MINISTERIO DE DEFENSA NACIONAL
COMANDO GENERAL DE LAS FUERZAS MILITARES
EJÉRCITO NACIONAL
DIRECCION DE PERSONAL

INFORME DE SUPERVISIÓN Y RECIBO A SATISFACCIÓN PARCIAL X TOTAL ___
No. 10

Bogotá (Cundinamarca) NOVIEMBRE 2025

1. CONTRATISTA	Nombre completo: ERICK ENRIQUE SÁNCHEZ DIAZ Identificación 1.023.886.053 Nacionalidad Colombiana Dirección Carrera 7 #31-21 Sur Bogotá Teléfono de contacto 3163551894 E-mail de contacto: erick.eagle@gmail.com
2. SUPERVISOR	Nombre completo CT. ALVARADO MORATO JONATAN ANDRÉS Cargo Oficial Administrador Informático Resolución de Delegación (Resolución No.00000352 fecha 07 DE ENERO DE 2025 Teléfono de contacto 3012618297 E-mail de contacto: jonatan.alvarado@ejercito.mil.co
3. No. DEL CONTRATO – FECHA DE SUSCRIPCIÓN	No. del contrato 125-CENACPERSONAL-2025 Fecha de suscripción 15/01/2025
4. PÓLIZA DE GARANTÍA ÚNICA	No. De la Póliza: **360-47-994000039956** Entidad que expide la póliza: Aseguradora Solidaria de Colombia Fecha del certificado de aprobación de la Póliza: 30/ENE/2025* Aprobado por: TC. BEJARANO GARCÍA JAIRO ANDRÉS
5. CRP	No.**5025** de fecha **30 de enero del 2025** Fecha expedición: **30 de enero del 2025** Unidas/Sub-unidad ejecutora CENAC PERSONAL Dependencia: DIPER Posición catálogo de gasto: **(A-02-02-02-008-003-13)** Fuente: NACIÓN Recurso **10** Valor: **$56.100.000.oo**
6. FECHA INICIO EJECUCIÓN CONTRACTUAL	15/ENE/2025
7. PLAZO DE EJECUCIÓN	15/AGO/2025
8. OBJETO CONTRACTUAL	PRESTAR LOS SERVICIOS PROFESIONALES COMO INGENIERO DE SISTEMAS CON EXPERIENCIA EN DESARROLLO DE SOFTWARE BAJO LA PLATAFORMA DE JAVA Y DESARROLLO EN PL/SQL
9. OBLIGACIONES CONTRACTUALES	Obligaciones Contractuales Generales: 1. Observar durante el tiempo de ejecución del contrato, buena disposición y actitud favorable hacia el trabajo, respeto hacia sus compañeros y miembros del Ejército Nacional, al igual que la aplicación permanente de principios y valores propios de la institución militar. 2. Dar estricto cumplimiento durante todas sus actuaciones, al marco normativo correspondiente a la integridad, trasparencia y lucha contra la corrupción. 3. Guardar y mantener la debida reserva y confiabilidad frente a los temas y asuntos tratados y conocidos dentro del desarrollo y ejecución del presente contrato, reserva que se hará extensiva a los derechos de autor que de los mismos se derive. 4. Aplicar medidas pasivas de seguridad con la documentación según clasificación con el fin de evitar la fuga de información de acuerdo con la normatividad vigente.

DIPER

<table>
<tr><td></td><td>
5. De conformidad con el artículo 3 del Decreto 2609 de 2012, el contratista será responsable por la gestión documental de las actividades realizadas durante la ejecución del Contrato, presentando en el cronograma de actividades y en su informe de forma mensual el barrido de los documentos en físico y el término del almacenamiento del Orfeo hasta el archivo de los documentos.
6. El contratista debe custodiar y a la terminación del presente contrato devolver los insumos, suministros, herramientas, dotación, implementación, inventarios y/o materiales que sean puestos a su disposición para la prestación del servicio objeto de este contrato.
7. Entregar al final de la prestación del servicio, backup de la información producto ejecución del contrato.
8. El contratista autoriza al MDN – EJÉRCITO NACIONAL a realizar los estudios de seguridad, comprobación de lealtad, realización de los exámenes de poligrafía y pruebas sicométricas antes y durante la contratación.
9. Las demás obligaciones establecidas en el estudio previo, documento que el contratista declara conocer, y que hace parte del presente contrato.
10. Las demás obligaciones que se deriven del objeto del contrato.

Obligaciones Especificaciones

1. Realizar levantamiento, análisis y socialización de los requerimientos y mejoras de los sistemas de información bajo la tecnología java.

2. Realizar verificación y ajuste en los datos, los formularios, los reportes de los sistemas de información bajo la tecnología java.

3. Realizar propuestas de solución de acuerdo a los requerimientos para ajustes de presentación de datos en los diferentes módulos reportes de los sistemas de información bajo la tecnología java.

4. Diseñar o desarrollar formularios y reportes para el manejo de la información de acuerdo a la verificación de los requerimientos.

5. Realizar todos los demás ajustes adicionales a las formularios, reportes, procesos y procedimientos requeridos por el supervisor del contrato cuando se presente la necesidad inmediata relacionada con el funcionamiento de los sistemas de información bajo la tecnología java.
</td></tr>
<tr><td>10. PERIODO DEL INFORME</td><td>NOVIEMBRE 2025</td></tr>
<tr><td>11. DOCUMENTOS QUE SOPORTAN EL CUMPLIMIENTO DE LAS OBLIGACIONES CONTRACTUALES</td><td>
Obligación No. 7: Apoyar en la revisión de los documentos soportes de las hojas de vida en el SIGEP para la contratación de los prestadores de servicios.
<table>
<tr><th>No</th><th>TIPO DE DOCUMENTO</th><th>No. DE RADICADO</th><th>ASUNTO</th></tr>
<tr><td></td><td>PLATAFORMA SIGEP II</td><td>N/A</td><td>REVISION Y VERIFICACION DE LA DOCUMENTACION QUE DEBE ESTAR CARGADA EN SECOP RESPECIVAMENTE:
1.DANTE
2.CAAID.
3.CCOES.
4.CENAC PER
5. DIPSO
6. DIPER
7.DIGEH
8.DADAE</td></tr>
</table>
</td></tr>
</table>

DIPER

Obligación No. 9: (9. Todas las demás que le sean asignadas de conformidad con el objeto contractual.)

