve_compte = cve_compte)	0..* PRIVILEGIOS_COMPTE.FK_PRIVILEGIOS_COMPTE_COMPONENTES 1 COMPONENTES.PK_COMPONENTES	
(cve_compte = cve_compte)	0..* Empresa.FK_Empresa_COMPONENTES 1 COMPONENTES.PK_COMPONENTES	
(compte = cve_compte)	0..* USUARIOS_PRIVILEGIOS.FK_USUARIOS_PRIVILEGIOS_COMPONENTES 1 COMPONENTES.PK_COMPONENTES	
	0..* COMPONENTES.FK_COMPONENTES_APLICACIONES 1 APLICACIONES.PK_APLICACIONES	
	0..* ARCHIVOS.FK_ARCHIVOS_COMPONENTES 1 COMPONENTES.PK_COMPONENTES	

Tabla 4.1.4.6 Relaciones



- **Datos generales (DATOS_GENERALES).** Tabla que almacena los datos generales del CVU del alumno o profesor (tablas 4.1.4.7-4.1.4.9).

PK	Nombre	Tipo	No Nulo	Único	Longitud	Precisión	Escala	Inicial	Notas
Falso	matricula_numemp	NUMBER	Falso	Falso	0	10	0		Matricula o Número de empleado
Verdadero	cve_datos_generales	NUMBER	Verdadero	Verdadero	0	10	0		Clave de datos generales
Falso	CURP	VARCHAR2	Falso	Falso	20	0	0		CURP
Falso	cve_usuario	NUMBER	Falso	Falso	0	10	0		Clave del usuario
Falso	email	VARCHAR2	Falso	Falso	300	0	0		Correo electrónico
Falso	tel_movil	VARCHAR2	Falso	Falso	300	0	0		Teléfono móvil
Falso	tel_fijo	VARCHAR2	Falso	Falso	300	0	0		Teléfono fijo
Falso	fecha_ing	DATE	Falso	Falso	0	0	0		Fecha ingreso de datos generales del usuario
Falso	cve_perfilAcad	NUMBER	Falso	Falso	0	8	0		Clave del perfil Académico
Falso	cve_perflabo	NUMBER	Falso	Falso	0	8	0		Clave del perfil laboral
Falso	rfc	VARCHAR2	Falso	Falso	20				Registro federal de contribuyente
Falso	edo_civil	VARCHAR2	Falso	Falso	20				Estado civil del usuario
Falso	sexo	CHAR	Falso	Falso	1				Sexo del usuario
Falso	fecha_nac	DATE	Falso	Falso					Fecha de nacimiento del usuario

Tabla 4.1.4.7 Columnas



CAPÍTULO IV



Nombre	Tipo	Columnas	Código Inicial	Notas
FK_DATOS_GENERALES_perfil_laboral	Public	cve_perflabo		
FK_DATOS_GENERALES_PerfilAcademico	Public	cve_perfilAcad		
PK_DATOS_GENERALES	Public	cve_datos_generales		
UQ_DATOS_GENERAL_cve_datos_ge	Public	cve_datos_generales		

Tabla 4.1.4.8 Restricciones

Columnas	Asociación	Notas
(cve_perfilAcad = cve_perfilAcad)	0..* DATOS_GENERALES.FK_DATOS_GENERALES_PerfilAcademico 1 PerfilAcademico.PK_PerfilAcademico	
(cve_perflabo = cve_perflabo)	0..* DATOS_GENERALES.FK_DATOS_GENERALES_perfil_laboral 1 perfil_laboral.PK_perfil_laboral	

Tabla 4.1.4.9 Relaciones

- **Privilegios (PRIVILEGIOS).** Tabla que almacena el catálogo de privilegios de todo el sistema (tablas 4.1.4.10-4.1.4.12).

PK	Nombre	Tipo	No Nulo	Único	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_priv	NUMBER	Verdadero	Falso	0	38	0		Clave de privilegio
Falso	nom_priv	VARCHAR2	Falso	Falso	50	0	0		Nombre del privilegio (Consulta, edición, alta y baja)
Falso	dsc_priv	VARCHAR2	Falso	Falso	50	0	0		Descripción del privilegio

Tabla 4.1.4.10 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
PK_PRIVILEGIOS	Public	cve_priv		

Tabla 4.1.4.11 Restricciones



Columnas	Asociación	Notas
(cve_priv = cve_priv)	0..* USUARIOS_PRIVILEGIOS.FK_USUARIOS_PRIVILEGIOS_PRIVILEGIOS 1 PRIVILEGIOS.PK_PRIVILEGIOS	
	0..* ROLES_PRIVILEGIOS.FK_ROLES_PRIVILEGIOS_PRIVILEGIOS 1 PRIVILEGIOS.PK_PRIVILEGIOS	
(cve_priv = cve_priv)	0..* PRIVILEGIOS_COMPTES.FK_PRIVILEGIOS_COMPTE_PRIVILEGIOS 1 PRIVILEGIOS.PK_PRIVILEGIOS	

Tabla 4.1.4.12 Relaciones

- **Usuarios (USUARIOS).** Almacena los usuarios que se autenticarán con el sistema (tablas 4.1.4.13-4.1.4.15).

PK	Nombre	Tipo	No Nulo	Unico	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_usuario	NUMBER	Verdadero	Verdadero	0	8	0		Clave del usuario
Falso	nom_usuario	VARCHAR2	Verdadero	Falso	100	0	0		Nombre abreviado del usuario
Falso	contrasena	VARCHAR2	Falso	Falso	1000	0	0		Contraseña del usuario
Falso	email	VARCHAR2	Verdadero	Verdadero	50	0	0		Correo electrónico
Falso	cve_tipo_usuario	NUMBER	Falso	Falso	0	8	0		clave del tipo de usuario fk
Falso	cve_edo_usuario	NUMBER	Falso	Falso	0	2	0		Clave del estado de usuario
Falso	fecha_vigen	DATE	Falso	Falso					fecha hasta que el usuario tendrá acceso

Tabla 4.1.4.13 Columnas



CAPÍTULO IV



Nombre	Tipo	Columnas	Código Inicial	Notas
FK_USUARIOS_ESTADO_USUARIO	Public	cve_edo_usuario		
FK_USUARIOS_TIPO_USUARIO	Public	cve_tipo_usuario		
PK_USUARIOS	Public	cve_usuario		
UQ_USUARIOS_cve_usuario	Public	cve_usuario		
UQ_USUARIOS_email	Public	email		

Tabla 4.1.4.14 Restricciones

Columnas	Asociación	Notas
(cve_usuario = cve_usuario)	0..* cap_libros.FK_cap_libros_USUARIOS 1 USUARIOS.PK_USUARIOS	
(cve_usuario = cve_usuario)	0..* libros.FK_libros_USUARIOS 1 USUARIOS.PK_USUARIOS	
(cve_tipo_usuario = cve_tipo_usuario)	0..* USUARIOS.FK_USUARIOS_TIPO_USUARIO 1 TIPO_USUARIO.PK_TIPO_USUARIO	
(cve_usuario = cve_usuario)	0..* Red_gpos_inves.FK_Red_gpos_inves_USUARIOS 1 USUARIOS.PK_USUARIOS	
(cve_usuario = cve_usuario)	0..* Estancias_inv.FK_Estancias_inv_USUARIOS 1 USUARIOS.PK_USUARIOS	
(cve_usuario = cve_usuario)	0..* investigacion.FK_investigacion_USUARIOS 1 USUARIOS.PK_USUARIOS	
(cve_edo_usuario = cve_edo_usuario)	0..* USUARIOS.FK_USUARIOS_ESTADO_USUARIO 1 ESTADO_USUARIO.PK_ESTADO_USUARIO	
(cve_usuario = cve_usuario)	0..* desarrollos.FK_desarrollos_USUARIOS 1 USUARIOS.PK_USUARIOS	



(cve_usuario = cve_usuario)	0..*	fomacion.FK_fomacion_USUARIOS	
	1	USUARIOS.PK_USUARIOS	

Tabla 4.1.4.15 Relaciones

- **Privilegios del usuario (USUARIOS_PRIVILEGIOS).** Tabla que almacena la información de los privilegios asignados a un usuario (tablas 4.1.4.16-4.1.4.18).

PK	Nombre	Tipo	No Nulo	Unico	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_usuario	NUMBER	Verdadero	Falso	0	38	0		Clave de usuario
Verdadero	cve_priv	NUMBER	Verdadero	Falso	0	15	4		Clave de privilegios
Verdadero	cve_compte	NUMBER	Verdadero	Falso	0	15	4		Clave de componente

Tabla 4.1.4.16 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
FK_USUARIOS_PRIVILEGIOS_COMPONENTES	Public	cvente		
FK_USUARIOS_PRIVILEGIOS_PRIVILEGIOS	Public	cve_priv		
FK_USUARIOS_PRIVILEGIOS_USUARIOS	Public	cve_usuario		
PK_USUARIOS_PRIVILEGIOS	Public	cve_usuario cve_priv cve_compte		

Tabla 4.1.4.17 Restricciones

Columnas	Asociación	Notas
(cve_priv = cve_priv)	0..* USUARIOS_PRIVILEGIOS.FK_USUARIOS_PRIVILEGIOS_PRIVILEGIOS 1 PRIVILEGIOS.PK_PRIVILEGIOS	
(compte = cve_compte)	0..* USUARIOS_PRIVILEGIOS.FK_USUARIOS_PRIVILEGIOS_COMPONENTES 1 COMPONENTES.PK_COMPONENTES	

Tabla 4.1.4.18 Relaciones



Posgrado.

- **Alumnos (ALUMNOS).** Tabla con datos de alumnos (tablas 4.1.4.19-4.1.4.21).

PK	Nombre	Tipo	No Nulo	Unico	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	matricula_alumno	NUMBER	Verdadero	Verdadero	0	8	0		Matricula del alumno, clave primaria
Falso	nom_alumno	VARCHAR2	Falso	Falso	4000	0	0		Nombre del alumno
Falso	ape_paterno	VARCHAR2	Falso	Falso	4000	0	0		Apellido Paterno del alumno
Falso	ape_materno	VARCHAR2	Falso	Falso	4000	0	0		Apellido materno del alumno
Falso	cve_posgrado	NUMBER	Falso	Falso	0	8	0		Clave del posgrado
Falso	num_semestre	NUMBER	Falso	Falso	0	8	0		Semestre cursado del alumno
Falso	fecha_ingreso	VARCHAR2	Falso	Falso	4000	0	0		Fecha de ingreso del alumno
Falso	crup_alumno	VARCHAR2	Falso	Falso	4000	0	0		Curp del alumno
Falso	mail_alumno	VARCHAR2	Falso	Falso	4000	0	0		Correo del alumno
Falso	cve_uni_academica	NUMBER	Falso	Falso	0	8	0		Clave de la unidad académica del alumno
Falso	cve_idioma	NUMBER	Falso	Falso	0	8	0		Clave del idioma
Falso	edo_civil	VARCHAR2	Falso	Falso	4000	0	0		Estado civil del alumno



DISEÑO Y CONSTRUCCIÓN DEL SISTEMA WEB



Falso	etnia	VARCHAR2	Falso	Falso	4000	0	0	Etnia del alumno
Falso	cve_plan_estudios	NUMBER	Falso	Falso	0	8	0	Clave del plan de estudios
Falso	proceden	VARCHAR2	Falso	Falso	80	0	0	País de procedencia del alumno
Falso	sexo	VARCHAR2	Falso	Falso	4000	0	0	Genero del alumno
Falso	cve_asp	NUMBER	Falso	Falso	0	8	0	Clave de su registro como aspirante

Tabla 4.1.4.19 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
FK_ALUMNOS_Aspirante	Public	cve_asp		
FK_ALUMNOS_plan_estudios	Public	cve_plan_estudios		
FK_ALUMNOS_Posgrado	Public	cve_posgrado		
FK_ALUMNOS_Unidad_academica	Public	cve_uni_academica		
PK_Alumnos	Public	matricula_alumno		
UQ_Alumnos_matricula_alumno	Public	matricula_alumno		

Tabla 4.1.4.20 Restricciones

Columnas	Asociación	Notas
(matricula = matricula_alumno)	0..* Tutoria.FK_Tutoria_ALUMNOS 1 ALUMNOS.PK_Alumnos	
(cve_uni_academica = cve_ua)	0..* ALUMNOS.FK_ALUMNOS_Unidad_academica 1 Unidad_academica.PK_Unidad_academica	
(matricula_alumno = matricula_alumno)	0..* INTEGRANTES.FK_INTEGRANTES_ALUMNOS 1 ALUMNOS.PK_Alumnos	
(matricula = matricula_alumno)	0..* Movilidad.FK_Movilidad_ALUMNOS 1 ALUMNOS.PK_Alumnos	



CAPÍTULO IV



(cve_plan_estudios = cve_plan)	0..*	ALUMNOS.FK_ALUMNOS_plan_estudios	
	1	plan_estudios.PK_plan_estudios	
(matricula = matricula_alumno)	0..*	MATERIAS_ALUMNO.FK_MATERIAS_ALUMNO_ALUMNOS	
	1	ALUMNOS.PK_Alumnos	
(cve_posgrado = cve_posgrado)	0..*	ALUMNOS.FK_ALUMNOS_Posgrado	
	1	Posgrado.PK_Posgrado	
(cve_esp = cve_esp)	0..*	ALUMNOS.FK_ALUMNOS_Aspirante	
	1	Aspirante.PK_Aspirante	

Tabla 4.1.4.21 Relaciones

- **Estado de la tesis (EDO_TESIS).** Tabla que almacena la información del estado de las tesis (tablas 4.1.4.22-4.1.4.24).

PK	Nombre	Tipo	No Nulo	Unico	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_edo_tesis	NUMBER	Verdadero	Verdadero	0	8	0		Clave única del estado de tesis
Falso	nom_edo_tesis	VARCHAR2	Falso	Falso	4000	0	0		Nombre del estado de la tesis
Falso	desc_tesis	VARCHAR2	Falso	Falso	4000	0	0		Descripción sobre la tesis

Tabla 4.1.4.22 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
PK_EDO_TESIS	Public	cve_edo_tesis		
UQ_EDO_TESIS_cve_edo_tesis	Public	cve_edo_tesis		

Tabla 4.1.4.23 Restricciones

Columnas	Asociación	Notas
(cve_edo_tesis = cve_edo_tesis)	0..* TESIS.FK_TESIS_EDO_TESIS	
	1 EDO_TESIS.PK_EDO_TESIS	

Tabla 4.1.4.24 Relaciones



- **Exámenes de las tesis (EXAMENES).** Tabla que almacena los exámenes de las tesis (tablas 4.1.4.25-4.1.4.27).

PK	Nombre	Tipo	No Nulo	Unico	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_examen	NUMBER	Verdadero	Falso	0	8	0		Clave del examen
Falso	fecha_examen	BFILE	Falso	Falso	0	0	0		Fecha de realización de examen
Falso	cve_tesis	NUMBER	Falso	Falso	0	8	0		Clave de la tesis

Tabla 4.1.4.25 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
FK_EXAMENES_TESIS	Public	cve_tesis		
PK_EXAMENES	Public	cve_examen		

Tabla 4.1.4.26 Restricciones

Columnas	Asociación	Notas
(cve_examen = cve_examen)	0..* JURADOS.FK_JURADOS_EXAMENES	
	1 EXAMENES.PK_EXAMENES	
(cve_tesis = cve_tesis)	0..* EXAMENES.FK_EXAMENES_TESIS	
	1 TESIS.PK_TESIS	

Tabla 4.1.4.27 Relaciones

- **Tesis (TESIS).** Tabla que almacena la información de las tesis de los alumnos (tablas 4.1.4.28-4.1.4.30).



CAPÍTULO IV



PK	Nombre	Tipo	No Nulo	Unico	Longitud	Precisión	Escala	Inicial	Notas
Verdadero	cve_tesis	NUMBER	Verdadero	Verdadero	0	8	0		Clave única de la tesis
Falso	titulo_tesis	VARCHAR2	Falso	Falso	4000	0	0		Título que reconoce la tesis
Falso	fecha_reg_proto	DATE	Falso	Falso	0	0	0		Fecha de registro del protocolo
Falso	comentario_tesis	VARCHAR2	Falso	Falso	4000	0	0		Comentario sobre la tesis
Falso	porcen_total_tesis	NUMBER	Falso	Falso	0	8	0		Porcentaje total de pertenencia a la tesis
Falso	aprov_proto	BLOB	Falso	Falso	0	0	0		Estado de aprobación del protocolo
Falso	cve_edo_tesis	NUMBER	Falso	Falso	0	8	0		Clave única del estado de tesis
Falso	fecha_reg_tesis	DATE	Falso	Falso	0	0	0		Fecha de registro de la tesis
Falso	comentario_proto	VARCHAR2	Falso	Falso	4000	0	0		Comentario de protocolo de tesis
Falso	fecha_aprov_proto	NUMBER	Falso	Falso	0	8	0		Fecha de aprobación a protocolo de tesis

Tabla 4.1.4.28 Columnas

Nombre	Tipo	Columnas	Código Inicial	Notas
FK_TESIS_EDO_TESIS	Public	cve_edo_tesis		
PK_TESIS	Public	cve_tesis		
UQ_TESIS_cve_tesis	Public	cve_tesis		

Tabla 4.1.4.29 Restricciones



Columnas	Asociación	Notas
(cve_tesis = cve_tesis)	0..* REVISION_TESIS.FK_REVISION_TESIS_TESIS 1 TESIS.PK_TESIS	
(cve_tesis = cve_tesis)	0..* EXAMENES.FK_EXAMENES_TESIS 1 TESIS.PK_TESIS	
(cve_tesis = cve_tesis)	0..* INTEGRANTES.FK_INTEGRANTES_TESIS 1 TESIS.PK_TESIS	
(cve_edo_tesis = cve_edo_tesis)	0..* TESIS.FK_TESIS_EDO_TESIS 1 EDO_TESIS.PK_EDO_TESIS	
(cve_tesis = cve_tesis)	0..* REU_SEG_TESIS.FK_REU_SEG_TESIS_TESIS 1 TESIS.PK_TESIS	
(cve_tesis = cve_tesis)	0..* Acad_tesis.FK_Acad_tesis_TESIS 1 TESIS.PK_TESIS	

Tabla 4.1.4.30 Relaciones

4.1.5 Diagrama de clases.

Las clases se describen por medio del diagrama de clases. La notación para una clase es una caja rectangular, que contiene el nombre de la clase. La notación general para el objeto se extiende mediante el nombre de la clase subrayado seguido del nombre del objeto. Por lo general, se utilizan más los diagramas de clases que los diagramas de objetos, ya que los diagramas de clases son más generales y corresponden a varios diagramas de objetos.³

Para el caso del Sistema de Productividad del Posgrado de una Universidad, se ha diseñado una estructura de clases como se muestra en la figura 4.1.5.1.

³ Weitzenfeld, A. (2005). Ingeniería de Software Orientada a Objetos con UML, Java e Internet.



En donde *modulo* corresponde a los identificados durante el análisis (“estructura”, “investigación”, “vinculación”, “permanencia”, “ingreso” e “infraestructura”).

En el paquete “general” se colocarán clases abstractas con las generalizaciones de cada tipo de componente que ayudarán en la determinación de configuraciones, propiedades y métodos comunes en todo el sistema.