No	TIPO DE DOCUMENTO	No. DE RADICADO	ASUNTO
	REVISIÓN DE LAS CARPETAS DE LAS SIGUIENTES UNIDADES	N/A	REVISION Y VERIFICACION DE LA DOCUMENTACION QUE DEBE ESTAR CARGADA EN SECOP RESPECIVAMENTE: 1.DANTE 2.CAAID. 3.CCOES. 4.CENAC PER 5. DIPSO 6. DIPER 7.DIGEH 8.DADAE

Informe de Noviembre/25:

2. Realizar verificación y ajuste en los datos, los formularios, los reportes de los sistemas de información bajo la tecnología java.

1. PRUEBAS CON BASE DE DATOS SISPE

Como parte del proceso de implementación del sistema SISPE, se creó una copia de la base de datos original del sistema ASTRA, con el objetivo de conservar la misma estructura de objetos, metadatos y configuraciones esenciales del entorno productivo.

En principio, la copia debía garantizar la interoperabilidad y compatibilidad con el nuevo sistema, dado que incluía la última versión estable de los datos.

Durante las pruebas iniciales, sin embargo, se identificaron errores en tiempo de ejecución relacionados con consultas SQL fallidas, referencias a tablas inexistentes y problemas de autenticación LDAP.

Tras un análisis detallado se concluyó que no todos los esquemas de la base de datos original fueron incluidos en la copia, lo cual generó inconsistencias en la resolución de dependencias a nivel de objetos.

2. FUNCIONAMIENTO DE LOS ESQUEMAS EN BASES DE DATOS ORACLE

En Oracle, un esquema representa un espacio lógico que agrupa objetos de base de datos (tablas, vistas, índices, procedimientos, triggers, sinónimos, etc.) pertenecientes a un mismo usuario propietario.

Cada esquema está vinculado de forma directa a un usuario, y actúa como su contenedor de objetos. Cuando una aplicación o consulta requiere acceder a un objeto de otro esquema, debe hacerlo mediante una referencia calificada (por ejemplo, NUEVO_REHU.TABLA_EMPLEADOS) o mediante un sinónimo público o privado que oculte dicha referencia.

Durante el proceso de copia de la base de datos, solo se replicó el esquema principal asociado a ASTRA, omitiéndose los esquemas NUEVO_REHU y COPER_ADMINSEG, los cuales forman parte de la arquitectura distribuida

DIPER

del sistema.

Estos esquemas contienen tablas dependientes y procedimientos almacenados críticos, por lo que su ausencia ocasionó fallos en las consultas y restricciones de integridad referencial.

3. USO DE DISCRIMINADORES PARA COMPARTIR TABLAS ENTRE SISTEMAS Y ENTORNOS

El esquema COPER_ADMINSEG contiene tablas de seguridad y configuración compartidas entre múltiples sistemas, entre ellos ASTRA y SISPE.

Para evitar duplicar información y permitir la reutilización de estas tablas, se utiliza una estrategia de discriminadores lógicos, generalmente implementada mediante campos de tipo código o identificador de sistema (por ejemplo, COD_SISTEMA o ORIGEN_APP).

Este discriminador permite distinguir qué registros pertenecen a cada aplicación sin necesidad de crear estructuras duplicadas.

De esta manera, ambos sistemas pueden compartir la misma tabla física sin interferir entre sí, garantizando la integridad de los datos mediante reglas de filtrado en las consultas (WHERE COD_SISTEMA = 'SISPE').

Esta práctica resulta ventajosa cuando los sistemas comparten componentes transversales, como módulos de autenticación, roles o configuración general, y se requiere una administración centralizada de usuarios y permisos.

1. Realizar levantamiento, análisis y socialización de los requerimientos y mejoras de los sistemas de información bajo la tecnología java.

4. ACCESO REMOTO MEDIANTE DATABASE LINKS (DB_LINK)

Para resolver la ausencia del esquema NUEVO_REHU en la nueva base de datos, se implementó un Database Link (DB_LINK), mecanismo nativo de Oracle que permite establecer conexiones remotas entre bases de datos.

Un DB_LINK actúa como un puntero lógico que habilita la ejecución de consultas sobre objetos ubicados en otra instancia Oracle, utilizando la sintaxis tabla@nombre_dblink.

Ventajas encontradas:

1. Permite el acceso transparente a datos remotos sin necesidad de replicarlos físicamente.
2. Facilita la integración entre bases distribuidas o ambientes separados (producción, pruebas, desarrollo).

DIPER

3. Reduce tiempos de migración y evita duplicidad de estructuras en escenarios temporales o de transición.

Desventajas:

1. Introduce dependencia de red: cualquier latencia o desconexión afecta la ejecución de consultas.
2. Puede impactar el rendimiento en operaciones de alto volumen o transacciones complejas.
3. Implica riesgos de seguridad si no se restringe adecuadamente el acceso y la autenticación del enlace.
4. Dificulta la portabilidad del sistema, ya que las consultas dependen de la existencia del DB_LINK.

4. Diseñar o desarrollar formularios y reportes para el manejo de la información de acuerdo a la verificación de los requerimientos.

5. ESTRATEGIA CON VISTAS (VIEWS) PARA COMPATIBILIDAD CON CÓDIGO EXISTENTE

Para evitar la modificación de las consultas originales del sistema, las cuales accedían directamente a las tablas mediante su nombre (por ejemplo, SELECT * FROM NUEVO_REHU.TABLA_X), se adoptó una estrategia basada en vistas locales.

Consistió en crear vistas con el mismo nombre de las tablas remotas, pero que internamente apuntan a las tablas accedidas mediante el DB_LINK.
Por ejemplo:

```
CREATE OR REPLACE VIEW TABLA_X AS
SELECT * FROM TABLA_X@DBLINK_NUEVO_REHU;
```

Esta técnica presenta ventajas significativas:

1. Mantiene la compatibilidad con el código original, evitando modificar cientos de consultas existentes.
2. Centraliza la configuración de acceso remoto, ya que el DB_LINK se gestiona en un único punto.
3. Permite reemplazar el acceso remoto por acceso local en el futuro (por ejemplo, si se replica el esquema), sin alterar la lógica de las aplicaciones.
4. Como desventaja, se debe considerar que las vistas pueden introducir un nivel adicional de abstracción, lo que puede afectar el rendimiento si se utilizan en grandes volúmenes de datos o en consultas muy anidadas.