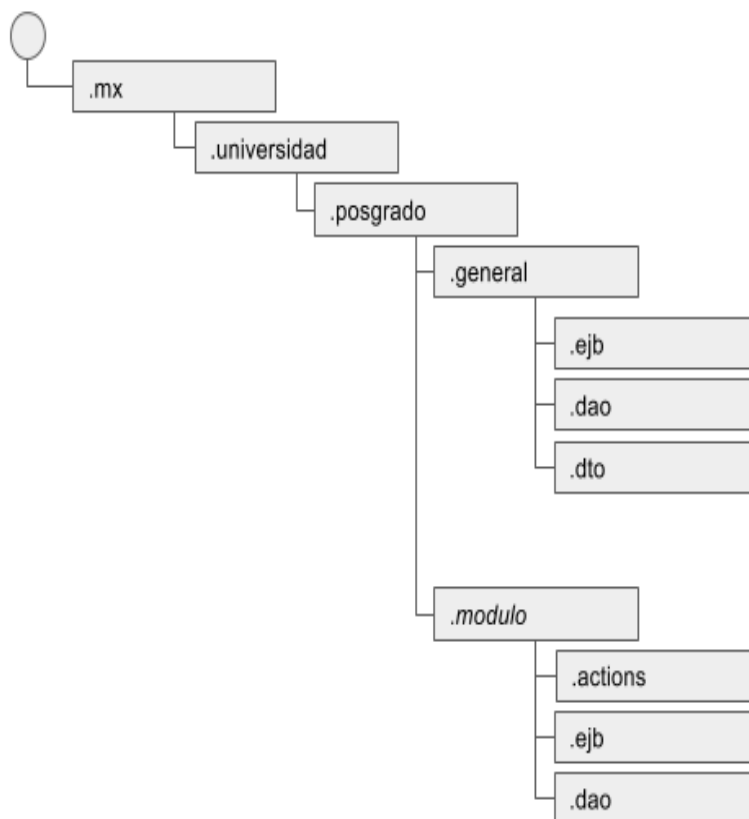


Fig. 4.1.5.1 Estructura de clases



Como todos los registros en base tendrán asociado un identificador y una descripción, se hace una generalización que simplifique el manejo de la persistencia. En la figura 4.1.5.2 se ilustra esta generalización de clases.

DIAGRAMA DE CLASES (mx.universidad.posgrado.general.dto)

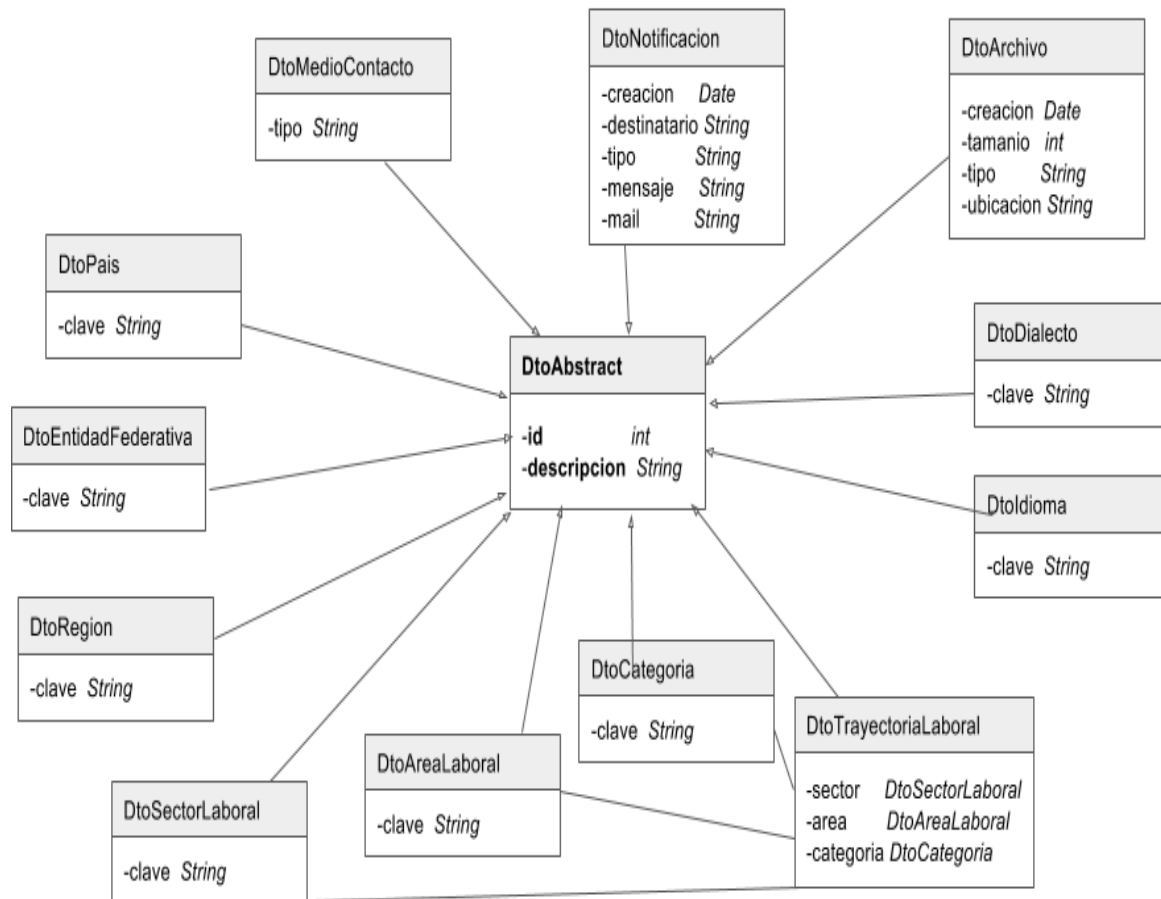


Fig. 4.1.5.2 Generalización de clases que mapean registros en base de datos

En los paquetes de negocio y acceso a datos, se utilizará una estructura de Interfaz-Implementación que nos brindará la modularización de estos paquetes para mejorar la robustez de nuestra aplicación.



DIAGRAMA DE CLASES (mx.universidad.posgrado.estructura)

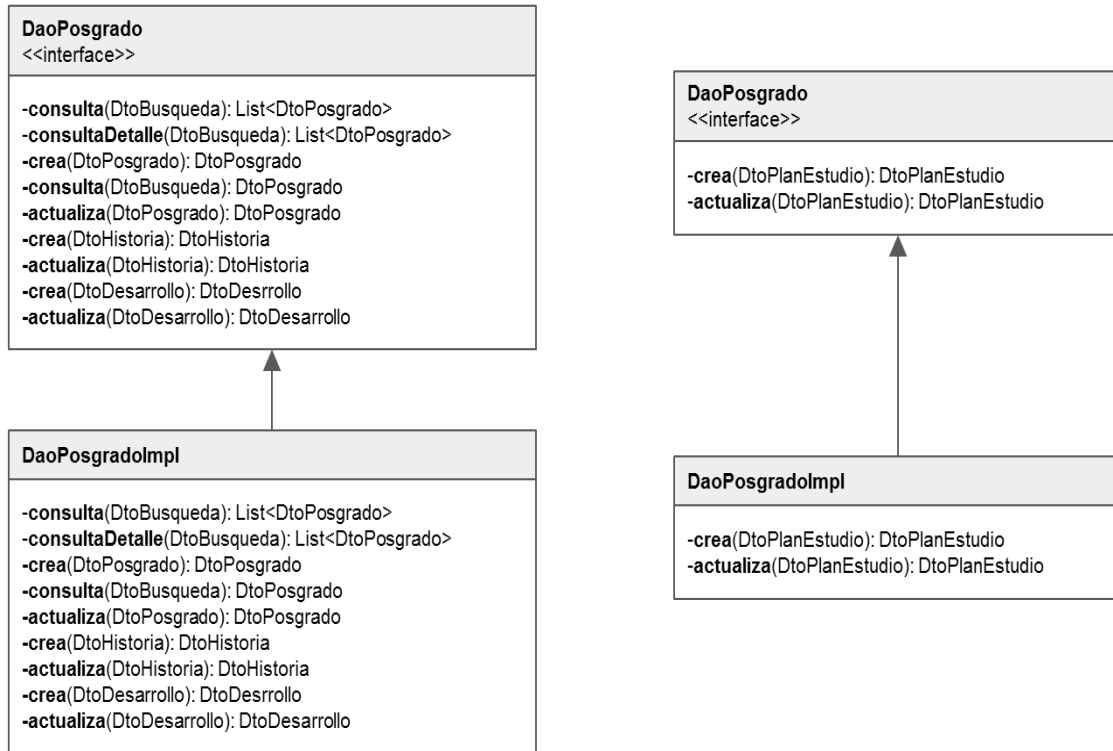


Fig. 4.1.5.3 Modelado de clases Interfaz-Implementación

El diagrama con las clases para el mapeo de tablas se muestra en las figuras 4.1.5.4 y 4.1.5.5.

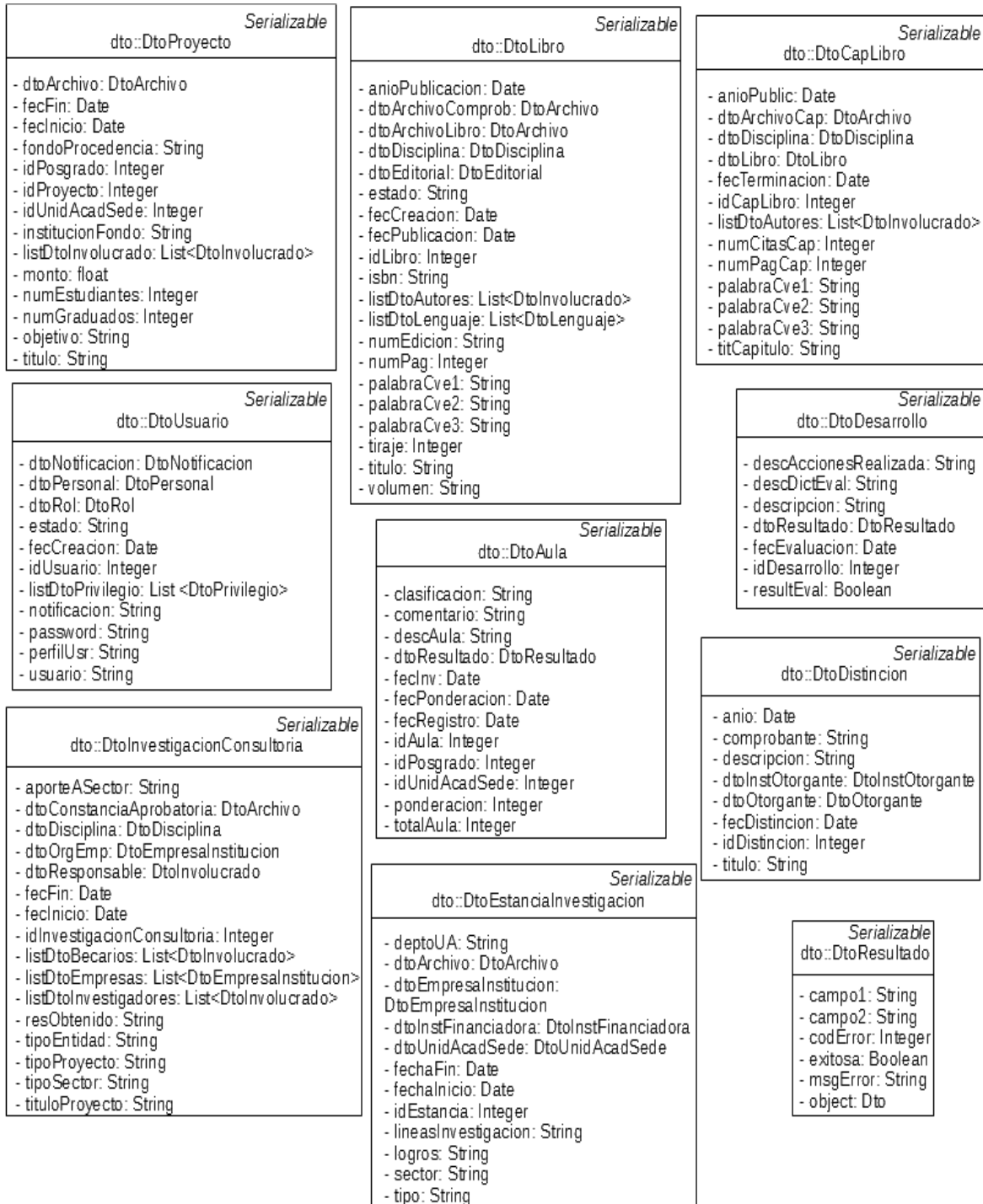


Fig. 4.1.5.4 Clases para mapeo de registros (I)



CAPÍTULO IV

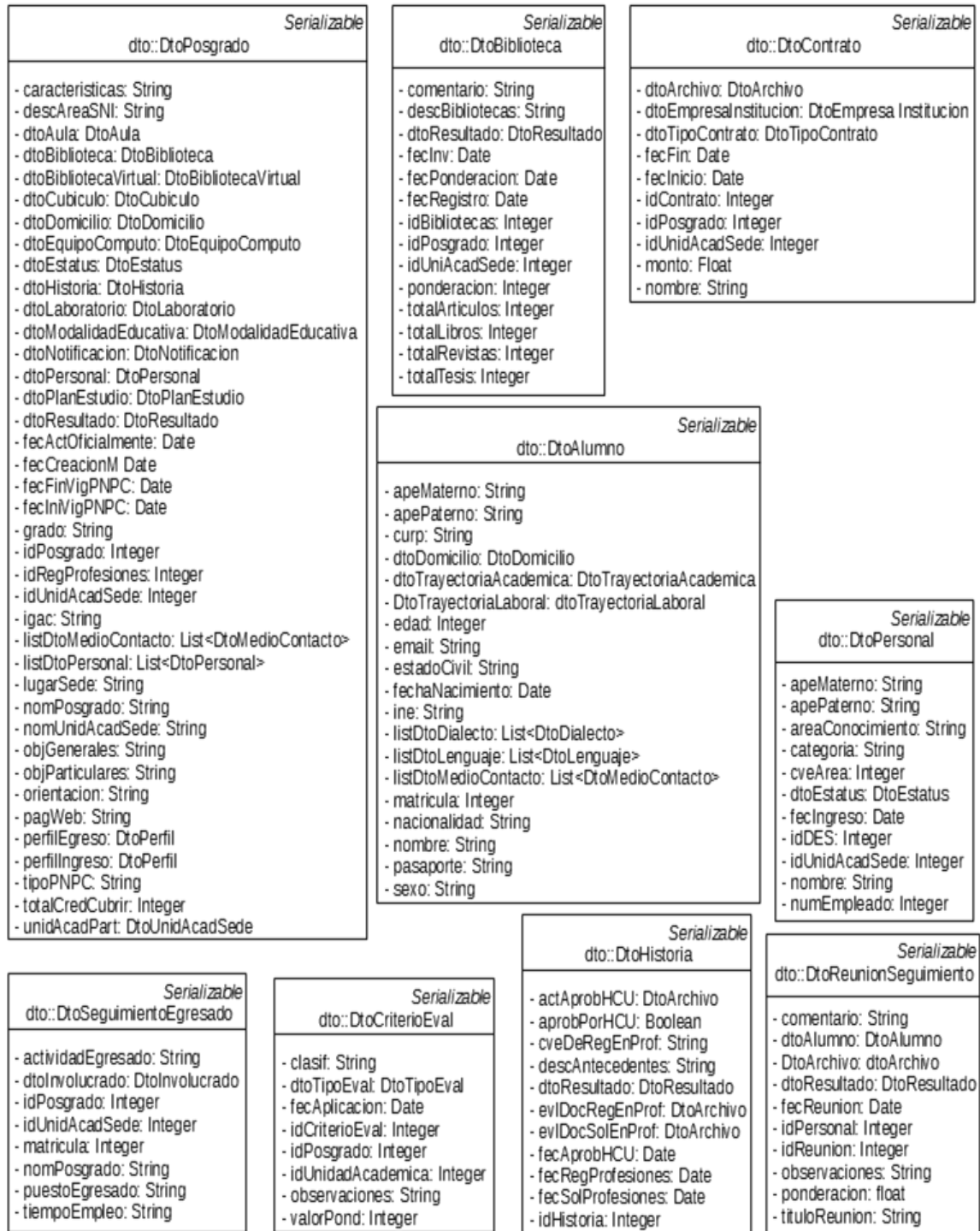


Fig. 4.1.5.5 Clases para mapeo de registros (II)



4.2 Construcción y diseño de la base de datos.

Un componente muy importante de nuestra aplicación es nuestro repositorio de información que como hemos mencionado será una base de datos relacional cuyo manejador será Oracle Database 11g.

Para la puesta a punto de nuestra base de datos seguiremos los siguientes pasos:

- Instalación del manejador.
- Creación de la base de datos.
- Creación del “tablespaces”.

En el caso de la instalación del manejador, usaremos el asistente de Oracle ya que cumple con los requisitos necesarios, no será necesario la modificación de los parámetros por default ya que estas adaptaciones se realizarán al crear la base de datos. Los únicos parámetros que debemos ingresar son las rutas de nuestro sistema de archivos. Vea figura 4.2.1 y 4.2.2.

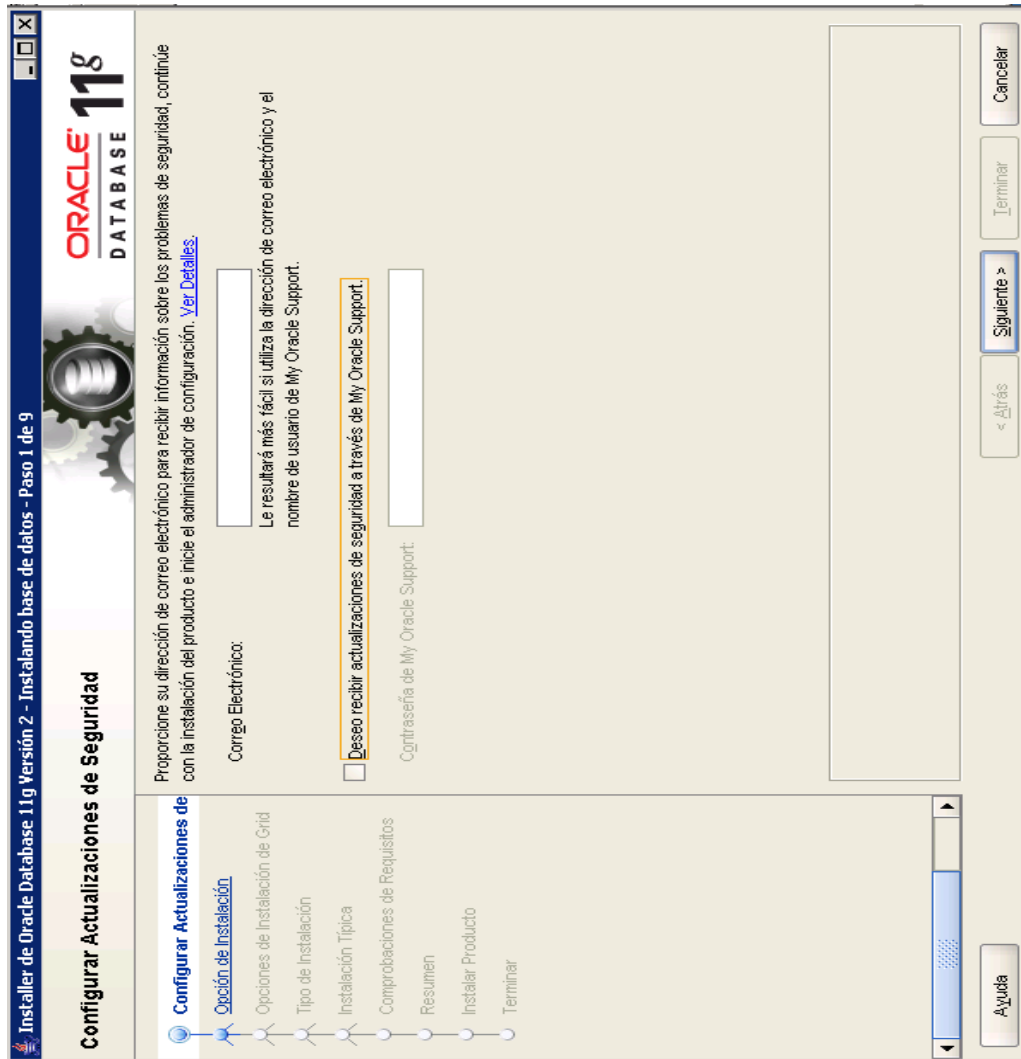


Fig. 4.2.1 Asistente de instalación del manejador de Oracle

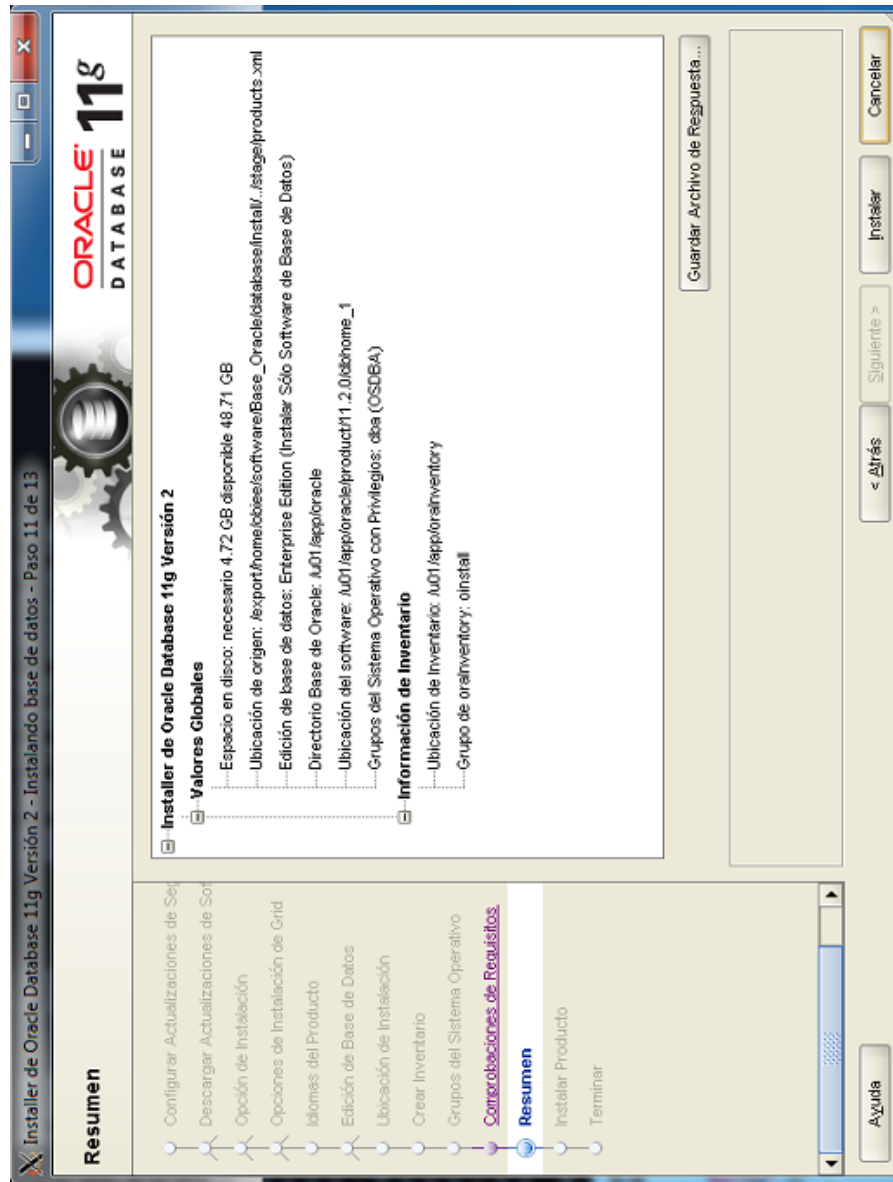


Fig. 4.2.2 Terminamos con el asistente de instalación



Después de haber instalado el manejador de Oracle Database 11g, proseguimos a crear la base de datos para el Data Ware House. Se usará el asistente de Oracle por la facilidad y efectividad. Para ejecutar el asistente podemos usar los siguientes comandos:

- \$ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
- \$PATH=ORACLE_HOME/bin:\$PATH
- \$export ORACLE_HOME PATH
- \$dbca

En la pantalla de bienvenida al asistente daremos en “Siguiente”. Vea figura 4.2.3.



Fig. 4.2.3 Asistente para la creación de la base de datos



En la siguiente pantalla elegiremos la operación de “Crear Base de Datos”. Vea figura 4.2.4.

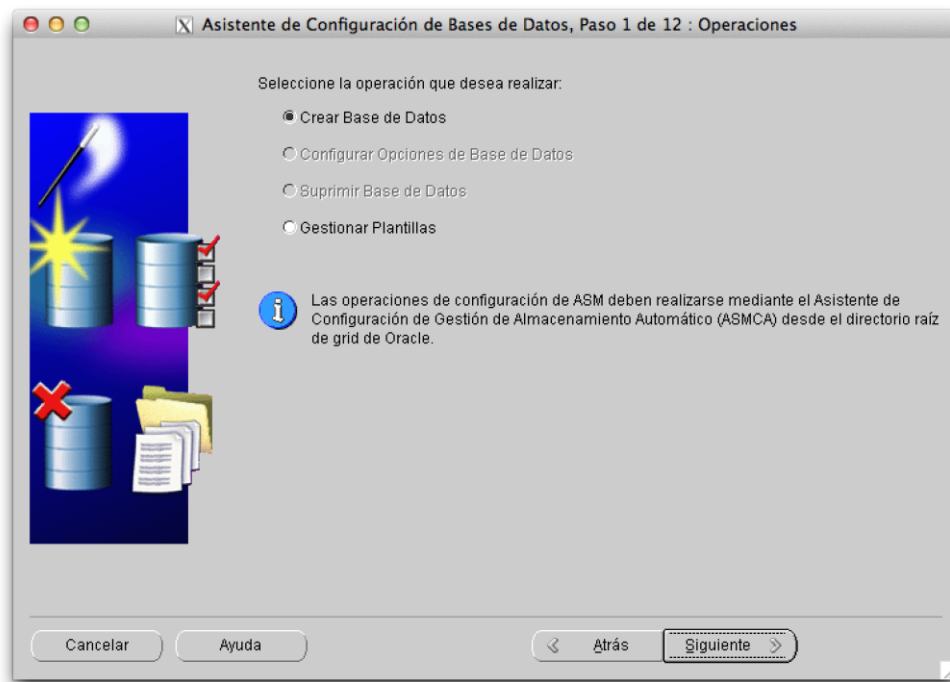


Fig. 4.2.4 Elegimos “Crear Base de Datos”



Después elegiremos la opción “Personalizar Base de Datos”. Vea figura 4.2.5.

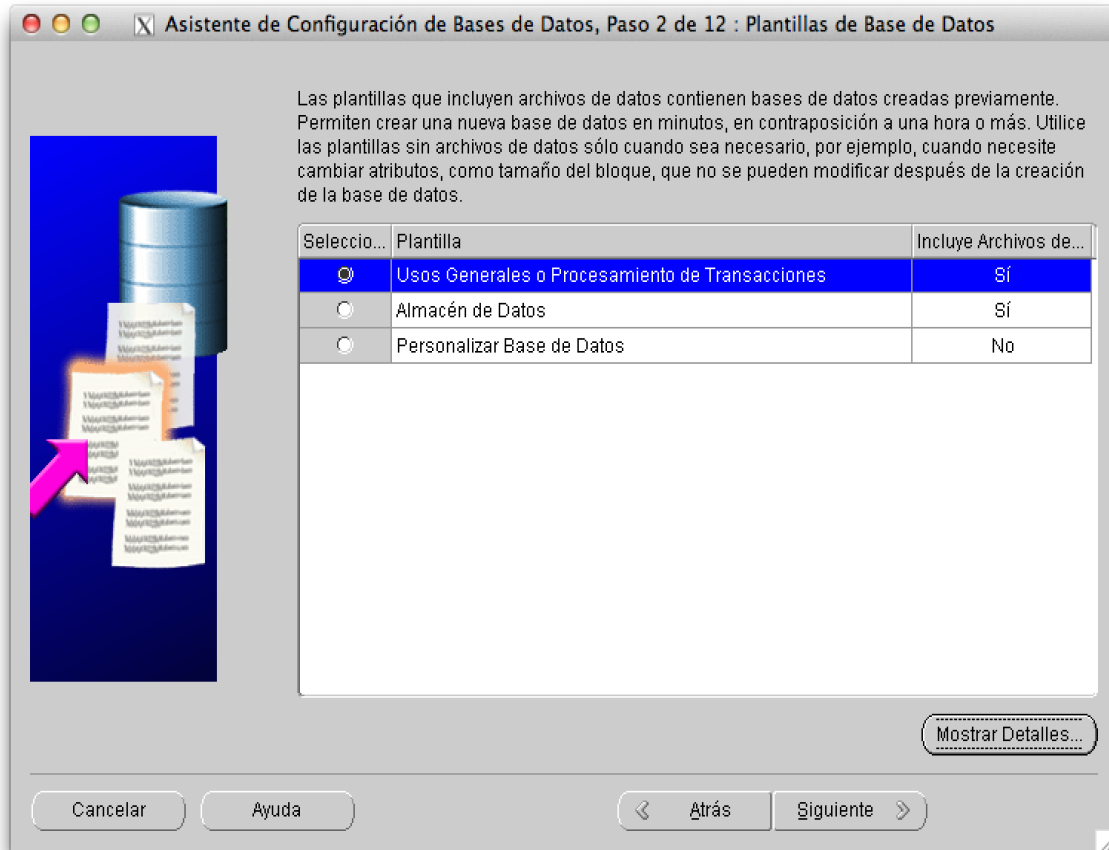


Fig. 4.2.5 Personalizamos la base de datos



Introduciremos un nombre de base de datos global y un SID. Vea figura 4.2.6.

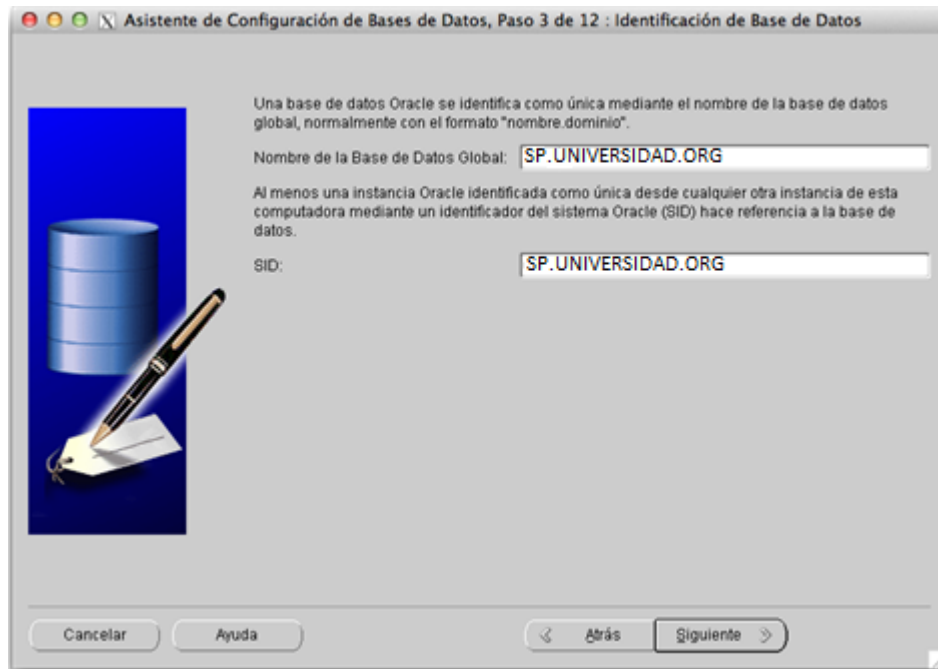


Fig. 4.2.6 Nombramos nuestra base de datos



Quitaremos la marca de selección para "Configurar Enterprise Manager" y "Activar Tareas de Mantenimiento Automáticas" de la pestaña correspondiente. Vea figura 4.2.7.

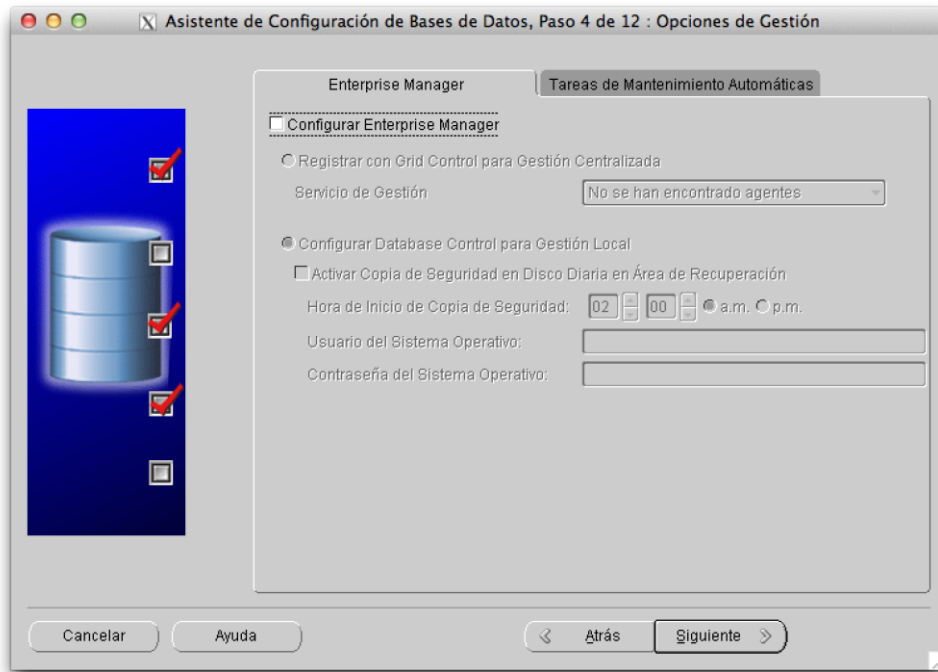


Fig. 4.2.7 Quitamos selecciones por default



Después elegiremos la opción “Usar la Misma Contraseña Administrativa para Todas las cuentas” e introducimos la contraseña para nuestra base de datos. Vea figura 4.2.8.

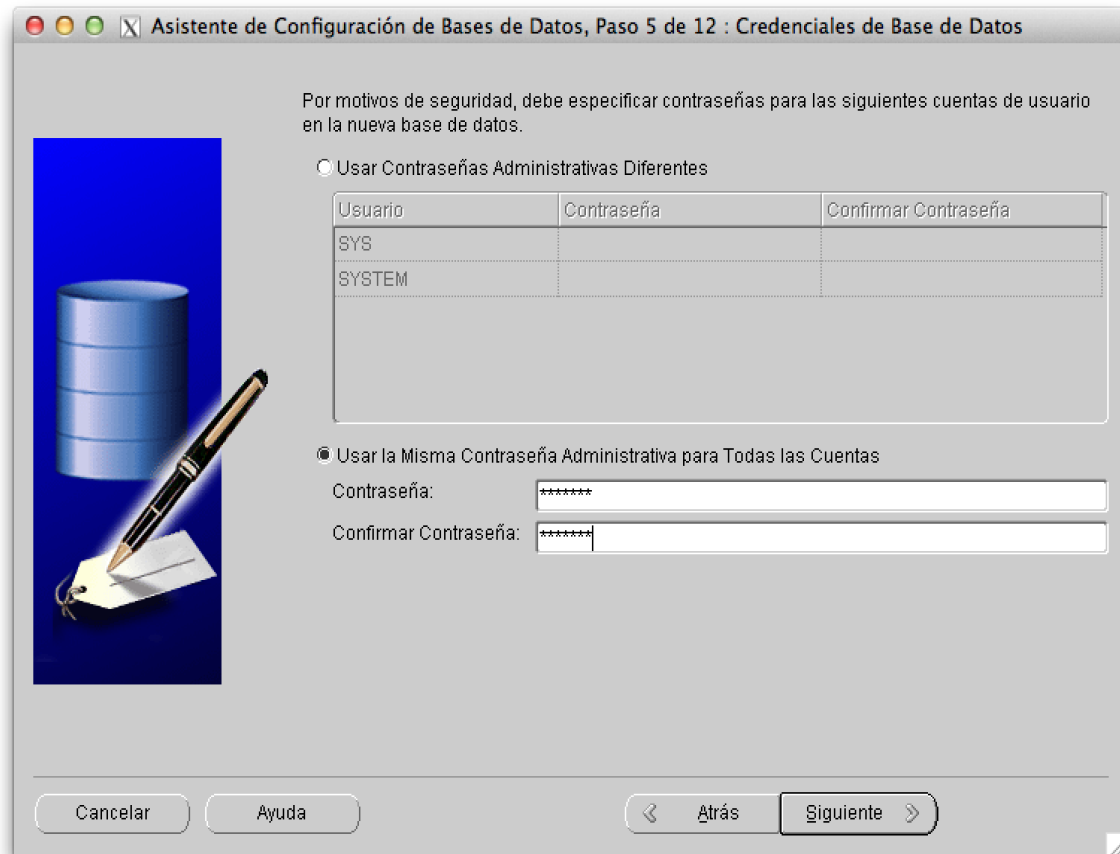


Fig. 4.2.8 Definimos la contraseña de la base de datos



En el caso del tipo de almacenamiento elegiremos “Sistema de Archivos”, “Usar Ubicación Común para todos los archivos de Base de datos” e introducimos la ruta de nuestra base de datos. Vea figura 4.2.9.

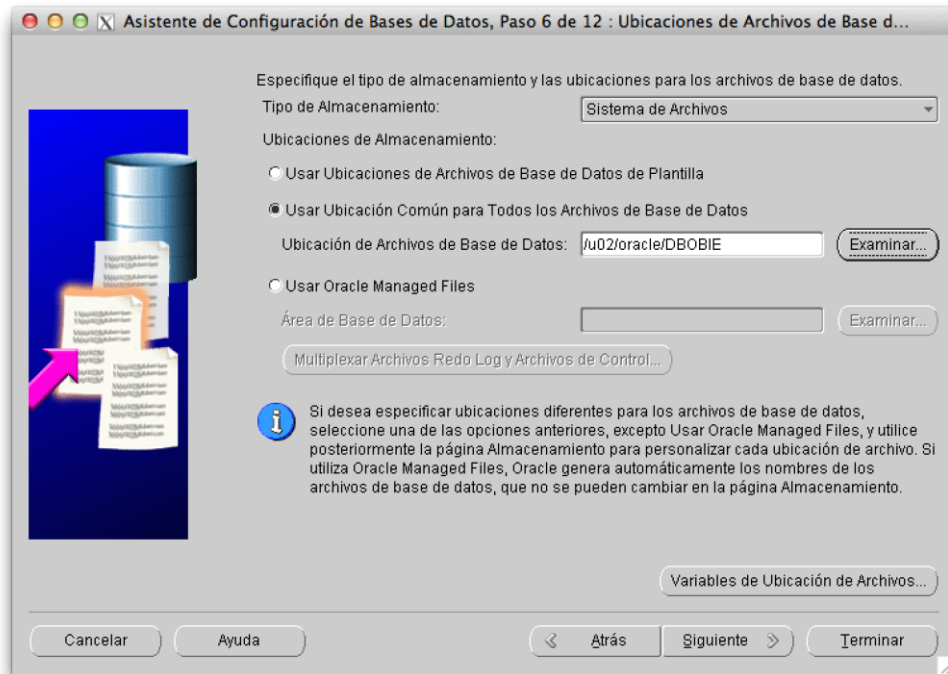


Fig. 4.2.9 Ubicación de archivos



También quitaremos la marca de selección para “Especificar Área de Recuperación Rápida” y “Activar Archivado”. Vea figura 4.2.10.