DIPER

Luego de implementar las soluciones descritas:

1. Se restableció el acceso completo a los datos requeridos por el sistema SISPE.
2. Se garantizó la interoperabilidad con los esquemas faltantes mediante conexión directa o DB_LINK.
3. Se preservó la compatibilidad con el código legado utilizando vistas equivalentes a las tablas originales.
4. Se optimizó el mantenimiento de los datos compartidos mediante el uso de discriminadores por sistema.
5. Estas acciones permitieron estabilizar el entorno y habilitar las pruebas funcionales del nuevo sistema sin afectar la integridad ni la estructura original de los datos.

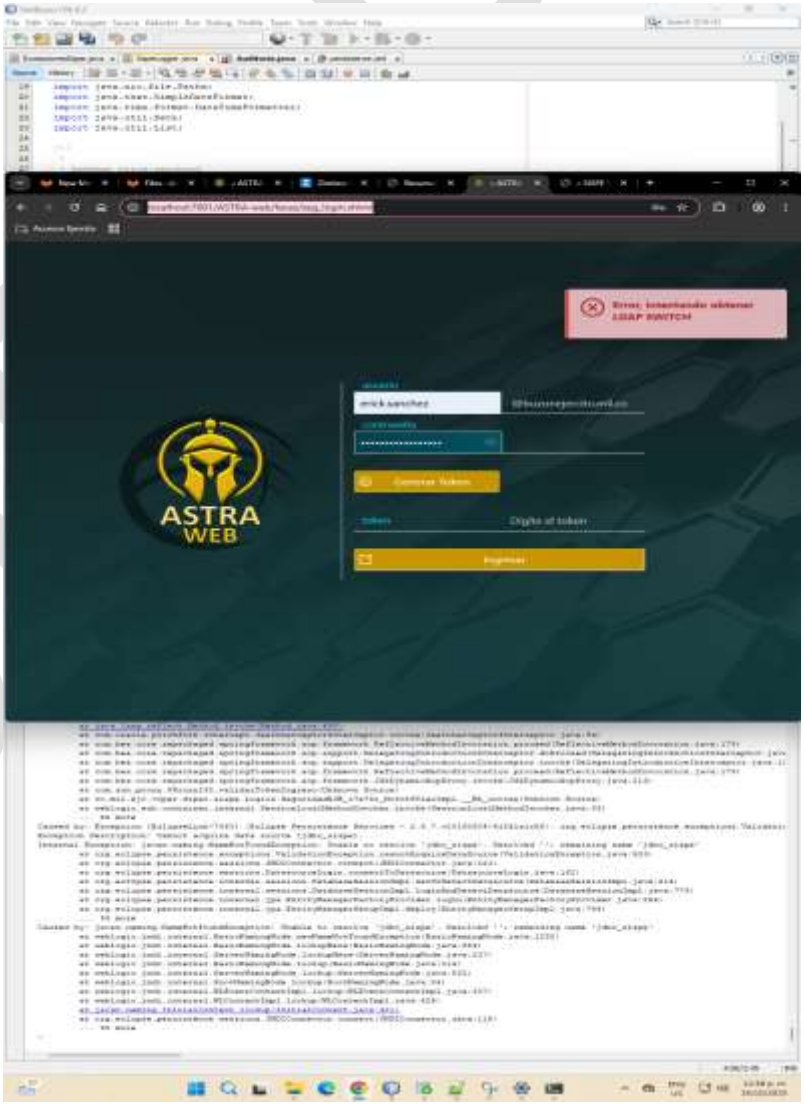


Imagen 1.

5. Realizar todos los demás ajustes adicionales a las formularios, reportes, procesos y procedimientos requeridos por el supervisor del contrato cuando se presente la necesidad inmediata relacionada con el funcionamiento de los sistemas de información bajo la tecnología java.

6. PROBLEMAS CON LA CREACIÓN Y CONFIGURACIÓN DE LA NUEVA BASE DE DATOS SISPE

Durante el proceso de implementación del nuevo sistema **SISPE**, se presentaron diversas incidencias técnicas relacionadas con la creación y conexión a la nueva base de datos. Inicialmente, se intentó realizar una copia completa de la base de datos "ASTRA", con el propósito de mantener una estructura idéntica en cuanto a tablas, vistas, secuencias, índices y demás objetos de esquema. Sin embargo, surgieron problemas de permisos al intentar realizar la exportación e importación (operaciones expdp / impdp o exp / imp en Oracle), los cuales impedían acceder a determinados objetos y esquemas dependientes.

En entornos Oracle, las bases de datos pueden contener múltiples esquemas, cada uno con su propio conjunto de objetos (tablas, vistas, procedimientos, triggers, secuencias, etc.). Estos esquemas son equivalentes a usuarios con privilegios específicos, y su correcta migración es fundamental para garantizar que todas las referencias entre objetos se mantengan operativas.

En este caso, la base de datos ASTRA incluía objetos pertenecientes a otros esquemas **(NUEVO_REHU y COPER_ADMINSEG)** que eran referenciados directamente por el esquema principal del sistema. La falta de exportación de estos esquemas generó errores de integridad referencial y de acceso a objetos inexistentes al crear la base de datos SISPE.

Posteriormente, una vez creada la base de datos SISPE, se presentaron problemas de conexión desde la aplicación, relacionados con la configuración de acceso al servidor de base de datos Oracle. Dichos problemas estaban asociados tanto a credenciales de autenticación incorrectas como a diferencias en los parámetros de conexión, específicamente en el nombre del servicio (SERVICE_NAME) y el puerto de escucha (PORT).

7. PARÁMETROS DE CONEXIÓN EN ORACLE

Una conexión estándar a una base de datos Oracle requiere la definición de los siguientes elementos:

1. **Usuario (USERNAME):** Corresponde al esquema al que se accederá dentro de la base de datos.
2. **Contraseña (PASSWORD):** Clave asociada al usuario, necesaria para la autenticación.

3. **Host:** Dirección IP o nombre del servidor donde se encuentra alojada la base de datos.

4. **Puerto (PORT):** Generalmente el 1521, aunque puede variar según la configuración del listener de Oracle.

5. **Nombre del servicio (SERVICE_NAME):** Identificador lógico que representa una base de datos o instancia Oracle específica dentro del servidor.

En algunos entornos también se utiliza el SID (System Identifier), aunque en versiones más recientes de Oracle se recomienda el uso del SERVICE_NAME, especialmente cuando se manejan configuraciones con Oracle RAC (Real Application Clusters) o servicios balanceados.

Un ejemplo típico de cadena de conexión Oracle es:

jdbc:oracle:thin:@// 172.22.3.116:1521/SISPE
donde:

- oracle:thin indica el driver JDBC en uso.
- 172.22.3.116 corresponde al host o IP del servidor.
- 1521 es el puerto del listener.
- SISPE es el SERVICE_NAME o identificador de la base de datos.