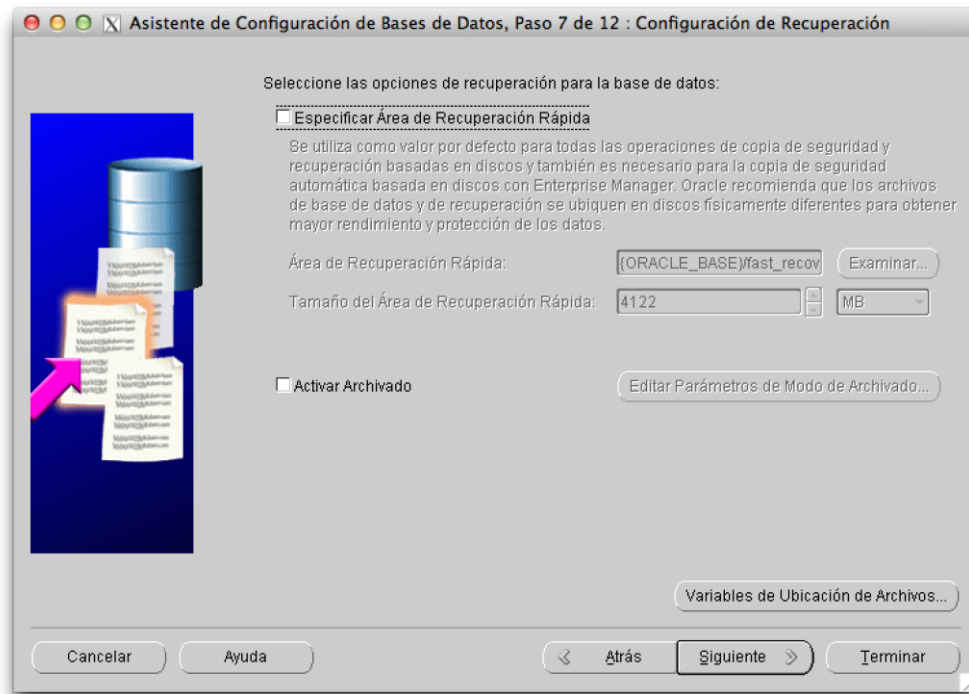


Fig. 4.2.10 Configuración de recuperación



Seleccionamos las opciones “Oracle text”, “Oracle OLAP”, “Oracle Warehouse Builder” y “No hay ningún script para ejecutar”. Vea figura 4.2.11.

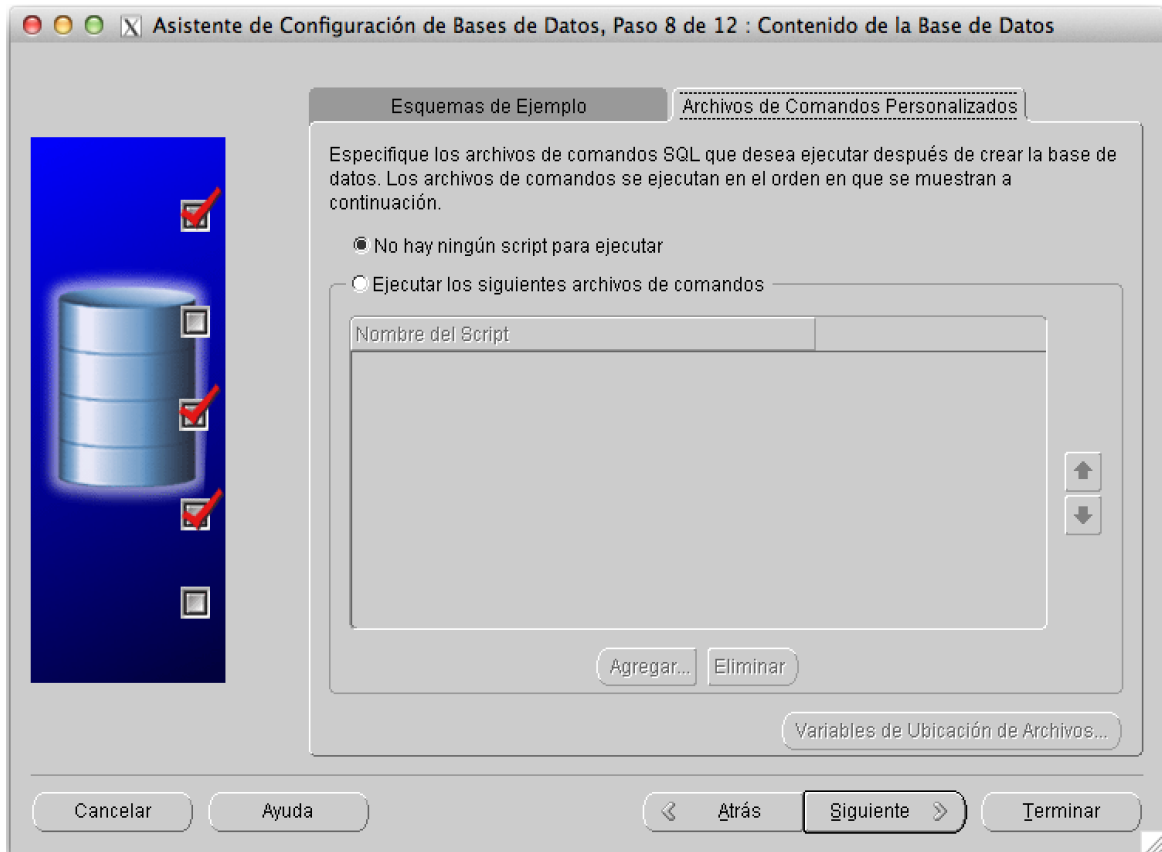


Fig. 4.2.11 Ningún script para ejecutar

Como parámetros de inicialización usaremos (vea figura 4.2.12):

- Memoria
 - Tamaño de Memoria (SGA y PGA): 3072 MB
 - Seleccionamos “Usar Gestión Automática de Memoria”
- Tamaño, editamos
 - Db_block_size = 32768



- Processes = 1000
- Open_cursors= 5000
- Juego de Caracteres
 - Seleccionamos “Usar Unicode (AL32UTF) “
 - Idioma: Inglés
 - Territorio: Estados Unidos
- Modo de Conexión
 - Dejamos valores por default.

Los parámetros processes y open_cursors se configuraron con valores iniciales, el parámetro de juego de caracteres e idioma se configuró por los tipos de caracteres que se iban almacenar en la base de datos y el db_block_size se seleccionó un valor holgado, porque un cambio en este parámetro se necesita recrear la base de datos, estos parámetros se ajustarán de acuerdo a la carga del sistema.

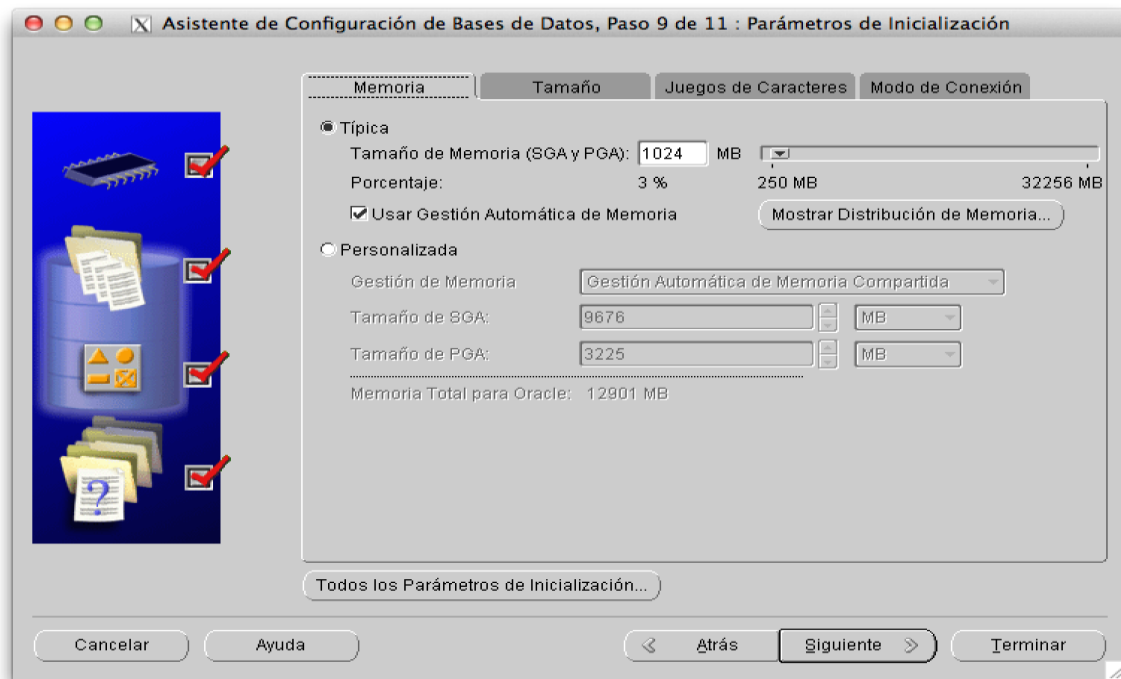


Fig. 4.2.12 Parámetros de inicialización



Para concluir el proceso, modificaremos el tablespaces de nuestra base; usando el sqlplus ejecutamos los comandos:

- CREATE TABLESPACE TBS_DATOS DATAFILE '/u02/oracle/SPPROD/datos01.dbf' SIZE 6144M REUSE AUTOEXTEND ON NEXT 1024K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL;
- CREATE TABLESPACE TBS_INDEX DATAFILE '/u02/oracle/SPPROD/indx01.dbf' SIZE 6144M REUSE AUTOEXTEND ON NEXT 1024K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL;

Una vez que hemos instalado el manejador de base de datos, procedemos a ejecutar el script para la creación del esquema [tablas, stored procedure (SP), etc.] que completará la base de datos. En nuestro caso, el script se verá de la siguiente manera:

```
-----  
-- DDL for Table CAT_APLICACIONES  
-----  
  
CREATE TABLE "SAU"."CAT_APLICACIONES"  
  ( "CVE_APP" NUMBER(38,0),  
    "NOM_APP" VARCHAR2(50 BYTE),  
    "DSC_APP" VARCHAR2(50 BYTE)  
  ) SEGMENT CREATION IMMEDIATE  
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  
  NOCOMPRESS LOGGING  
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
  TABLESPACE "TBS_DATOS" ;  
-----  
-- DDL for Table CAT_COMPONENTES  
-----  
  
CREATE TABLE "SAU"."CAT_COMPONENTES"  
  ( "CVE_COMPTE" NUMBER(38,0),  
    "CVE_APP" NUMBER(38,0),  
    "NOM_COMPTE" VARCHAR2(50 BYTE),  
    "DSC_COMPTE" VARCHAR2(200 BYTE)
```



```

) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;
-----
-- DDL for Table CAT_ESTADOS_USUARIOS
-----

CREATE TABLE "SAU"."CAT_ESTADOS_USUARIOS"
( "CVE_EDO_USUARIO" NUMBER(2,0),
  "NOM_EDO_USUARIO" VARCHAR2(50 BYTE),
  "DESC_EDO_USUARIO" VARCHAR2(50 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;
-----
-- DDL for Table CAT_PRIVILEGIOS
-----

CREATE TABLE "SAU"."CAT_PRIVILEGIOS"
( "CVE_PRIV" NUMBER(38,0),
  "NOM_PRIV" VARCHAR2(50 BYTE),
  "DSC_PRIV" VARCHAR2(50 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;
-----
-- DDL for Table CAT_PRIVILEGIOS_COMPTES
-----

CREATE TABLE "SAU"."CAT_PRIVILEGIOS_COMPTES"
( "CVE_PRIV" NUMBER(38,0),
  "CVE_COMPTE" NUMBER(38,0)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "TBS_DATOS" ;
-----
-- DDL for Table CAT_TIPOS_USUARIOS
-----
```



```
CREATE TABLE "SAU"."CAT_TIPOS_USUARIOS"
( "CVE_TIPO_USUARIO" NUMBER(8,0),
  "NOM_TIPO_USUARIO" VARCHAR2(200 BYTE),
  "DESC_TIPO_USUARIO" VARCHAR2(100 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;
```

-- DDL for Table SEG_ROLES

```
CREATE TABLE "SAU"."SEG_ROLES"
( "CVE_ROL" NUMBER(38,0),
  "NOM_ROL" VARCHAR2(50 BYTE),
  "CVE_EDO_REG" NUMBER(1,0),
  "DESCRIPCION" VARCHAR2(200 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;
```

-- DDL for Table SEG_ROLES_PRIVILEGIOS

```
CREATE TABLE "SAU"."SEG_ROLES_PRIVILEGIOS"
( "CVE_PRIV" NUMBER(38,0),
  "CVE_ROL" NUMBER(38,0),
  "CVE_COMPTE" NUMBER(38,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;
```

-- DDL for Table SEG_USUARIOS

```
CREATE TABLE "SAU"."SEG_USUARIOS"
( "CVE_USUARIO" NUMBER(8,0),
  "NOM_USUARIO" VARCHAR2(100 BYTE),
  "CONTRASENA" VARCHAR2(1000 BYTE),
  "EMAIL" VARCHAR2(50 BYTE),
  "CVE_TIPO_USUARIO" NUMBER(8,0),
```




```
"CVE_EDO_USUARIO" NUMBER(2,0),
"FECHA_VIGEN" DATE,
"CVE_EDO_REG" NUMBER(1,0),
"MATRI_NOEMP" VARCHAR2(20 BYTE),
"CVE_POSGRADO" NUMBER(8,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;

-----
-- DDL for Table SEG_USUARIOS_PRIVILEGIOS
-----

CREATE TABLE "SAU"."SEG_USUARIOS_PRIVILEGIOS"
( "CVE_USUARIO" NUMBER(38,0),
  "CVE_PRIV" NUMBER(15,4),
  "CVE_COMPTE" NUMBER(15,4)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;

-----
-- DDL for Table SEG_USUARIOS_ROLES
-----

CREATE TABLE "SAU"."SEG_USUARIOS_ROLES"
( "CVE_USUARIO" NUMBER(38,0),
  "CVE_ROL" NUMBER(38,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_DATOS" ;

-----
-- DDL for Index PK_APLICACIONES
-----

CREATE UNIQUE INDEX "SAU"."PK_APLICACIONES" ON "SAU"."CAT_APLICACIONES"
("CVE_APP")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_INDEX" ;
```



CAPÍTULO IV



-- DDL for Index PK_COMPONENTES

```
CREATE UNIQUE INDEX "SAU"."PK_COMPONENTES" ON "SAU"."CAT_COMPONENTES"  
("CVE_COMPTE")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "TBS_INDEX" ;
```

-- DDL for Index PK_ESTADO_USUARIO

```
CREATE UNIQUE INDEX "SAU"."PK_ESTADO_USUARIO" ON  
"SAU"."CAT_ESTADOS_USUARIOS" ("CVE_EDO_USUARIO")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "TBS_INDEX" ;
```

-- DDL for Index PK_PRIVILEGIOS

```
CREATE UNIQUE INDEX "SAU"."PK_PRIVILEGIOS" ON "SAU"."CAT_PRIVILEGIOS"  
("CVE_PRIV")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "TBS_INDEX" ;
```

-- DDL for Index PK_TIPO_USUARIO

```
CREATE UNIQUE INDEX "SAU"."PK_TIPO_USUARIO" ON  
"SAU"."CAT_TIPOS_USUARIOS" ("CVE_TIPO_USUARIO")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
TABLESPACE "TBS_INDEX" ;
```

-- DDL for Index PK_ROLES

```
CREATE UNIQUE INDEX "SAU"."PK_ROLES" ON "SAU"."SEG_ROLES" ("CVE_ROL")  
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```



```
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_INDEX" ;
-----
-- DDL for Index PK_SEG_ROLES_PRIVILEGIOS
-----

CREATE UNIQUE INDEX "SAU"."PK_SEG_ROLES_PRIVILEGIOS" ON
"SAU"."SEG_ROLES_PRIVILEGIOS" ("CVE_PRIV", "CVE_ROL", "CVE_COMPTE")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_INDEX" ;
-----
-- DDL for Index PK_USUARIOS
-----

CREATE UNIQUE INDEX "SAU"."PK_USUARIOS" ON "SAU"."SEG_USUARIOS"
("CVE_USUARIO")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_INDEX" ;
-----
-- DDL for Index PK_SEG_USUARIOS_PRIVILEGIOS
-----

CREATE UNIQUE INDEX "SAU"."PK_SEG_USUARIOS_PRIVILEGIOS" ON
"SAU"."SEG_USUARIOS_PRIVILEGIOS" ("CVE_USUARIO", "CVE_PRIV",
"CVE_COMPTE")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_INDEX" ;
-----
-- DDL for Index PK_USUARIOS_ROLES
-----

CREATE UNIQUE INDEX "SAU"."PK_USUARIOS_ROLES" ON
"SAU"."SEG_USUARIOS_ROLES" ("CVE_USUARIO", "CVE_ROL")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TBS_INDEX" ;
-----
-- DDL for View V_USUARIOS_ROLES
-----
```



```
CREATE OR REPLACE FORCE VIEW "SAU"."V_USUARIOS_ROLES" ("CVE_USUARIO",
"CVE_ROL", "NOM_ROL") AS
  select sur.cve_usuario, sur.cve_rol, sr.NOM_ROL from seg_usuarios_rols
sur, seg_roles sr
where sur.cve_rol = sr.CVE_ROL;
-----
-- DDL for Procedure RESET_SEQ
-----
set define off;

CREATE OR REPLACE PROCEDURE "SP"."RESET_SEQ" ( p_seq_name in varchar2 )
is
  l_val number;
begin
  execute immediate
  'select ' || p_seq_name || '.nextval from dual' INTO l_val;

  execute immediate
  'alter sequence ' || p_seq_name || ' increment by -' || l_val ||
  ' minvalue 0';

  execute immediate
  'select ' || p_seq_name || '.nextval from dual' INTO l_val;

  execute immediate
  'alter sequence ' || p_seq_name || ' increment by 1 minvalue 0';
end;

/
-----
-- DDL for Procedure RESTRUC_POSGRADO
-----
set define off;

CREATE OR REPLACE PROCEDURE "SP"."RESTRUC_POSGRADO"
( p_cve_posgrado in sp_posgrados.cve_posgrado%TYPE, p_bander out number
)
is

TYPE t_posgrado is table of sp_posgrados%ROWTYPE;
TYPE t_planes is table of sp_planes_estudios%ROWTYPE;
v_posgrados t_posgrado;
v_planes t_planes;
seq_posgrado number :=0;
seq_plan number :=0;
begin
p_bander := 0;
select * bulk collect into v_posgrados from sp_posgrados where
cve_posgrado = p_cve_posgrado;

  for i in 1..v_posgrados.count
  loop
```



DISEÑO Y CONSTRUCCIÓN DEL SISTEMA WEB



```
select seq_sp_posgrados.nextval into seq_posgrado from dual;
select seq_sp_planes_estudios.nextval into seq_plan from dual;

if (v_posgrados(i).CVE_EDO_POSG = 2) then
  INSERT INTO "SP"."SP_POSGRADOS" (CVE_POSGRADO, NOM_POSGRADO,
FECHA_ALTA, CVE_ORIENTACION, CVE_GRADO_ACAD, PAG_WEB, CVE_UA_SEDE,
CAMPUS, CVE_COORDINADOR,
CVE_CARACT, CVE_ARE_SNI, ANIO_INGRESO, CVE_DOMICILIO, TELEFONO, EMAIL,
CVE_CAMPO_CO, CVE_MOD_EDU, CVE_LGAC, CVE_EDO_POSG, CVE_EDO_REG,
ANTECEDENTES, FECHA_APROB_REST, CVE_CAMPUS, CVE_UA_ASOCIADA,
NOMBRE_LGAC)
SELECT seq_posgrado as CVE_POSGRADO, NOM_POSGRADO,
SYSDATE AS FECHA_ALTA, CVE_ORIENTACION, CVE_GRADO_ACAD, PAG_WEB,
CVE_UA_SEDE, CAMPUS,
CVE_COORDINADOR, CVE_CARACT, CVE_ARE_SNI, ANIO_INGRESO, CVE_DOMICILIO,
TELEFONO,
EMAIL, CVE_CAMPO_CO, CVE_MOD_EDU, CVE_LGAC, 3 as CVE_EDO_POSG,
CVE_EDO_REG,
ANTECEDENTES, FECHA_APROB_REST, CVE_CAMPUS, CVE_UA_ASOCIADA, NOMBRE_LGAC
FROM sp_posgrados
WHERE CVE_POSGRADO = p_cve_posgrado;
INSERT INTO "SP"."SP_PLANES_ESTUDIOS" (CVE_PLAN, CVE_PLAN_SASE,
DESC_ESTRU, DESC_CONT, EDO_ARTE, PLANT_PROBLE, OBJ_PART, TEND_MER_LAB,
OBJ_GEN, EGRE_COMP, EGRE_CONO, EGRE_HAB, EGRE_VALORES, INGRE_COMP,
INGRE_CONO, INGRE_HAB, INGRE_VALORES, ESTRU_GLOB, ESTRU_CURS,
TAB_UNI_APRE, CVE_POSGRADO, PORCENTAJE_ESTRUCTURA,
PORCENTAJE_JUSTIFICACION, PORCENTAJE_OBJETIVOS,
PORCENTAJE_EGRESO, PORCENTAJE_INGRESO, PORCENTAJE_MAPA,
PORCENTAJE_UNIDAD)
SELECT seq_plan as CVE_PLAN, CVE_PLAN_SASE, DESC_ESTRU, DESC_CONT,
EDO_ARTE, PLANT_PROBLE, OBJ_PART, TEND_MER_LAB,
OBJ_GEN, EGRE_COMP, EGRE_CONO, EGRE_HAB, EGRE_VALORES, INGRE_COMP,
INGRE_CONO, INGRE_HAB, INGRE_VALORES, ESTRU_GLOB, ESTRU_CURS,
TAB_UNI_APRE, seq_posgrado as CVE_POSGRADO, PORCENTAJE_ESTRUCTURA,
PORCENTAJE_JUSTIFICACION, PORCENTAJE_OBJETIVOS,
PORCENTAJE_EGRESO, PORCENTAJE_INGRESO, PORCENTAJE_MAPA, PORCENTAJE_UNIDAD
FROM SP_PLANES_ESTUDIOS
WHERE CVE_POSGRADO = v_posgrados(i).cve_posgrado;
p_bander := 1;
else
p_bander := 0;
end if;
end loop;
```



```
        commit;
end;

/
-----
-- DDL for Procedure SEND_MAIL_UNAM
-----
set define off;

CREATE OR REPLACE PROCEDURE "SP"."SEND_MAIL_UNAM" ( fromm in varchar2,
too in varchar2, sub in varchar2 , mensaje in varchar2 ) is

--fromm varchar2 (300):='consultant@empresa.com';

--too varchar2 (300) :='consultant@empresa.com';

objConnection utl_smtp.connection;

BEGIN

execute      immediate      'alter      session      set
smtp_out_server='||''192.168.1.0:25''';

objConnection := UTL_smtp.open_connection('192.168.1.0','25');

UTL_smtp.helo(objConnection, '192.168.1.0');

-- Abre conexión con un Servidor SMTP, puerta estándar es 25
-- v_conn := utl_smtp.open_connection (v_smtp,25);
-- utl_smtp.helo (objConnection, v_smtp);
-- utl_smtp.command(objConnection, 'STARTTLS');

-- utl_smtp.command (objConnection, 'AUTH LOGIN');

-- utl_smtp.command (objConnection,
UTL_RAW.cast_to_varchar2(utl_encode.base64_encode
(utl_raw.cast_to_raw('seguimiento'))));
--utl_smtp.command (objConnection,
UTL_RAW.cast_to_varchar2(utl_encode.base64_encode
(utl_raw.cast_to_raw('U4nam2015'))));

UTL_smtp.mail(objConnection, fromm);

UTL_smtp.rcpt(objConnection, too);

UTL_smtp.open_data(objConnection);

UTL_smtp.write_data(objConnection, 'From: '||fromm || UTL_tcp.CRLF);
```



```
UTL_smtp.write_data(objConnection, 'To: ' || too || UTL_tcp.CRLF);
UTL_smtp.write_data(objConnection, 'Subject: ' || sub || UTL_tcp.CRLF);
UTL_smtp.write_data(objConnection, 'MIME-Version: ' || '1.0' ||
UTL_tcp.CRLF);
UTL_smtp.write_data(objConnection, 'Content-Type: ' || 'text/html;');
UTL_smtp.write_data(objConnection, 'Content-Transfer-Encoding: ' ||
'"8Bit"' || UTL_tcp.CRLF);
UTL_smtp.write_data(objConnection, UTL_tcp.CRLF);
UTL_smtp.write_data(objConnection, UTL_tcp.CRLF || '<HTML>');
UTL_smtp.write_data(objConnection, UTL_tcp.CRLF || '<BODY>');
UTL_smtp.write_data(objConnection, UTL_tcp.CRLF || '<FONT          COLOR="red"
FACE="Courier New">' || mensaje || '</FONT>');
UTL_smtp.write_data(objConnection, UTL_tcp.CRLF || '</BODY>');
UTL_smtp.write_data(objConnection, UTL_tcp.CRLF || '</HTML>');
UTL_smtp.close_data(objConnection);

UTL_smtp.quit(objConnection);
EXCEPTION
WHEN utl_smtp.transient_error OR utl_smtp.permanent_error THEN
utl_smtp.quit(objConnection);
raise_application_error(-20000,
'A tentativa de enviar o e-mail gerou o seguinte erro: ' || sqlerrm);
WHEN OTHERS THEN
RAISE;
END;
```



Al finalizar tendremos un árbol de tablas y SP como el que se muestra en las figuras 4.2.13 y 4.2.14.

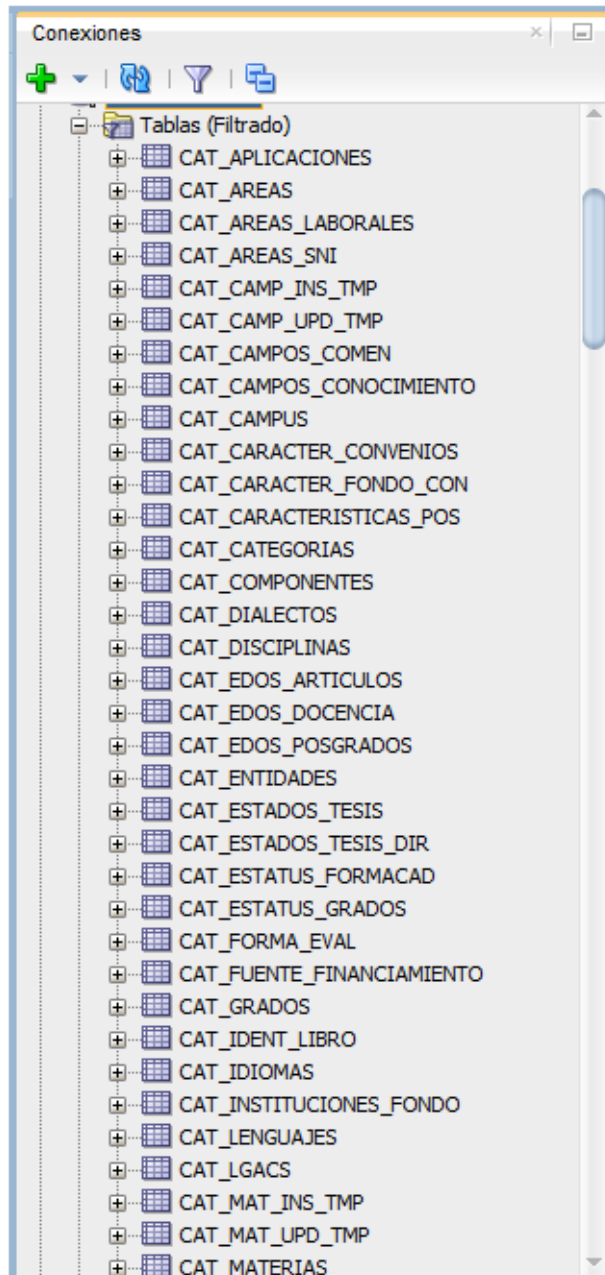


Fig. 4.2.13 Árbol de tablas

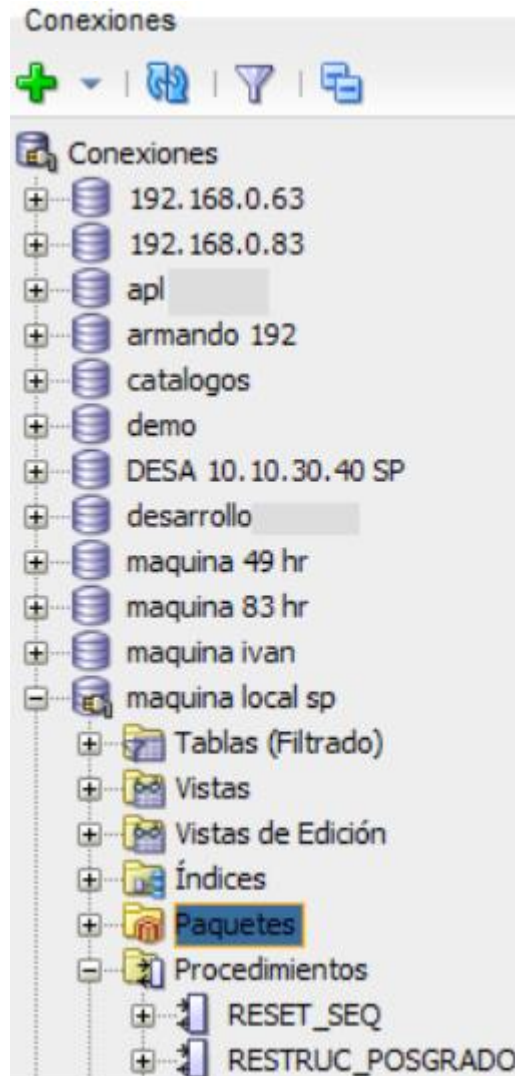


Fig. 4.2.14 Árbol de Stored Procedure



4.3 Construcción y diseño del sistema web.

Introducción.

El sistema Web o (Front-End) de productividad del posgrado se desarrolló a través de un entorno Web visible al usuario, el cual fue diseñado y construido para un fácil manejo e interacción.

La interfaz de usuario se realizó de un modo muy limpio y buscando optimizar los espacios para menús y contenidos como se planteaba desde el apartado 3.4, en el modelo preliminar para el diseño funcional del sistema, y para la distribución de los módulos.

Desde las interfaces de usuario se pueden realizar funciones como son: registrar toda la información de los posgrados, planes de estudios, historia, infraestructura, seguimiento a estudiantes y así como los reportes de los resultados del posgrado, adicionalmente todos los académicos podrán cargar todo su currículum vitae, que permitirá alimentar los reportes de la productividad académica de cada uno de los profesores o planta académica de cada uno de los posgrado.

El sistema de productividad del Posgrado fue construido en base a dos subsistemas (CVU y Posgrados), cada subsistema es distribuido en dos capas:

- **Capa de Presentación:** La función de la capa de presentación es recibir las peticiones provenientes de la capa cliente, solicitar el procesamiento de información a la capa de negocio y generar las respuestas



correspondientes al usuario. La capa de presentación se desarrolla a través de un módulo Web, el cual representa a una aplicación Web. Un módulo Web se crea ensamblando, archivos JSF (JavaServer Faces), y contenido estático como, por ejemplo, páginas HTML en una sola unidad desplegable. Los módulos Web se representan físicamente en archivos WAR (Web Archive), que son ficheros basados en el estándar de JEE 6 que son desplegados en JBoss AS 7.

- **Capa de Negocio:** La función de la capa de negocio es implementar las reglas de negocio, el modelo de objetos, servicios y acceso a los datos. Esta capa estará constituida por 2 módulo JAR (Java Archive), estos 2 módulos se almacenan en un archivo EAR (Enterprise Archive), el módulo JAR se crea ensamblando, archivos Java y archivos de propiedades que son ficheros basados en el estándar de JEE 6, que son desplegados en JBoss AS 7.

Los dos subsistemas se implementaron utilizando un patrón MVC (Modelo, Vista, Controlador).

Cada subsistema del sistema de productividad del Posgrado será distribuido en dos paquetes de archivos de despliegues (véase la figura 4.3.1):

- Archivo EAR (Enterprise Archive) que contiene la capa de negocio, el modelo de objetos, Objetos de Servicios y Acceso a Datos.
- Archivo WAR (Web Archive) que contiene la capa de presentación JSF (XHTML), HTML 5 y controladores hacia las peticiones entre la capa de presentación y la capa de negocio. Cada módulo Web debe contener un descriptor denominado “web.xml”, que declara el contenido del módulo.

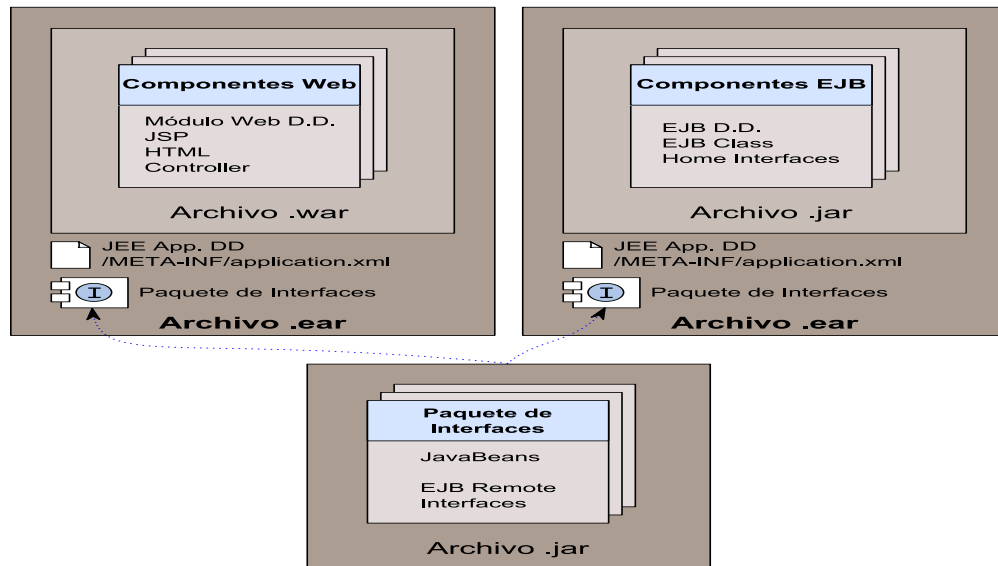


Figura 4.3.1 Archivos de paquetes de despliegues

Instalación de Servidor de Aplicaciones JBOSS 7.1.

Prerrequisitos para la Instalación.

Se realizaron las siguientes validaciones de los prerrequisitos mínimos para instalar JBoss AS 7.1:

- Validación de que el servidor tenga mínimo 4 GB de memoria RAM, es un valor necesario para la configuración de JBoss 7.1.
- Validación de que tenga mínimo un CPU Dual Core.
- Validación en el servidor que tenga mínimo 4 GB de memoria SWAP.
- Validación de la arquitectura del servidor Centos 6.7 x86-64
- Validación de Espacio en disco de 5 GB para el servidor de aplicaciones y 20 GB para el almacenamiento de Archivos de la aplicación.
- Validación de parámetros de kernel del sistema operativo.



Instalación de JDK 7u55.

Después de haber configurado los parámetros de sistema operativo y haber validado que se cumpla con todos los requerimientos y tener el software listo para su uso en el servidor procedimos a la instalación del software.

Accedemos al sistema operativo con el usuario “jboss” y nos ubicamos en la carpeta “/software” la cual contiene el instalador.

- Desempaquetamos el software del jdk, véase la figura 4.3.2, ejecutando los siguientes comandos:
 - \$cd /u03
 - \$tar xvfz jdk-7u55-linux-x64.tar.gz

```
File Edit View Window Help
Quick Connect Profiles
# - nice - max nice priority allowed to raise to values: [-20, 19]
# - rtprio - max realtime priority
#
#<domain> <type> <item> <value>
#
#* soft core 0
#* hard rss 10000
#@student hard nproc 20
#@faculty soft nproc 20
#@faculty hard nproc 50
#ftp hard nproc 0
#@student - maxlogins 4
# End of file
jboss soft nofile 4096
jboss hard nofile 65536
jboss soft nproc 2047
jboss hard nproc 16384
[jboss@posivdesback ~]$ cd /u03
[jboss@posivdesback u03]$ ls
jboss-as-7.1.1.Final jdk1.7.0_55 lost+found
jboss-as-7.1.1.Final.tar jdk-7u55-linux-x64.tar ojdbc6.jar
[jboss@posivdesback u03]$ tar -xvf jdk-7u55-linux-x64.tar
```

Figura 4.3.2 Desempaquetar JDK



- Configuramos las variables de ambientes en el profile del archivo, véase la figura 4.3.3, realizando las siguientes instrucciones:
 - Agregamos la variable JAVA_HOME=/u03/jdk1.7.0_55
 - Actualizamos la variable PATH=\$JAVA_HOME/bin:\$PATH
 - Agregamos la línea export JAVA_HOME PATH

```
File Edit View Window Help
Quick Connect Profiles
[jboss@posivback u03]$ cd /software/
[jboss@posivback software]$ ls
cvuEAR.ear          lost+found          seguridadEAR.ear
jboss-as-7.1.1.Final.tar  ojdbc6.jar
jdk-7u55-linux-x64.tar  seguimientoPosgradoEAR.ear
[jboss@posivback software]$ cd /u03
[jboss@posivback u03]$ ls
jboss-as-7.1.1.Final  jdk1.7.0_55
[jboss@posivback u03]$ cd
[jboss@posivback ~]$ more .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

JAVA_HOME=/u03/jdk1.7.0_55
PATH=$JAVA_HOME/bin:$PATH:$HOME/bin
export JAVA_HOME PATH
[jboss@posivback ~]$
```

Figura 4.3.3 Configuración de variables de ambiente

Instalación de JBoss AS 7.1.1⁴.

Accedemos al sistema operativo con el usuario “jboss” y nos ubicamos en la carpeta “/software” la cual contiene el instalador.

⁴ Gilda S., Logue E., Mison D., Ryan D. (2012). JBoss Enterprise Application Platform 6 Installation Guide. https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6/pdf/Installation_Guide/JBoss_Enterprise_Application_Platform-6-Installation_Guide-en-US.pdf



- Se desempaquetó el software de JBoss, véase la figura 4.3.4, realizando las siguientes instrucciones:
 - cd /software
 - \$ tar xvfz jboss-as-7.1.1.Final.tar.gz

```
File Edit View Window Help
SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.