8. SOLUCIÓN ADOPTADA

Para resolver los problemas de acceso, se solicitaron nuevas credenciales con privilegios adecuados y se actualizó la configuración del archivo de conexión (tnsnames.ora o equivalentes en el servidor de aplicaciones). Esto permitió establecer una conexión exitosa entre el sistema SISPE y la nueva base de datos Oracle.

Adicionalmente, se verificó la correcta asociación de los roles y permisos de los usuarios para garantizar que los objetos (tablas, vistas, procedimientos y secuencias) pudieran ser accedidos sin restricciones desde la aplicación.

Este ajuste también permitió la reconfiguración de los DataSources en el servidor de aplicaciones (WebLogic en este caso), asegurando la compatibilidad de las cadenas de conexión con el nuevo entorno, y garantizando la estabilidad en las operaciones de lectura y escritura de datos.

DIPER

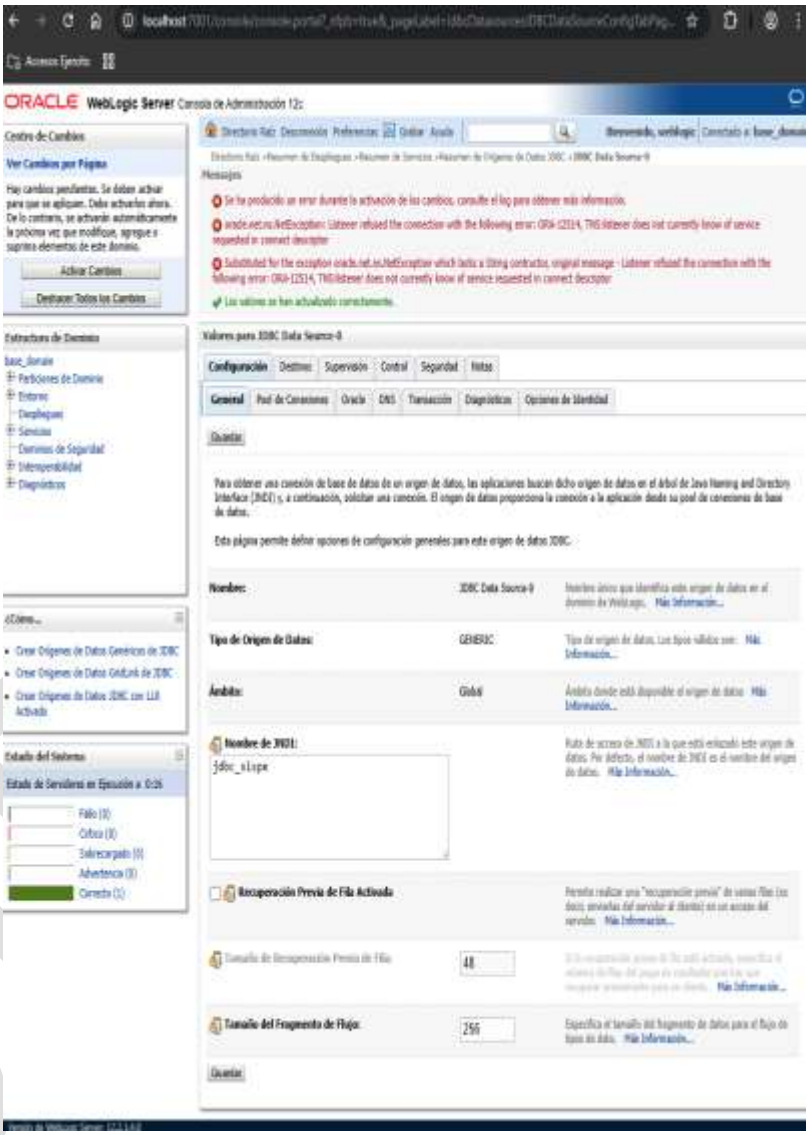


Imagen 2.

3. Realizar propuestas de solución de acuerdo a los requerimientos para ajustes de presentación de datos en los diferentes módulos reportes de los sistemas de información bajo la tecnología java.

9. REVISIÓN FUNCIONAL Y CORRECCIÓN DE ERRORES EN LA VISUALIZACIÓN DE INFORMACIÓN DE CANDIDATOS

Durante la fase de pruebas del sistema ASTRA, se identificó un incidente relacionado con la visualización de la información de los candidatos en el

	entorno de pruebas. En este ambiente, los datos cargados no eran completamente equivalentes a los del entorno productivo, lo cual dificultaba la reproducción de algunos comportamientos esperados de la aplicación. Para realizar un análisis detallado, se procedió a habilitar las trazas de depuración (logs) desde la interfaz gráfica del sistema, activando específicamente los registros asociados al módulo de visualización de candidatos. Esto permitió capturar en tiempo real los mensajes de error y advertencia generados por la aplicación al intentar renderizar la información proveniente de la base de datos. Durante esta revisión se detectó que el error no afectaba a todos los registros, sino únicamente a ciertos candidatos que poseían información adicional o estructuras de datos incompletas. En términos técnicos, el problema radicaba en porciones de código que no validaban adecuadamente la existencia o la integridad de determinados campos antes de procesarlos o mostrarlos en pantalla. En consecuencia, al encontrarse con valores nulos o datos inesperados, el sistema arrojaba excepciones que interrumpían la carga de información, afectando parcial o totalmente la visualización de otros candidatos. Con base en los hallazgos, se implementaron ajustes en el código fuente para corregir la causa raíz del problema, introduciendo mecanismos de control de excepciones y validaciones previas a nivel de lógica de negocio y capa de presentación. Estos cambios permitieron: 1. Manejar de forma segura los datos faltantes o inconsistentes sin detener la ejecución del componente. 2. Asegurar que la interfaz continúe operando incluso si uno o varios registros presentan errores de integridad. 3. Registrar en los logs los casos anómalos sin afectar la experiencia general del usuario. 4. Incrementar la resiliencia de la interfaz de visualización, evitando que un error puntual en un conjunto de datos comprometa la funcionalidad completa del módulo. El resultado fue una interfaz a prueba de fallos en condiciones de datos incompletos, manteniendo la integridad visual y funcional del módulo de candidatos. Asimismo, se documentaron los cambios aplicados y se verificó la correcta visualización de los registros tras la actualización, confirmando la estabilidad del componente en las pruebas posteriores.

DIPER

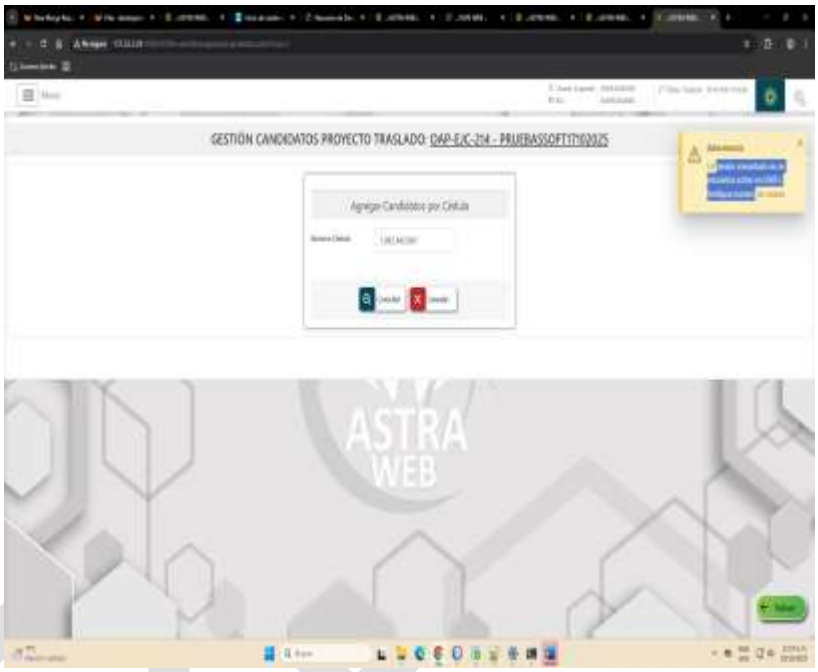


Imagen 3.

10. CONFIGURACIÓN DE SEGUNDA FUENTE DE DATOS (JDBC) PARA ACCESO AL ESQUEMA COPER_ADMINSEG

Como parte de la integración entre el sistema SISPE y los datos heredados del sistema ASTRA, fue necesario establecer una segunda fuente de datos (data source) que permitiera el acceso directo al esquema COPER_ADMINSEG, alojado en el mismo servidor Oracle pero bajo un contexto distinto.

En Oracle, cada esquema mantiene su propio espacio de objetos (tablas, vistas, secuencias, procedimientos, etc.), por lo que acceder a un esquema diferente requiere una conexión explícita que preserve su contexto de seguridad, permisos y credenciales.

Dado que el nuevo sistema SISPE debía consultar información existente en dicho esquema sin replicarla físicamente, se optó por configurar un segundo origen de conexión JDBC.

11. DETALLES TÉCNICOS DE LA CONFIGURACIÓN

La implementación de esta segunda fuente de datos implicó ajustes tanto en la configuración del servidor de aplicaciones como en el código del sistema

Java:

1. Configuración en WebLogic (JDBC Data Source):

   Se creó un nuevo recurso JDBC, denominado por convención jdbc/COPER_ADMINSEG_DS, configurado con los parámetros de conexión específicos del esquema.

2. Cambios en el código Java:

   Fue necesario modificar la capa de persistencia para permitir que ciertos *repositories* o *DAO* se conectaran explícitamente al nuevo origen de datos.
   Esto se logró mediante la creación de una unidad de persistencia secundaria (Persistence Unit) en el archivo persistence.xml, con un nombre distinto, por ejemplo:

3. <persistence-unit name="COPER_ADMINSEG_PU" transaction-type="JTA">
4. <jta-data-source>jdbc/COPER_ADMINSEG_DS</jta-data-source>
5. <properties>
6. <property name="hibernate.dialect" value="org.hibernate.dialect.Oracle12cDialect"/>
7. <property name="hibernate.show_sql" value="false"/>
8. </properties>
9. </persistence-unit>

En el código Java, las clases que requerían acceso a este esquema se anotaron con la unidad de persistencia correspondiente:

```
@PersistenceContext(unitName = "COPER_ADMINSEG_PU")
private EntityManager emCoper;
```

De esta forma, se aislaron las operaciones que debían ejecutarse sobre COPER_ADMINSEG, manteniendo la independencia entre las transacciones y los contextos de sesión de ambos esquemas.

10. Configuraciones XML y Spring (en caso de uso):
    En sistemas que utilizan Spring o frameworks similares, se añadieron las definiciones de *bean* para la segunda fuente de datos, estableciendo las propiedades de conexión, gestión de transacciones y EntityManagerFactory independientes del origen principal (jdbc/SISPE_DS).

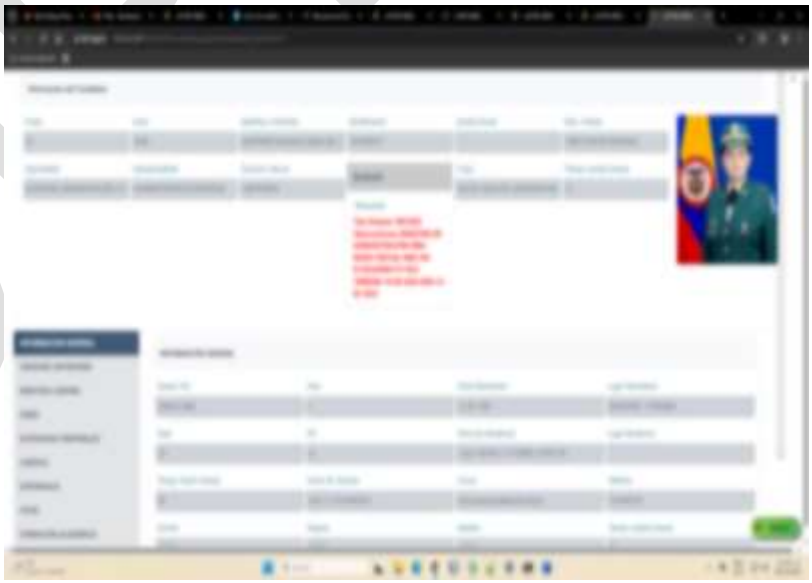
DIPER

12. BENEFICIOS Y JUSTIFICACIÓN DE LA ESTRATEGIA

La configuración de esta segunda fuente de datos trajo las siguientes ventajas técnicas:

1. Evita la duplicación de datos entre esquemas, reduciendo espacio y riesgos de desincronización.
2. Asegura la trazabilidad de la información, dado que los datos sensibles de seguridad permanecen en su esquema original (COPER_ADMINSEG).
3. Permite independencia lógica y de mantenimiento, ya que cada sistema (ASTRA y SISPE) conserva su propio contexto transaccional, pero pueden interoperar bajo un modelo controlado.
4. Cumple buenas prácticas de integración entre sistemas Oracle y aplicaciones Java EE, mediante uso de JTA y EntityManager dedicados.