Last login: Tue Sep 22 23:44:19 2015 from 10.10.31.2
[root@posivback ~]#
[root@posivback ~]#
[root@posivback ~]# cd /u03
[root@posivback u03]# cd /software/
[root@posivback software]# ls
cvuEAR.ear          lost+found          seguridadEAR.ear
jboss-as-7.1.1.Final.tar  ojdbc6.jar
jdk-7u55-linux-x64.tar   sequimientaPosgradoEAR.ear
[root@posivback software]# tar xvfz jboss-as-7.1.1.Final.tar.gz
```

Figura 4.3.4 Desempaquetar JBOSS

- Agregamos dentro del archivo /u03/jboss-as-7.1.1.Final/standalone/configuration/standalone-full.xml, los siguientes 2 datasources, configurando la conexión, usuarios y contraseñas de la base de datos, véase la figura 4.3.5:
 - <datasource jndi-name="java:jboss/spDS" >
 - <datasource jndi-name="java:jboss/sauDS" >



```
<password>AdmSau_#</password>
</security>
<timeout>
  <blocking-timeout-millis>5000</blocking-timeout-millis>
  <query-timeout>5</query-timeout>
</timeout>
<statement>
  <prepared-statement-cache-size>25</prepared-statement-ca
che-size>
</statement>
</datasource>
<drivers>
  <driver name="oracle" module="com.oracle">
    <xa-datasource-class>oracle.jdbc.OracleDriver</xa-dataso
urce-class>
  </driver>
  <driver name="h2" module="com.h2database.h2">
    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-dat
asource-class>
  </driver>
</drivers>
</datasources>
</subsystem>
```

Figura 4.3.5 Configuración DataSources

Nota: La configuración inicial del heap de la JVM fue -xmx2048M – Xms2048m y el pool de conexiones fue configurado con un valor inicial de 10 y máximo de 60 conexiones, estos valores se van ir ajustando de acuerdo a la carga del sistema”

- Iniciamos el servidor de aplicaciones con el siguiente comando y validar en la salida que tenga el mensaje “Starting” y los “Deployed”, véase la figura 4.3.6:
 - /u03/jboss-as-7.1.1.Final/bin/standalone.sh -b 192.168.1.86 -c standalone-full.xml



```
[jboss@posivdesfront bin]$ tail -f nohup.out
16:37:22,620 INFO [org.ocpsoft.rewrite.param.DefaultParameterStore] (MSC service thread 1-14) Loaded [1] org.ocpsoft.rewrite.spi.GlobalParameterProvider [org.ocpsoft.rewrite.instance.WildcardParameterProvider<0>]
16:37:22,652 INFO [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-7) Rewrite 2.0.12.Final initialized.
16:37:22,653 INFO [org.jboss.web] (MSC service thread 1-7) JBAS018210: Registering web context: /seguimientoPosgradoFront
16:37:22,685 INFO [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-14) Rewrite 2.0.12.Final initialized.
16:37:22,686 INFO [org.jboss.web] (MSC service thread 1-14) JBAS018210: Registering web context: /cvuFront
16:37:22,687 INFO [org.jboss.as] (MSC service thread 1-14) JBAS015951: Admin console listening on http://192.168.130.25:9990
16:37:22,688 INFO [org.jboss.as] (MSC service thread 1-14) JBAS015874: JBoss AS 7.1.1.Final "Brontes" started in 10383ms - Started 1802 of 1890 services (87 services are passive or on-demand)
16:37:22,727 INFO [org.jboss.as.server] (DeploymentScanner-threads - 2) JBAS018559: Deployed "cvuFront.war"
16:37:22,727 INFO [org.jboss.as.server] (DeploymentScanner-threads - 2) JBAS018559: Deployed "seguimientoPosgradoFront.war"
16:37:22,727 INFO [org.jboss.as.server] (DeploymentScanner-threads - 2) JBAS018559: Deployed "seguridadFront.war"
```

Figura 4.3.6 Inicio de JBoss.

Interfaces de Usuario.

A continuación se describirán las funciones con las que cuenta la interfaz del usuario:

En la primera página de inicio del sistema, el sistema requiere que seleccione el usuario a que subsistema requiere entrar, puede entrar a los subsistemas de CVU y Posgrado, como se muestra en la figura 4.3.7.



UNIVERSIDAD U

Dirección General de Posgrado e Investigación

Sistema de Productividad del Posgrado



Figura 4.3.7 Página de Inicio



Ingresa a tu cuenta del CVU

Tipo de Acceso
 Control Profesor Alumno

Usuario

Contraseña

Iniciar Sesión

[Regístrate](#)

Copyright © 2016 Universidad,
Av. Uni, C.P. 11111, Estado, México. Todos los derechos reservados.

Figura 4.3.8 Página de autenticación del subsistema CVU



En la página de autenticación de CVU, el sistema requiere que el usuario introduzca sus datos de autenticación, usuario, contraseña y seleccione el tipo de acceso al sistema, como se muestra en la figura 4.3.8.

En la página de Bienvenida del sistema de productividad del posgrado en el subsistema de CVU, en la sección de contenido muestra una leyenda de bienvenida, en la sección de encabezado muestra el logo, nombre del usuario y botón de salida del sistema, en la sección de Pie de página muestra domicilio de la universidad y en la sección del menú lateral muestra todos los módulos que permitirán registrar los datos del CVU como son datos generales, formación académica, experiencia laboral, estancias de investigación, proyectos de investigación, docencia, distinciones, Libros y Artículos publicados, como se muestra en la figura 4.3.9.

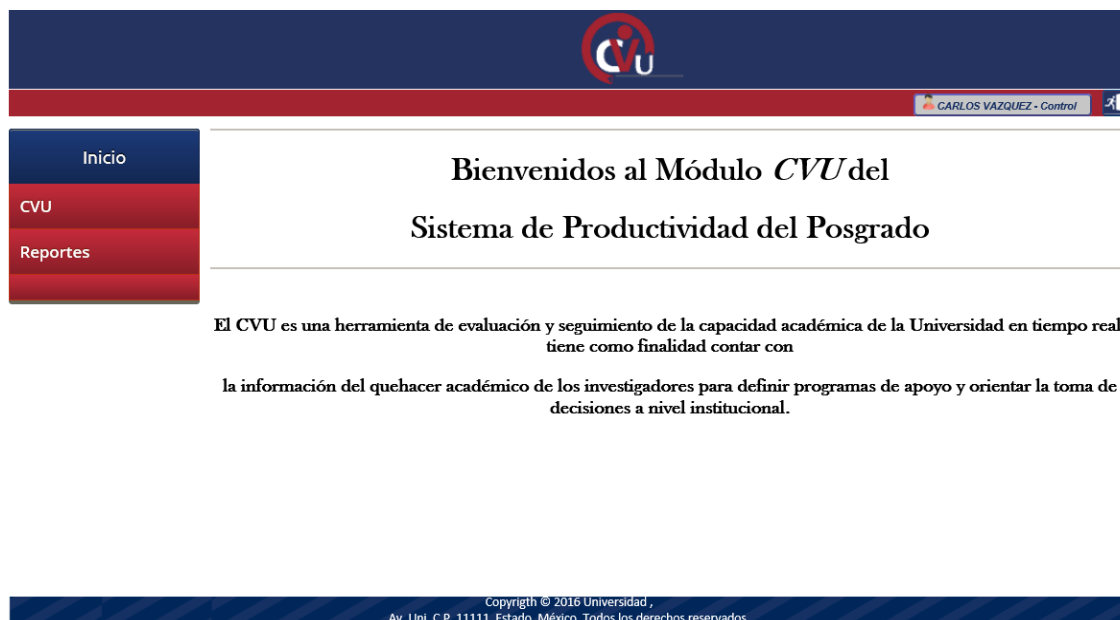


Figura 4.3.9. Página de Bienvenida del subsistema CVU



En la página de Listado de Libros del sistema de productividad del posgrado en el subsistema de CVU, En la sección de contenido mostrará el listado de libros con la información general del libro, el botón para poder registrar nuevos libros y el buscador sobre el listado mostrado, entre los datos mostrados se encuentran el título del libro, ISBN, Editorial, Edición, Tipo y Operaciones o acciones que se pueden realizar como son Consulta detalle, Actualización y Eliminación, como se muestra en la figura 4.3.10.



Figura 4.3.10 Página de listado de Libros

En la página de alta de Libros del sistema de productividad del posgrado en el subsistema de CVU, El usuario debe introducir los datos del libro, tal como se muestra en la figura 4.3.11.



Inicio

CVU

Reportes

Agregar Libro

Usuario CVU

Nombre: CARLOS VAZQUEZ RIVERA

Libro:

ISBN: Identificador Libro: Selecciona

Título del Libro:

Editorial: Edición: Año Publicación:

Volumen: Paginas: Tiraje:

Idioma: Selecciona Área: Selecciona

Disciplina: Selecciona

Palabra Clave 1: Palabra Clave 2: Palabra Clave 3:

Liga:

Carátula:

Resumen del libro:

Carta de aceptación:

Figura 4.3.11 Página de Alta de Libros

Sistema de Productividad del Posgrado

Sistema de Productividad del Posgrado

SPPU

Ingresa a tu cuenta del SPP

Tipo de Acceso

Control Profesor Alumno

Usuario

Contraseña

[Regístrate](#)

Figura 4.3.12 Página de autenticación del subsistema de Posgrado.



En la página de autenticación del subsistema del posgrado, el sistema requiere que el usuario introduzca sus datos de autenticación, usuario, contraseña y seleccione el tipo de acceso al sistema, como se muestra en la figura 4.3.12.

En la página de Bienvenida del sistema de productividad del posgrado en el subsistema de Posgrados, en la sección de contenido muestra una leyenda de bienvenida, en la sección de encabezado de la página muestra el logo, nombre del usuario y botón de salida del sistema, en la sección de Pie de página muestra domicilio de la universidad y la sección de menú lateral mostrará todos los módulos para poder registrar los datos del Posgrado como son Posgrados, estructura y personal académico, estudiantes, infraestructura, Resultados y vinculación y reportes, como se muestra en la figura 4.3.13.

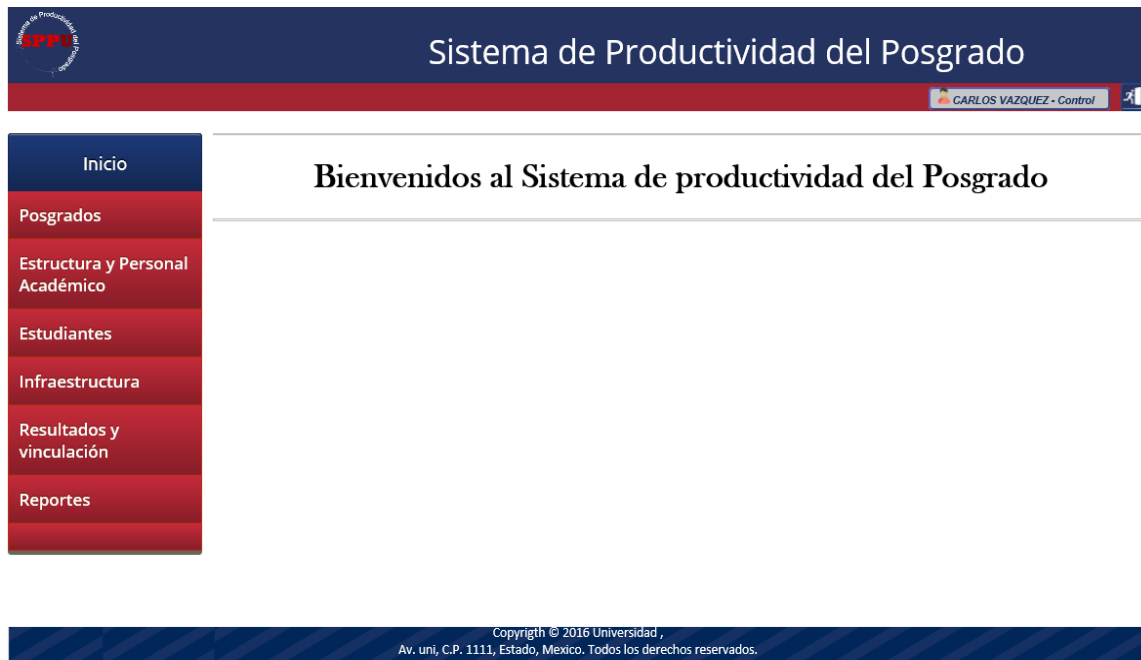


Figura 4.3.13 Página de Bienvenida del subsistema Posgrado



En la página de Listado de Posgrados del sistema de productividad del posgrado, se en lista los posgrados de la universidad, mostrando los datos del nombre del posgrado, Estado del posgrado, Nivel PNPC, Fecha de creación, página web y operaciones o acciones cómo son consulta detalle, Actualización, restructuración, alta y suspensión, como se muestra en la figura 4.3.14.

Posgrado	Estado	Nivel PNPC	Fecha de Creación	Página Web	Acciones
Doctorado en Ciencias Biomédicas	Aprobado	Reciente creacion	10/03/2009	http://doctoradobiomedicas.unam.mx	
Doctorado en Ciencias con Especialidad en Matemática Educativa	Reestructurado	Reciente creacion	3/10/2005	http://doctoradome-unam.mx/	
Maestría en Ciencias Biomédicas	En creación	Consolidado	1/08/2006	http://maestriabiomedicas.com.mx	
Maestría en Ciencias Biomédicas-Clic	Aprobado	Consolidado	12/01/2016	http://www.unam.mx	
MAESTRÍA EN CIENCIAS: GESTIÓN SUSTENTABLE DEL TURISMO	En creación	Consolidado	4/02/2014	No Disponible	
Maestría en Docencia del Idioma Ingles	En creación	Consolidado	7/10/2009	No Disponible	
MAESTRÍA EN SALUD PÚBLICA	En creación	No pertenece a pnpc	20/10/2014	http://unam.edu.mx/	

Figura 4.3.14 Página de Listado de Posgrados



CAPÍTULO IV



En la página de Actualización del Posgrado del sistema de productividad del posgrado, muestra un formulario que permite actualizar los datos principales del posgrado, como se muestra en la figura 4.3.15.

The screenshot displays the 'Sistema de Productividad del Posgrado' interface. On the left is a navigation menu with options: Inicio, Posgrados, Estructura y Personal Académico, Estudiantes, Infraestructura, Resultados y vinculación, and Reportes. The main content area is titled 'Actualizar Datos Generales' and contains a form with two tabs: 'Datos Generales' (active) and 'Historia'. The form fields are as follows:

Posgrado:	Doctorado en Ciencias Biomédicas		
Fecha de Creación:	10/03/2009	Grado Académico:	Doctorado
Campus:	Escuelas zona centro	Unidad Académica Sede:	10102 Unidad académica de ciencias quimic
Coordinador:	MICHELLE GEORGINA RODRIGUEZ BAILC	Unidad Académica Participante:	-----Selecciona-----
Clave de Profesiones:			
Nivel PNP:	Reciente creacion	Alta PNP:	mar 10, 2009
Vigencia PNP:	mar 10, 2018	Área SNI:	Área III Medicina y Ciencias de la Salud
Orientación:	Profesional		
PDF plan:	<input type="text"/> Browse...	Página Web:	doctoradobiomedicas.unam.mx
Teléfono:	47 19310 ext. 3601	Correo Electrónico:	iprojas@yahoo.com

Figura 4.3.15 Página de Actualización del Posgrado



4.4 Aplicación de pruebas y mantenimiento.

Pruebas.

La prueba es un proceso autónomo, y el número de tipos diferentes de pruebas varía tanto como los diferentes enfoques de desarrollo, durante mucho tiempo la única defensa contra los errores de programación fueron el cuidadoso diseño y la inteligencia natural del programador, actualmente se cuenta con técnicas de diseño que reducen el número de errores iniciales del código.

El sistema o software se prueba para descubrir errores cometidos al realizar el diseño y construcción, el jefe del proyecto, los ingenieros del software o especialistas en pruebas son quienes desarrollan la estrategia para la prueba del sistema, esta actividad requiere una mayor cantidad de esfuerzo dedicado al proyecto, si es realizada sin ningún plan, se desperdicia tiempo, se dedica un esfuerzo innecesario y es posible que no se detecten errores, por lo que es necesario establecer una estrategia sistemática para probar el sistema.

En las primeras etapas la prueba se concentra en un solo componente o en un grupo pequeño de componentes relacionados, y se aplica para descubrir errores en la lógica de datos y del procesamiento que se ha encapsulado en el componente. Una vez que se prueban los componentes, deben integrarse hasta que todo el sistema se haya construido, se ejecuta una serie de pruebas de alto nivel para descubrir errores en la satisfacción de los requisitos del cliente, a medida que se van descubriendo los errores deben diagnosticarse y corregirse empleando un proceso de depuración.



Un plan y un procedimiento de pruebas efectivos llevarán a la construcción ordenada del sistema y al descubrimiento de errores en cada etapa del proceso de construcción.

Tipos de Pruebas.

Existen características genéricas en las pruebas de software que se han plasmado en diversos documentos, tales como: un equipo de software debe efectuar revisiones técnicas formales y efectivas, la prueba comienza al nivel de componentes y trabaja hacia la integración de todo el sistema, la prueba la dirige el desarrollador del software, la prueba y depuración son actividades diferentes, pero la segunda se incluye en cualquier estrategia de prueba. Se describen algunos tipos de pruebas.

- Caja negra.

Las pruebas de la caja negra tratan de encontrar errores en las siguientes categorías:

- funciones incorrectas o faltantes
- errores de interfaz
- errores en estructuras de datos o en acceso a bases de datos externas
- errores de comportamiento o desempeño
- errores de inicialización y término.



- Caja blanca.

La prueba de caja blanca, en ocasiones llamada prueba de caja de cristal, es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. El ingeniero del software podrá derivar casos de prueba que:

- garanticen que todas las rutas independientes dentro del módulo se han usado al menos una vez
- ejerciten los lados verdaderos y falsos de todas las decisiones lógicas
- ejecuten todos los bucles en sus límites y dentro de sus límites operacionales
- ejerciten estructuras de datos internos para asegurar si validez.

- Prueba de integración.

Es una técnica sistemática para construir la arquitectura del software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar componentes a los que se aplicó una prueba de unidad y construir una estructura de programa que determine el diseño.

- Módulo.

Esta prueba asegura que la información fluye apropiadamente hacia adentro y hacia afuera del sistema sujeto a la prueba. Se examinan las estructuras de datos locales para asegurar que todos los datos temporales mantienen la integridad durante todos los pasos de la ejecución de un algoritmo, se recorren todos los caminos independientes en toda la estructura para asegurar que al menos una vez fueron ejecutados.



- Regresión.

Si las pruebas tienen éxito, el resultado es el descubrimiento de errores y estos deben corregirse. Cada vez que el sistema se corrige también cambia algún aspecto de la configuración del sistema. La prueba de regresión es una actividad que ayuda a asegurar que los cambios no introduzcan comportamientos indeseables o errores adicionales.

- Estrés.

Las pruebas de estrés se proponen encontrar errores debidos a recursos bajos, verifica que el sistema funciona apropiadamente, usa scripts utilizados en pruebas de desempeño.

- Alfa.

Los usuarios finales son quienes aplican las pruebas alfa en el lugar de trabajo del desarrollador. El sistema se utiliza en un entorno natural mientras el desarrollador “mira sobre el hombro” de los usuarios típicos y registra los errores y los problemas de uso. Las pruebas alfa se realizan en un entorno controlado.

- Beta

Las pruebas beta se aplican en el lugar de trabajo de los usuarios finales. A diferencia de la prueba alfa, por lo general el desarrollador no está. Por tanto, la prueba beta es una aplicación “en vivo” del software en un entorno que no controla el desarrollador. El usuario final registra todos los problemas (reales o imaginarios) que encuentre durante la prueba y los informa de manera regular al desarrollador.



- Validación

Las pruebas de validación empiezan tras la culminación de la prueba de integración, la prueba se concentra en las acciones visibles para el usuario y en la salida del sistema que éste puede reconocer, la validación se alcanza cuando el sistema funciona de tal manera que satisface las especificaciones de requisitos del sistema.

- Unitarias

Ejecutadas principalmente sobre un componente, un módulo o un programa.

- Ejecutadas usualmente por los desarrolladores.
- Técnicas de caja blanca.
- Es importante definir el nivel de dependencia (a mayor dependencia mayor efectividad).
- Los desarrolladores utilizan el framework Junit a nivel de DAO y Services y Selenium a nivel Web para para realizar estas pruebas.



Pruebas realizadas al sistema.

- Caja negra

Error de inicialización, usuario o contraseña incorrectos. Véase Figura 4.4.1.

The image shows a login interface for CVU. At the top left is the CVU logo, which consists of a blue circle containing a stylized 'CVU' in red and blue. Below the logo is the text 'Ingresa a tu cuenta del CVU'. The main form area has a title 'Tipo de Acceso' with two radio buttons: 'Control Profesor' (unselected) and 'Alumno' (selected). Below this are two input fields: 'Usuario' containing the text '12345' and 'Contraseña' which is empty. A red error message is displayed below the fields: 'Introduzca la contraseña La cuenta de Usuario o la Contraseña que has introducido no coinciden'. At the bottom of the form is a dark blue button labeled 'Iniciar Sesión' and a blue link labeled 'Regístrame'.

Figura 4.4.1 Pantalla de error en usuario o contraseña



- Caja blanca

Módulos existentes al ingreso del sistema con el componente de base de datos.

Véase Figura 4.4.2.