Esta modificación, aunque requirió ajustes de código y configuración XML, no afectó la arquitectura general del sistema SISPE, sino que la extendió, proporcionando una base más robusta para futuras integraciones con otros esquemas o servicios de datos.



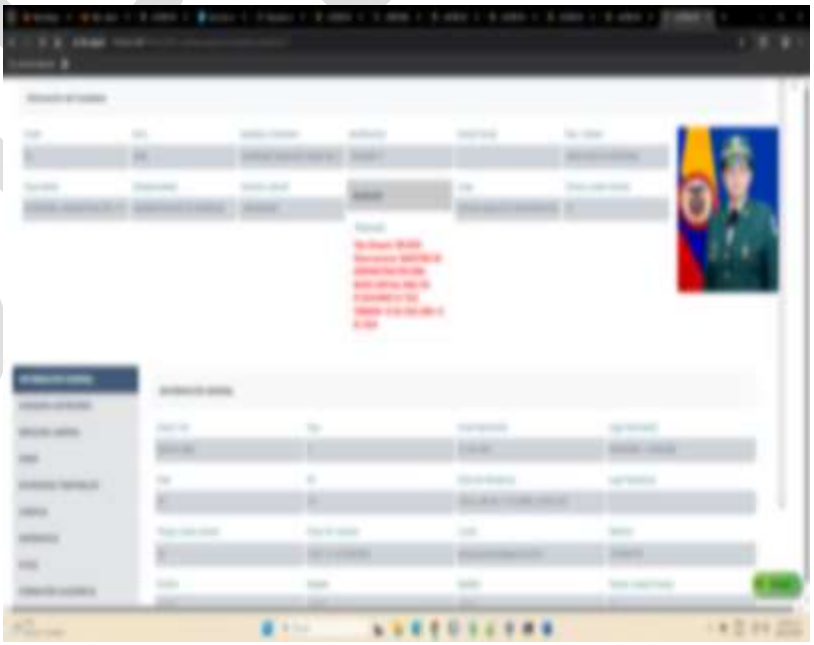
Producción, probando, funciona
http://172.22.2.29:7005/ASTRA-web/faces/gestionCandidatos.xhtml?cid=5
Imagen 4.

DIPER



http://172.22.2.28:7005/ASTRA-web/faces/gestionCandidatos.xhtml?cid=2

Imagen 5.



http://172.22.2.103:7005/ASTRA-web/faces/gestionCandidatos.xhtml?cid=2

Imagen 6.

DIPER



Imagen 7.



Imagen 8.

SISPE



Imagen 9.

Nueva base de datos funcionando

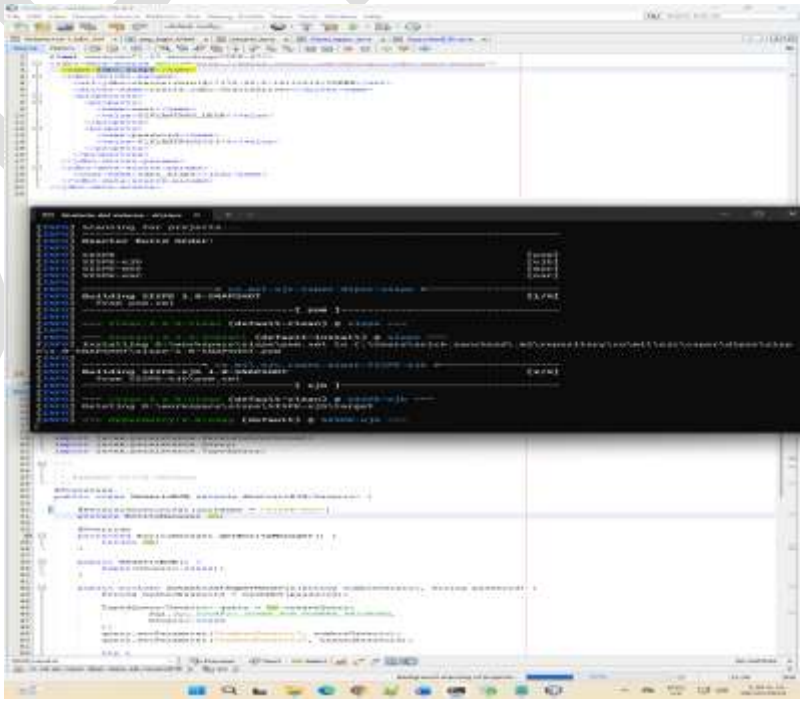


Imagen 10.

13. INTEGRACIÓN ENTRE RAMAS GIT (SISPE3 Y DEVELOPMENT)

Durante el proceso de control de versiones y consolidación del código fuente, se llevó a cabo una integración entre las ramas sispe3 y development dentro del repositorio Git del proyecto.

Esta operación representó un alto grado de complejidad técnica, debido a conflictos derivados de la inclusión inadvertida de carpetas y archivos generados automáticamente por el entorno de compilación Java, principalmente los contenidos en las carpetas target/.

- **Contexto técnico**

En proyectos basados en Maven o Gradle, las carpetas target/ contienen los archivos compilados (.class), artefactos temporales, dependencias descargadas y empaquetados transitorios (JAR, WAR) que no deben ser versionados.

El control de versiones Git debe manejar únicamente el código fuente, recursos de configuración y scripts de construcción, pero no los binarios ni archivos temporales generados por el compilador.

En este caso, se identificó que algunos de estos artefactos habían sido incluidos en commits previos, lo cual produjo conflictos de fusión (merge conflicts) entre las ramas, además de un incremento innecesario del tamaño del repositorio y posibles inconsistencias al cambiar de entorno o rama.

14. MEDIDAS CORRECTIVAS Y AJUSTES IMPLEMENTADOS

Para resolver la problemática, se ejecutaron las siguientes acciones:

- **Depuración del contenido versionado:**

Se eliminaron del repositorio las carpetas target/ y sus subdirectorios de todas las ramas afectadas mediante comandos git rm -r --cached target, asegurando que dichos archivos no volvieran a ser rastreados por Git.

- **Actualización del archivo. gitignore:**

Se ampliaron las reglas de exclusión para evitar la inclusión de:

Carpetas target/ y subcarpetas.
Archivos .class (productos de compilación).

Archivos temporales de IDE (.settings/, .classpath, .project de Eclipse o .idea/ de IntelliJ).

Archivos generados por compilaciones parciales o pruebas unitarias (*.log, *.tmp, *.bak).

DIPER

Ejemplo de las reglas agregadas:

/target/
**/target/
*.class
*.log
*.tmp
.idea/
.settings/

- **Anotaciones y documentación en el proyecto:**

Se documentó en el README.md y en los manuales de desarrollador la política de exclusión de archivos temporales, además de añadir anotaciones y comentarios en los scripts de build para evitar futuras inclusiones accidentales.

- **Validación post-integración:**

Una vez consolidada la fusión de ramas, se realizaron pruebas de compilación y despliegue para verificar que:

El proyecto compila correctamente en limpio (mvn clean install).

No existen referencias a archivos obsoletos o duplicados.

El árbol de trabajo del repositorio se mantiene liviano y coherente entre entornos.

- **Gracias a la aplicación de estas medidas, se logró:**

1. Restablecer la integridad y consistencia entre las ramas sispe3 y development.
2. Garantizar que el repositorio mantenga únicamente el código fuente y configuraciones necesarias.
3. Reducir el riesgo de conflictos futuros durante integraciones o despliegues.
4. Mejorar la trazabilidad del código y la eficiencia de los procesos de build y merge en entornos de desarrollo colaborativo.

DIPER


Compressing objects: 100% (12/12), done.
Writing objects: 100% (10/10), 1.20 KiB | 110.00 KiB/s, done.
Total 10 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: To create a merge request for development, visit:
remote: http://repositorioGit.ejercito.mil.co/erick.sanchezd/sispe/merge_requests/new?merge_request%5Bsource_br
ch%5D=development
remote:
To repositorioGit.ejercito.mil.co:erick.sanchezd/sispe.git
cf37f34..3881a6f development -> development
D:\eeeWspace\sispe>git status
On branch development
Your branch is up to date with 'origin/development'.
nothing to commit, working tree clean
D:\eeeWspace\sispe>git pull
Already up to date.
D:\eeeWspace\sispe>git checkout feature/sispe3
Updating files: 100% (757/757), done.
Switched to branch 'feature/sispe3'
D:\eeeWspace\sispe>git checkout development
Updating files: 100% (757/757), done.
Switched to branch 'development'
Your branch is up to date with 'origin/development'.
D:\eeeWspace\sispe>git checkout feature/sispe4
error: pathspec 'feature/sispe4' did not match any file(s) known to git
D:\eeeWspace\sispe>git checkout -b feature/sispe4
Switched to a new branch 'feature/sispe4'
D:\eeeWspace\sispe>git status
On branch feature/sispe4
nothing to commit, working tree clean
D:\eeeWspace\sispe>git rm -r --cached SISPE-ear/target SISPE-ejb/target SISPE-web/target

Imagen 11.

15. ERRORES Y PROBLEMAS DE ALTA COMPLEJIDAD EN EL PROCESO DE AUTENTICACIÓN (LOGIN)

Durante la fase de pruebas e integración del sistema SISPE, se presentaron errores intermitentes y de difícil trazabilidad relacionados con el proceso de inicio de sesión (login) de usuarios.

El comportamiento observado consistía en que, de manera no consistente, el sistema no lograba recuperar todos los datos asociados al usuario autenticado, específicamente la carga de roles, asignaciones de roles por usuario y permisos de acceso a proyectos.

- **Descripción técnica del problema**

El sistema realiza el proceso de autenticación validando las credenciales del usuario y posteriormente cargando la información complementaria desde distintas tablas relacionales (USUARIO, ROL, USUARIO_ROL_PROYECTO, entre otras).

En ciertos casos, aunque el usuario lograba autenticarse exitosamente, el contexto de sesión no contenía todos los objetos esperados (por ejemplo, faltaban roles, permisos o asociaciones a proyectos), lo que provocaba comportamientos erráticos en la interfaz y errores en la autorización de módulos.

El problema presentaba una alta complejidad técnica debido a que la causa no era evidente ni reproducible en todos los entornos. En algunos casos, el fallo estaba relacionado con tiempos de respuesta variables en la base de datos o inconsistencias temporales en las asociaciones de usuario-rol-proyecto, y en otros con la forma en que el framework gestionaba las transacciones y sesiones en paralelo durante la autenticación.

- **Estrategia de diagnóstico y trazabilidad**

Ante la falta de consistencia en los resultados, se optó por una estrategia de instrumentación intensiva del código, agregando múltiples líneas de trazado (logging) para registrar en detalle cada paso del flujo de autenticación.

Estas trazas fueron implementadas tanto en la capa de servicios (EJB/Service) como en la capa DAO o de persistencia, utilizando niveles de log diferenciados (INFO, DEBUG, ERROR) con el objetivo de obtener una lectura clara y secuencial del proceso.

Entre los puntos críticos trazados se incluyeron:

1. Inicio del proceso de login: registro de credenciales recibidas y método de autenticación utilizado (interno o LDAP).
2. Carga de usuario: validación de existencia en la tabla USUARIO y estado activo.

DIPER

3. Consulta de roles base (ROL) y relaciones USUARIO_ROL_PROYECTO.
4. Inicialización de sesión y contexto de seguridad (SecurityContext / HttpSession).
5. Verificación de integridad de datos cargados antes de continuar al dashboard principal.

Cada traza fue diseñada para mostrar, además del resultado, los tiempos de ejecución, identificadores de sesión, usuario en contexto y transacción activa, con el fin de detectar posibles inconsistencias entre llamadas concurrentes o sesiones en conflicto.

- **Ajustes de código y solución implementada**

Tras analizar los registros obtenidos, se detectaron varios factores que contribuían a la inconsistencia, entre ellos:

1. Dependencias no sincronizadas entre consultas JPA, ejecutadas sin asegurar la finalización de la transacción anterior.
2. Datos incompletos en la tabla intermedia USUARIO_ROL_PROYECTO para algunos usuarios de prueba.
3. Problemas de cacheo o retención de entidades en la capa de persistencia (EntityManager), lo cual generaba retornos parciales en consultas subsecuentes.

Para corregir el problema se implementaron los siguientes ajustes:

1. Refactorización de la secuencia de carga de datos del usuario, garantizando que las consultas a roles y permisos se ejecutaran dentro de la misma transacción activa.
2. Limpieza y revalidación de las relaciones de usuario-rol en la base de datos de pruebas.
3. Inclusión de validaciones adicionales antes de iniciar la sesión para asegurar que la información mínima de roles esté disponible.
4. Mejoras en la gestión del **contexto de seguridad**, asegurando que los roles se asocien correctamente al Principal de la sesión activa.