```

Command Prompt - tail -1000f server.log

/WEB-INF/DispatcherUsuarioBuscadoCvu.xml
/WEB-INF/DispatcherRegistroUsuario.xml
20:22:05,521 Información [org.richfaces.log.Cache] (MSC service thread 1-13) Creating LRUMap cache instance of 512 itens capacity
20:22:05,526 Información [org.richfaces.log.Application] (MSC service thread 1-13) RichFaces Core Implementation by JBoss by Red Hat, version 4.3.7.Final
20:22:05,528 Advertencia [org.richfaces.log.Application] (MSC service thread 1-13) JMS API was found on the classpath; if you want to enable RichFaces Push JMS
20:22:05,529 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) RewriteFilter starting up...
20:22:05,538 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [4] org.ocpsoft.rewrite.servlet.spi.RewriteLifecycleListene
rg.ocpsoft.rewrite.servlet.config.lifecycle.JoinRewriteLifecycleListener(2147483647), org.ocpsoft.rewrite.servlet.impl.DefaultRewriteLifecycleListener(214748364
20:22:05,539 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.RequestCycleWrapper [o
20:22:05,539 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.RewriteProvider [org.ocpsoft.r
20:22:05,540 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.RewriteResultHandler [o
20:22:05,540 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.InboundRewriteProducer [o
20:22:05,541 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.OutboundRewriteProducer [o
20:22:05,542 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.ContextListener [org.oc
20:22:05,542 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [0] org.ocpsoft.rewrite.servlet.spi.RequestListener [ ]
20:22:05,543 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.servlet.spi.RequestParameterProvide
20:22:05,544 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.el.spi.ExpressionLanguageProvider [o
20:22:05,545 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.spi.InvocationResultHandler [org.oc
20:22:05,546 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [0] org.ocpsoft.common.spi.ServiceEnricher [ ]
20:22:05,547 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.spi.ConfigurationCacheProvider [org
20:22:05,549 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Loaded [3] org.ocpsoft.rewrite.config.ConfigurationProvider [org.o
org.ocpsoft.rewrite.annotation.config.AnnotationConfigProvider(100)]
20:22:05,551 Información [org.ocpsoft.rewrite.servlet.impl.DefaultHttpRewriteProvider] (MSC service thread 1-13) Loaded [0] org.ocpsoft.rewrite.spi.RuleCachePro
20:22:05,555 Información [org.ocpsoft.rewrite.param.DefaultParameterStore] (MSC service thread 1-13) Loaded [1] org.ocpsoft.rewrite.spi.GlobalParameterProvider [o
20:22:05,696 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-4) Rewrite 2.0.12.Final initialized.
20:22:05,697 INFO [org.jboss.web] (MSC service thread 1-4) JBAS018210: Registering web context: /seguimientoPosgradoFront
20:22:05,802 Información [org.ocpsoft.rewrite.servlet.RewriteFilter] (MSC service thread 1-13) Rewrite 2.0.12.Final initialized.
20:22:05,803 INFO [org.jboss.web] (MSC service thread 1-13) JBAS018210: Registering web context: /cvuFront
20:22:05,983 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS018565: Replaced deployment "seguridadFront.war" with deployment "seguridadFront.war"
20:22:05,984 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS018565: Replaced deployment "seguridadEAR.ear" with deployment "seguridadEAR.ear"
20:22:05,986 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS018565: Replaced deployment "seguimientoPosgradoFront.war" with deployment "seguimie
20:22:05,988 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS018565: Replaced deployment "seguimientoPosgradoEAR.ear" with deployment "seguimie
20:22:05,910 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS018565: Replaced deployment "cvuFront.war" with deployment "cvuFront.war"
20:22:32,340 INFO [stdout] (http-localhost-127.0.0.1-8080-4) null
20:22:32,388 Advertencia [javax.enterprise.resource.webcontainer.jsf.context] (http-localhost-127.0.0.1-8080-4) JSF1091: no se ha encontrado ningún tipo MIME pa
20:22:32,391 Advertencia [javax.enterprise.resource.webcontainer.jsf.context] (http-localhost-127.0.0.1-8080-4) JSF1091: no se ha encontrado ningún tipo MIME pa
20:22:40,936 INFO [stdout] (http-localhost-127.0.0.1-8080-4) null
20:22:40,965 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Inicia: 'DA0usuarios::consulta Detalle'
20:22:40,966 INFO [stdout] (http-localhost-127.0.0.1-8080-4) dev
20:23:02,035 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Error de E/S: The Network Adapter could not establish the connection
20:23:02,036 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Finaliza: 'DA0usuarios::Consulta Detalle'
20:23:02,046 Advertencia [javax.enterprise.resource.webcontainer.jsf.context] (http-localhost-127.0.0.1-8080-4) JSF1091: no se ha encontrado ningún tipo MIME pa
20:23:32,529 INFO [stdout] (http-localhost-127.0.0.1-8080-4) null
20:23:32,584 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Inicia: 'DA0usuarios::consulta Detalle'
20:23:32,584 INFO [stdout] (http-localhost-127.0.0.1-8080-4) dev
20:23:32,755 INFO [stdout] (http-localhost-127.0.0.1-8080-4) SELECT u.CVE_USUARIO, u.NOM_USUARIO, u.EMAIL, u.FECHA_VIGEN, u.CONTRASENA, u.MATRI_NOEMP, u
tu, CAT_ESTADOS_USUARIOS eu where u.CVE_TIPO_USUARIO= tu.CVE_TIPO_USUARIO AND u.CVE_EDO_USUARIO= eu.CVE_EDO_USUARIO AND u.CVE_EDO_REG= 1 AND u.MATRI_NOE
20:23:33,260 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Finaliza: 'DA0usuarios::Consulta Detalle'
20:23:33,275 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Inicia: 'DA0Permisos::consulta Lista'
20:23:33,275 INFO [stdout] (http-localhost-127.0.0.1-8080-4) dev
20:23:33,305 INFO [stdout] (http-localhost-127.0.0.1-8080-4) SELECT SEG_ROLES.CVE_ROL, SEG_ROLES.NOM_ROL, SEG_ROLES.CVE_EDO_REG, CASE NUL < SEG_USUARIOS_ROLES.(
lse 'TRUE' END AS rol_habilitado, SEG_CUENTAS_ROLES.CVE_TIPO_CUENTA from SEG_USUARIOS_ROLES SEG_USUARIOS_ROLES INNER JOIN SEG_ROLES SEG_ROLES ON( SEG_USUARI
EG_CUENTAS_ROLES ON( SEG_CUENTAS_ROLES.CVE_TIPO_CUENTA= 3 AND SEG_CUENTAS_ROLES.CVE_ROL= SEG_ROLES.CVE_ROL) ORDER BY SEG_ROLES.CVE_ROL
20:23:33,365 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Finaliza: 'DA0Permisos::Consulta Lista'
20:23:33,369 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Inicia: 'DA0usuarios::consulta Personal'
20:23:33,369 INFO [stdout] (http-localhost-127.0.0.1-8080-4) dev
20:23:33,399 INFO [stdout] (http-localhost-127.0.0.1-8080-4) Finaliza: 'DA0usuarios::Consulta Personal'
20:23:33,399 INFO [stdout] (http-localhost-127.0.0.1-8080-4) numEmpleado 13456

```

Figura 4.4.2 Componente de Base de Datos



- Prueba de integración

Prueba de uso de la interfaz para adjuntar archivos. Véase Figura 4.4.3.

Agregar Libro

Inicio

CVU

Usuario CVU

Nombre: JUAN CARLOS ROA

Libro:

ISBN: Identificador Libro:

Título del Libro:

Editorial: Edición: Año Publicación:

Volumen: Páginas: Tiraje:

Idioma: Área:

Disciplina:

Palabra Clave 1: Palabra Clave 2: Palabra Clave 3:

Liga:

Carátula: Adjunte un archivo: Archivos aceptados PDF, DOC, XLS y PPT con tamaño no mayor a 2 Mb.

Resumen del libro: Adjunte un archivo: Archivos aceptados PDF, DOC, XLS y PPT con tamaño no mayor a 2 Mb.

Carta de aceptación: Adjunte un archivo: Archivos aceptados PDF, DOC, XLS y PPT con tamaño no mayor a 2 Mb.

Autores:

Nombre: JUAN CARLOS Apellido Paterno: ZAVALETA Apellido Materno: BAUTISTA Orden de Aparición:

Nombre: Apellido Paterno: Apellido Materno: Orden de aparición:

Nombre	Apellido Paterno	Apellido Materno	Orden
--------	------------------	------------------	-------

Figura 4.4.3 Pruebas de archivos



- Módulo.

Pantallas de verificación ingreso de información durante la ejecución de un algoritmo. Véase Figuras 4.4.4 y 4.4.5.

Sistema de Productividad del Posgrado

Carlos Vazquez - Control

Actualizar Datos Generales

Datos Generales | Historia

Posgrado: Doctorado en Ciencias Biomédicas

Fecha de Creación: 10/03/2009 Grado Académico: Doctorado

Campus: Escuelas 2 ca 10102 Unidad académica de ciencias quimic ca

Coordinador: MICHELLE ca

Clave de Profesiones: ca

Nivel PNPC: Reciente creacion Alta PNPC: mar 10, 2009

Vigencia PNPC: mar 10, 2018 Área SNI: Área III Medicina y Ciencias de la Salud

Orientación: Profesional

PDF plan: Browse... Página Web: doctoradobiomedicas.unam.mx

Teléfono: 47 19310 ext. 3601 Correo Electrónico: iprojas@yahoo.com

Registro actualizado. Aceptar

Figura 4.4.4 Ingreso de información

Posgrados - Información General

Agregar Posgrado

Posgrado	Estado	Nivel PNPC	Fecha de Creación	Página Web	Acciones
Doctorado en Ciencias Biomédicas	Aprobado	Reciente creación	10/03/2009	http://doctoradobiomedicas.unam.mx	
Doctorado en Ciencias con Especialidad en Matemática Educativa	Reestructurado	Reciente creación	3/10/2005	http://doctoradome-unam.mx/	
Maestría en Ciencias Biomédicas	En creación	Consolidado	1/08/2006	http://maestriabiomedicas.com.mx	
Maestría en Ciencias Biomédicas-Clic	Aprobado	Consolidado	12/01/2016	http://www.unam.mx	
MAESTRÍA EN CIENCIAS: GESTIÓN SUSTENTABLE DEL TURISMO	En creación	Consolidado	4/02/2014	No Disponible	
Maestría en Docencia del Idioma Ingles	En creación	Consolidado	7/10/2009	No Disponible	
MAESTRÍA EN SALUD PÚBLICA	En creación	No pertenece a pnpic	20/10/2014	http://unam.edu.mx/	

Figura 4.4.5 Verificación de ingreso de información



Mantenimiento.

El mantenimiento del sistema explica casi el 60 por ciento del esfuerzo que emplea una organización de desarrollo, y el porcentaje continúa elevándose conforme se produce más software. El mantenimiento del software se define identificando cuatro actividades diferentes: mantenimiento correctivo, mantenimiento adaptativo, mejora o mantenimiento de perfeccionamiento y mantenimiento preventivo o reingeniería. Sólo cerca del 20 por ciento del trabajo de mantenimiento se emplea en componer errores. El restante 80 por ciento se dedica a adaptar los sistemas existentes a los cambios en su entorno externo, realizar las mejoras que solicitan los usuarios y rediseñar una aplicación para usarla en un futuro.

El mantenimiento de software de un producto consiste en las actividades realizadas sobre la aplicación una vez entregado el producto. El software es sometido a reparaciones y modificaciones cada vez que se detecta un fallo o se necesita cubrir una nueva necesidad de los usuarios. En esta fase recae el mayor porcentaje del coste de un sistema.

El glosario de IEEE [IEEE 610]⁵ describe el mantenimiento de software como: El proceso de modificar un sistema o componente de software entregado para corregir defectos, mejorar el desempeño o algún otro atributo, o adaptarlo al cambio del entorno. Lehman⁶ afirma como que si un programa no tiene una adaptación continua a las necesidades existentes, con el tiempo es cada vez menos útil.

⁵ IEEE (2010). Standards Glossary.

https://www.ieee.org/education_careers/education/standards/standards_glossary.html

⁶ Lehman, M. M. (1980). On Understanding Laws, Evolution, and Conservation in the Large-Program Life Cycle.



Las actividades del mantenimiento son similares a aquéllas del desarrollo. El mantenimiento enfoca simultáneamente cuatro aspectos mayores de la evolución del sistema:

- Mantener el control sobre las funciones diarias del sistema
- Mantener el control sobre las modificaciones del sistema
- Perfeccionar las funciones aceptables existentes
- Impedir que el desempeño del sistema se degrade a niveles inaceptables

Tipos de mantenimiento.

- Correctivo

A pesar de las pruebas y verificaciones que aparecen en etapas anteriores del ciclo de vida del software, los programas pueden tener defectos. El mantenimiento correctivo tiene por objetivo localizar y eliminar los posibles defectos de los programas.

- Preventivo

Consiste en la modificación del software para mejorar sus propiedades sin alterar sus especificaciones funcionales. Se pueden incluir sentencias que comprueben la validez de los datos de entrada, reestructurar los programas para mejorar su legibilidad, o incluir nuevos comentarios que faciliten la posterior comprensión del programa⁷.

⁷ Roger S. Pressman. (2006). Estrategias de prueba del software. En Ingeniería del Software. Un enfoque práctico.



- Cualitativo

Cuanto más atributos cualitativos se gestionen durante el desarrollo del software, mejor será su facilidad de mantenimiento. Desde un enfoque de calidad, estos atributos son: fiabilidad, modularidad, facilidad de comprensión, facilidad de prueba y facilidad de expansión.

- Perfectivo

Cambios en la especificación, normalmente debidos a cambios en los requisitos de un producto de software, implican un nuevo tipo de mantenimiento llamado perfectivo. Se puede definir el mantenimiento perfectivo como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

- Aumentativo

Se trata de la modificación del software para mejorar sus propiedades, aumentar su calidad sin alterar su funcionalidad, puede hacerse a través de la inclusión de sentencias que comprueben la validez de datos de entrada o incluir comentarios que faciliten la comprensión del sistema.

- Vertical

Se usa para hacer búsqueda de virus que puedan afectar la estabilidad del sistema, el bajo rendimiento, pérdida de productividad y sus probables errores, con el mantenimiento vertical se mejora y optimiza el software que se ha instalado.

- Horizontal

Se enfoca en el hardware, la limpieza interna de la computadora, y está compuesto por:



- mantenimiento general
- mantenimiento correctivo
- mantenimiento predictivo
- preventivo

A nivel perfectivo vamos a aplicar dos tipos de mantenimiento. El primero será a nivel de JBoss, se va a optimizar el uso de la memoria de la máquina virtual de java, se configurará de acuerdo a los parámetros óptimos que necesite la aplicación, conforme vaya creciendo el consumo o el uso del sistema.

El segundo es a nivel de la base de datos, se ajustarán los parámetros, se realizarán las mejoras en las sentencias SQL que tengan problemas de rendimiento, se ejecutarán tareas administrativas como estadísticas de la base de datos para que el optimizador de la base de datos se ejecute de manera correcta, y realizar la desfragmentación de tablas en el caso de mucha fragmentación⁸.

⁸ Viveros P., Stegmaier R., Kristjanpoller F., Barbera L., Crespo A. (2013). Propuesta de un modelo de gestión de mantenimiento y sus principales herramientas de apoyo. *Ingeniare. Revista chilena de ingeniería*. http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052013000100011&lng=es&tlng=es. 10.4067/S0718-33052013000100011



4.5 Evaluación del sistema web.

En función a los requerimientos generales y particulares del usuario, planteados en el capítulo 3, se realizó la evaluación del sistema web. A continuación se muestran algunos puntos representativos de tales condiciones.

Para el módulo del Sistema de Productividad del Posgrado de la Universidad (SPPU), se plantearon, entre otros, los siguientes requerimientos en el capítulo 3:

- Cada módulo podrá trabajar de manera independiente pero la información debe ser compartida con el resto de módulos, en caso de ser necesario.
- Cada usuario tendrá un perfil que le brindará permisos para la consulta o modificación de información.

En la figura 4.5.1 se muestra la pantalla de ingreso del módulo Sistema de Productividad del Posgrado. Para satisfacer la segunda condición referida antes de la imagen, en este módulo el usuario debe indicar el perfil con el que cuenta (Tipo de Acceso), y dependiendo de este perfil tendrá diferentes permisos para consultar o modificar la información. Además, se observa que este módulo es independiente, pero comparte información con los otros módulos, ya que de acuerdo al perfil del usuario se activarán los módulos correspondientes; de esta forma se atiende el primero de los requerimientos del usuario, señalados anteriormente.



Sistema de Productividad del Posgrado

Ingresa a tu cuenta del SPP

Tipo de Acceso

Control Profesor Alumno

Usuario
13456

Contraseña
.....

Iniciar Sesión

[Regístrate](#)

Copyright © 2016 Universidad,
Av. uni, C.P. 1111, Estado, Mexico. Todos los derechos reservados.

Fig. 4.5.1 Pantalla de ingreso del módulo SPPU

El usuario planteó las siguientes condiciones, entre otras, para el módulo de Estructura del Posgrado:

- Se debe permitir mostrar todos los posgrados existentes registrados por la dirección de posgrados.
- Se imprimirá un reporte completo de la estructura del posgrado en formato PDF.



El cumplimiento a estos dos requerimientos se muestra en la figura 4.5.2, en la cual se puede observar el listado de los posgrados existentes, ordenado en dos páginas. También en esta pantalla se tiene acceso a la impresión del reporte de la estructura del posgrado de la universidad, desde el botón “Reportes” del bloque Menú.

Dos de los requerimientos planteados por el usuario para el módulo Investigación, fueron los siguientes:

- El sistema permitirá el registro y control de diversos proyectos de investigación de la universidad, categorizados en distintos rubros:
 - Libros. (...)
- El sistema permitirá la consulta en forma de listado de cada categoría.

Como se muestra en la figura 4.5.3, el módulo permite a un usuario con perfil Profesor, registrar libros. En la figura 4.5.4 se observa el listado de libros registrados por el mismo usuario, el cual puede realizar distintas acciones, como editar o borrar el registro. Con esto se cubren estas necesidades del usuario.

Sistema de Productividad del Posgrado

Carlos Vazquez - Control

Posgrados - Información General

Agregar Posgrado

Posgrado	Estado	Nivel PNPC	Fecha de Creación	Página Web	Acciones
Doctorado en Ciencias Biomédicas	Aprobado	Reciente creación	10/03/2009	http://doctoradobiomedicas.unam.mx	
Doctorado en Ciencias con Especialidad en Matemática Educativa	Reestructurado	Reciente creación	31/02/2005	http://doctoradome.unam.mx/	
Maestría en Ciencias Biomédicas	En creación	Consolidado	1/08/2006	http://maestriabiomedicas.com.mx	
Maestría en Ciencias Biomédicas-Clic	Aprobado	Consolidado	12/01/2016	http://www.unam.mx	
MAESTRÍA EN CIENCIAS. GESTIÓN SUSTENTABLE DEL TURISN	En creación	Consolidado	4/02/2014	No Disponible	
Maestría en Docencia del Idioma Inglés	En creación	Consolidado	7/10/2009	No Disponible	
MAESTRÍA EN SALUD PÚBLICA	En creación	No pertenece a pnpc	20/10/2014	http://unam.edu.mx/	

Copyright © 2016 Universidad .
Av. uni, C.P. 1111, Estado, México. Todos los derechos reservados.

Inicio

Posgrados

Estructura y Personal Académico

Estudiantes


Infraestructura

Resultados y Vinculación

Reportes

Fig. 4.5.2 Pantalla de Información General de los Posgrados





JUAN CARLOS ROA - Profesor

Agregar Libro

Inicio

CVU

Usuario CVU

Nombre: JUAN CARLOS ROA

Libro:

Identificador Libro:

✖ El ISBN no cumple con un formato válido

Título del Libro:

Editorial:	<input type="text"/>	Edición:	<input type="text"/>	Año Publicación:	<input type="text"/>
Volumen:	<input type="text"/>	Páginas:	<input type="text"/>	Tiraje:	<input type="text"/>
Idioma:	<input type="text" value="Selecciona"/>	Área:	<input type="text" value="Selecciona"/>		
Disciplina:	<input type="text" value="Selecciona"/>				

Palabra Clave 1: Palabra Clave 2: Palabra Clave 3:

Liga:

Carátula:	<input type="button" value="Browse..."/>
Resumen del libro:	<input type="button" value="Browse..."/>
Carta de aceptación:	<input type="button" value="Browse..."/>

Fig. 4.5.3 Formulario de registro para agregar un libro



Fig. 4.5.4 Listado de los libros registrados por un usuario

Para el módulo de Resultados y Vinculación, uno de los requerimientos que planteó el usuario es el siguiente:

- Se mostrará una pantalla para consultar la información del total de contratos que están dados de alta en el posgrado. Esta información podrá ser extraída en forma de reporte.

Como se observa en la figura 4.5.5, esta condición se cumple, ya que el módulo permite la visualización de los contratos dados de alta para cada uno de los posgrados, y mediante el botón “Reportes” de la barra Menú es posible generar el reporte correspondiente.

Listado de Contratos

Posgrado: Doctorado en Ciencias con Especialidad en Matemática Educativa

Nombre Contrato: Inicio: Fin:

Menor: Mayor: Buscar Contratos

Agregar Contrato

Nombre Contrato	Empresa	F. Inicio	Monto	Operaciones
CONTRATO DE MANTENIMIENTO	UNAM	17/12/2015	60000.0	
CONTRATO DE SOPORTE	UNAM	29/01/2016	567.89	
CONTRATO DE RENOVACIÓN	UNAM	10/12/2015	15000.0	
Contrato infra	UNAM	26/11/2015	10.0	

[Regresar](#)

Fig. 4.5.5 Información de Contratos en Posgrado



Alta Protocolo

Nombre: MAGDA PAULLINA GARCIA ANORVE **Matrícula:** 8202325

Posgrado: maestría cic

Guardar

Regresar

Datos

Fecha: 31/05/2016

Título:

Archivo:

Estatus: En Creación

Comentario Director:

Fig. 4.5.6 Formulario para adjuntar archivos PDF relativos a tesis



El usuario planteó la siguiente condición, entre otras, para el módulo de Permanencia de Estudiantes:

- El sistema aceptará archivos PDF que contengan la información relativa a las tesis y sus avances de cada alumno. Cuando un alumno adjunte avances de su tesis, el sistema notificará vía correo al asesor correspondiente.

En la figura 4.5.6 se muestra la pantalla donde los alumnos pueden adjuntar archivos PDF referentes a sus trabajos de tesis; aunque no se muestra en esta imagen, cuando el archivo es registrado por el sistema, el asesor del alumno recibe una notificación mediante correo electrónico. Con esto se cubre el requerimiento establecido anteriormente.

Uno de los requerimientos planteados por el usuario para el módulo Ingreso, es el siguiente:

- Se mostrará una pantalla para capturar los datos de la convocatoria. Se permitirá adjuntar archivos que respalden la información ingresada.

La necesidad del usuario referente a que el sistema cuente en el módulo Ingreso con una pantalla para capturar los datos de convocatorias de ingreso a la Universidad, se cubre con lo mostrado en la figura 4.5.7. De igual forma se observa que cuenta con una casilla para adjuntar archivos que respalden la información ingresada.

Inicio

Posgrados

Estructura y Personal Académico

Estudiantes

Infraestructura

Resultados y Vinculación

Reportes

Listado de Convenios

Posgrado: Doctorado en Ciencias con Especialidad en Matemática Educativa

Agregar Convenio

Lista de Convenios					
Convenio	Objetivo	Fecha de Termino	Fecha de Inicio	Acciones	
CONVENIO UNAM - IPN	Proyecto de intercambio de estudiantes	19/11/2015	26/11/2015	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UNAM - CONACYT	Proyecto de desarrollo de investigación	4/11/2015	12/11/2015	<input type="checkbox"/>	<input checked="" type="checkbox"/>
UNAM - IPN	Movilidad de estudiantes	10/12/2015	11/12/2015	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Regresar

Fig. 4.5.7 Pantalla de captura para los datos de convocatoria



Para el módulo Infraestructura, se plantearon, entre otros, los siguientes requerimientos:

- Se mostrará una pantalla para consultar la información del inventario de cubículos dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de aulas dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de laboratorios dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario del equipo de cómputo para profesores y alumnos dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de bibliotecas dados de alta en el posgrado.
- Se mostrará una pantalla para consultar la información del inventario de biblioteca virtual dados de alta en el posgrado.

En la figura 4.5.8 se muestra el inventario de aulas para un programa en particular del Posgrado; mediante pestañas, en esa misma pantalla se pueden consultar los inventarios de cubículos, laboratorios, equipo de cómputo, bibliotecas, y bibliotecas virtuales para el mismo programa de Posgrado. De esta forma, se da cumplimiento a las necesidades del usuario en este sentido.



Inventario de la Infraestructuras

Posgrado: Doctorado en Ciencias con Especialidad en Matemática Educativa
Unidad A. Sede: Unidad académica de matemáticas

Aulas
Cubículos
Biblioteca Virtual
Biblioteca
Equipo de cómputo
Laboratorio

No. de Aulas	Fecha de Inventario	Operaciones
1	2/02/2016	
4	14/01/2016	
10	14/01/2016	
2	8/01/2016	

[Regresar](#)

Fig. 4.5.8 Formulario de captura de inventario de aulas

Con lo anterior, se demuestra la satisfacción de los requerimientos del usuario para el Sistema de Productividad del Posgrado planteados por la Universidad.



CONCLUSIONES

Desde el planteamiento de la propuesta, el desarrollo del presente trabajo de tesis ha representado un reto para nuestro grupo. Al final, nuestra labor nos ha permitido llegar a las siguientes conclusiones:

- Se cumplió con el objetivo planteado originalmente para el desarrollo de esta tesis, el cual nos llevó a analizar, diseñar e implementar un sistema de productividad de un posgrado de una universidad.
- La información y documentos se encuentran almacenados de manera centralizada y segura, permitiendo la consulta fácil y rápida, reduciendo tiempos de generación de documentos y reportes.
- Se fundamentó la solución del sistema utilizando un conjunto de tecnologías, base de datos y servidores de aplicaciones que nos permiten manejar de manera óptima la gran cantidad de usuarios e información requerida por la universidad.
- La culminación de este proyecto nos permitió aplicar los conocimientos teóricos que nos brindó la universidad, así como mostrar la experiencia acumulada en nuestra vida laboral.
- La información está en línea, a disposición de cualquier usuario registrado, sin importar su ubicación geográfica ni Unidad Académica. De esta forma se reducen los tiempos de traslado para trámites y documentación necesarios en los procesos administrativos del Posgrado de la Universidad.
- Se mejoraron los tiempos de respuesta a los requerimientos solicitados por el CONACyT de manera más eficaz y eficiente, asegurando la certificación de los posgrados por el PNPC (Programa Nacional de Posgrado de Calidad).



CONCLUSIONES



- Se mejoró el seguimiento al proceso de titulación y a tutorías, aumentando la eficiencia terminal; además el seguimiento a los egresados permite elevar el cumplimiento de los objetivos de todos los programas educativos del Posgrado.
- El sistema permite controlar, administrar e incrementar la cantidad del número de investigadores inscritos en el Sistema Nacional de Investigadores (SNI). Lo que a su vez contribuye a garantizar la evaluación ante el PNPC.
- El sistema desarrollado permite incrementar la productividad académica, por medio de un manejo más eficiente de los apoyos que la Universidad obtiene a través del CONACyT. Todo debido a que el sistema facilita generar indicadores que miden la productividad de libros y artículos publicados, desarrollos e instancias de investigación, distinciones y docencias.
- Con el desarrollo del sistema, concluimos que la etapa de análisis del problema es la más crítica, ya que en ésta es importante lograr una correcta delimitación de los alcances, herramientas a emplear, técnicas y diseños que se utilizarán para llevar a buen término el proyecto.
- El desarrollo del sistema favoreció a mejorar el seguimiento de la disponibilidad y calidad de la infraestructura de los posgrados de la Universidad.
- Se consiguió mejorar la ejecución y control de los procesos administrativos y académicos de los posgrados.

Con la aportación de cada uno de los miembros con su experiencia laboral, la planeación de proyectos, la gestión del tiempo, el apoyo mutuo, y el planteamiento del problema, conseguimos la integración del equipo, y de esta forma llegar a buen término este trabajo de tesis.



REFERENCIAS BIBLIOGRÁFICAS

- Bass L., Clements P., Kazman R. (2003). Software Architecture in Practice, 2nd Edition, Addison Wesley.
- Coutin, A. (2002) Arquitectura de información para sitios web. Anaya. ISBN: 84-415-1387-2.
- Date, C. (2001). Introducción a los sistemas de bases de datos. Pearson Educación. ISBN: 968-444-419-2.
- Ferrer, J. (2014). Implantación de aplicaciones web en entornos de internet, intranet y extranet. Paracuellos de Jamara, Madrid.
- Figueroa, R. (2002). Manual de bases de datos. DGSCA Guías y textos de cómputo.
- Gómez, M. (2013). Notas del curso bases de datos. Universidad Autónoma Metropolitana. ISBN: 978-607-477-880-9.
- Gonzalo, L. (1996). Ingeniería de Sistema de Software, ISDEFE, 1ra edición.
- Grupo Isaias Carrillos Perez. (2008). Metodología del desarrollo de software. New York, Editorial Edit and write.
- Hansen, G. (1997). Diseño y administración de Bases de Datos, Prentice Hall Iberia, 2da edición.
- ITSA (2008). Metodologías De Desarrollo De Software, Canada, Editorial Canada Pen.
- Korth, H, et al., Fundamentos de Bases de Datos, Mc Graw Hill, 3ra edición.
- Lehman, M. (1980). On Understanding Laws, Evolution, and Conservation in the Large-Program Life Cycle. Journal of Systems and Software 1: 213-221.
- Leonard, A. (2010). JSF 2 CookBook. Packt Publishing. ISBN-13: 978-1847199522



- Lynch, P. (2009). Web Style Guide: Basic Design Principles for Creating Web Sites. Yale University Press. 3rd. Edition. ISBN: 978-0300137378.
- Pérez, M. (2013). Programación orientada a objetos con ORACLE PL/SQL. Createspace.
- Pressman, R. (2010). Ingeniería del Software, Un Enfoque Práctico. McGraw-Hill, 7a edición.
- Roger, S. Pressman. (2006). Estrategias de prueba del software. En Ingeniería del Software. Un enfoque práctico(382-461). México: McGraw-Hill.
- Silberschatz, A. (2002). Fundamentos de Bases de Datos. McGraw-Hill.
- Sommerville, I. (2005). Software Engineering. Pearson Addison Wesley. 6ta edición.
- Teaching soft group. (2012). Programación en Oracle 11g. SQL, SQL*Plus y PL/SQL. Alfaomega grupo editor, S.A de C.V. ISBN: 978-84-9964-118-8.
- Tong, K. (2009). Beginning JSF 2 APIs and JBoss Seam. aPress.
- Yourdon, E. (1994). Análisis Estructurado Moderno. Prentice Hall.
- Weitzenfeld, A. (2005). Ingeniería de Software Orientada a Objetos con UML, Java e Internet. Thomson Editores, S.A. de C.V. ISBN: 970-686-190-4.
- Aguilar G., Concha E., Corona N., Macario Alejandro & Reyes J. (2010). Sistema de punto de venta para una empresa fotográfica (Tesis de licenciatura). UNAM, México. Recuperado de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/1022/Tesis.pdf?sequence=1>
- Linares V. (2004). Análisis de sistemas de navegación de sitios Web (Tesis de licenciatura). UOC, Catalunya. Recuperado de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/447/1/27698tfc.pdf>



REFERENCIAS WEB

- Centeno, E. (2008). Formas normales: 1FN, 2FN, 3FN, 4FN, 5FN. Consultado el 10 de junio de 2016
http://istprpercenteno.blogspot.mx/2008_06_01_archive.html
- Chaudhary N., Dickenson R., Gilda S. (2013). JBoss Enterprise Application Platform 6 Administration and Configuration Guide. 12 de Mayo 2016, de redhat. Sitio Web: https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6/pdf/Administration_and_Configuration_Guide/JBoss_Enterprise_Application_Platform-6-Administration_and_Configuration_Guide-en-US.pdf
- Gilda, S. Logue, E. Mison, D. Ryan, D. JBoss Enterprise Application Platform 6 Instalaltion Guide. 13 de Junio de 2016, de redhat. Sitio Web: https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6/pdf/Installation_Guide/JBoss_Enterprise_Application_Platform-6-Installation_Guide-en-US.pdf
- CONACYT. (2015). Marco de referencia para la evaluación y seguimiento de programas de posgrado presenciales. 7 de Mayo 2016, de CONACYT. Sitio web: <http://www.conacyt.mx/index.php/el-conacyt/convocatorias-y-resultados-conacyt/convocatorias-pnpc/convocatorias-cerradas-pnpc/9005-marco-de-referencia-modalidad-escolarizada/file>
- CONACYT. Sistema de consultas. 2 de Junio 2016, de CONACYT. Sitio web: <http://svrtmp.main.conacyt.mx/ConsultasPNPC/intro.php>
- Eisele, M. (2013). Selecting your Java EE 6 Application Server. 19 de Septiembre de 2016. Sitio Web: <http://blog.eisele.net/2013/01/selecting-your-java-ee-6-application.html>
- Johnson R., Hoeller J., Donald K. (2004-2012). Spring Framework Reference Documentation. 7 de Junio de 2016. Sitio web: <http://docs.spring.io/spring-framework/docs/3.2.0.RC2/reference/htmlsingle/>



- IEEE. Standards Glossary. 11 de Mayo 2016, de IEEE Advancing Technology for Humanity. Sitio Web: https://www.ieee.org/education_careers/education/standards/standards_glossary.html
- Java. (2013). The Java EE 6 Tutorial. 5 de Mayo 2016, de Java. Sitio Web: <http://docs.oracle.com/javaee/6/tutorial/doc/bnaaw.html>
- Java. (2008). JSRs: Java Specification Requests. 5 de Mayo 2016, de Java Community Process. Sitio Web: <https://jcp.org/en/jsr/detail?id=316>
- Kelapure, R. (2011). Java EE and Spring Framework Shootout. 19 de Septiembre de 2016. Sitio Web: <http://www.slideshare.net/kelapure/java-ee-spring-shootout>
- Lorza, I. (2013). Cuadro comparativo de los modelos de base de datos. 19 de Septiembre de 2016. Sitio Web: <http://es.slideshare.net/lorza1/ventajas-desventajas-de-los-modelos-de-bd>
- ORACLE. (2007). Arquitectura de Oracle Database 11g. 3 de Junio 2016. Sitio web: <http://www.oracle.com/technetwork/es/documentation/317491-esa.pdf>
- ORACLE. (2009). Familia de productos Oracle Database 11g. 6 de Junio de 2016. Sitio web: <http://www.oracle.com/technetwork/es/database/enterprise-edition/documentation/productos-oracle-database-11g-1704970-esa.pdf>
- Rebellabs. (2016). Java Tools and Technologies Landscape Report 2016. 19 de Septiembre de 2016. Sitio Web: <http://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016/>
- Serbat, A. Rojo, A. La Guía de HTML 5, CSS3 y JAVASCRIPT. 16 de Mayo 2016, de ITESM. Sitio Web: <http://cs.mty.itesm.mx/profesores/raul.perez/DAW/IntroHtml5.pdf>



REFERENCIAS BIBLIOGRÁFICAS

- Viveros, Pablo, Stegmaier, Raúl, Kristjanpoller, Fredy, Barbera, Luis, & Crespo, Adolfo. (2013). Propuesta de un modelo de gestión de mantenimiento y sus principales herramientas de apoyo. *Ingeniare. Revista chilena de ingeniería*, 21(1), 125-138. Recuperado en 29 de mayo de 2016, de http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052013000100011&lng=es&tlng=es. 10.4067/S0718-33052013000100011
- Wikipedia. (2016). Anexo: Comparación de sistemas administradores de bases de datos relacionales. 19 de Septiembre de 2016. Sitio Web: https://es.wikipedia.org/wiki/Anexo:Comparación_de_sistemas_administradores_de_bases_de_datos_relacionales