- **Resultados**

Tras los cambios, el proceso de autenticación mostró mayor estabilidad y consistencia. Las trazas agregadas permanecen activas en modo DEBUG para futuros diagnósticos, y el sistema ahora cuenta con una capa de registro más robusta y granular, lo que facilita el seguimiento y resolución de

DIPER

incidencias relacionadas con seguridad o acceso.



Imagen 12.

Superados los inconvenientes relacionados con los datos y la carga de información, el sistema permitió el ingreso correctamente. Sin embargo, se identificaron problemas residuales en los estilos visuales y en la resolución de fuentes de datos. Se evidenció que ciertas referencias en las constantes del sistema aún apuntaban al nombre anterior ASTRA, lo cual afectaba la correcta aplicación de los estilos y rutas de recursos visuales bajo la nueva denominación SISPE.

Para mitigar esta situación, se realizó una revisión exhaustiva de las hojas de estilo (CSS) y de los archivos de configuración asociados, ajustando las referencias a imágenes, iconografía y recursos gráficos, de modo que apuntaran a las nuevas rutas y elementos visuales definidos para SISPE. Con ello se busca mantener la coherencia visual, la correcta carga de recursos estáticos y la identidad gráfica del nuevo entorno.



Imagen 13.

Se logró obtener un login completamente funcional con menú operativo, tras un proceso iterativo en el que se subieron, probaron y depuraron seis versiones sucesivas del software, enfocadas en la resolución de errores y la estabilización de las funcionalidades principales del sistema.

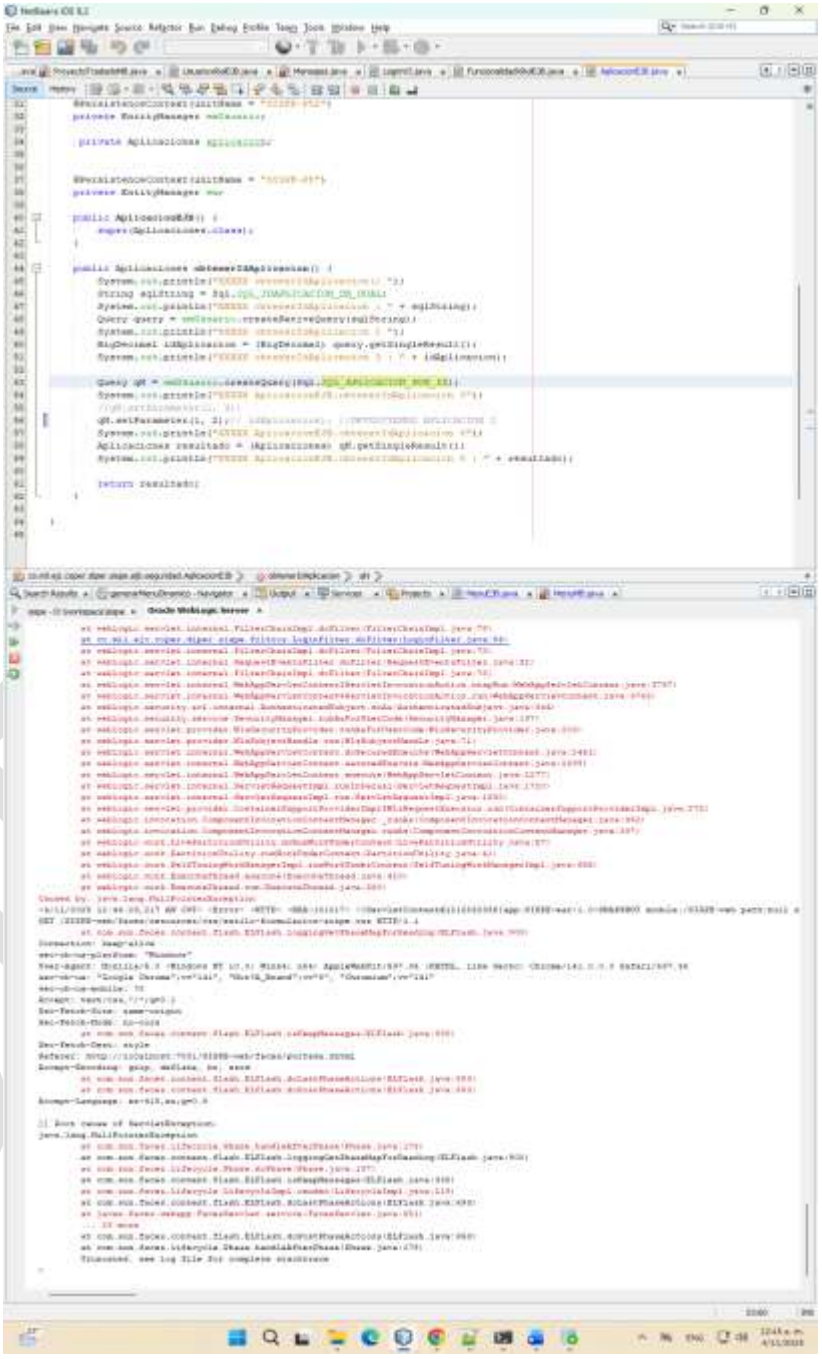
16. CONCLUSIONES ADICIONALES DEL AJUSTE DE ASTRA A SISPE

Durante el proceso de transición y adaptación del sistema ASTRA hacia SISPE, se identificaron varios aspectos críticos de carácter estructural:

1. Estructura y denominación de las tablas base: Se evidenció que las tablas principales del sistema requieren modificaciones tanto en su estructura como en sus nombres, con el fin de alinearlas con la nueva arquitectura y nomenclatura de SISPE. Estas diferencias han generado inconsistencias entre el modelo de datos y las capas superiores de la aplicación.
2. Necesidad de refactorización integral: Cada cambio estructural en las tablas implica la necesidad de realizar un refactor en todas las capas del software (entidades JPA, servicios, controladores y vistas), debido a la fuerte dependencia del código respecto a los nombres y estructuras originales de ASTRA.
3. Impacto en las relaciones foráneas y consultas tipadas: Los mayores problemas se han presentado en las tablas que mantienen relaciones foráneas con las estructuras originales, ya que existen numerosas consultas fuertemente tipadas (tanto en JPA como en SQL nativo) que dependen de los nombres antiguos. Esto ha generado errores en tiempo de ejecución y dificultades para garantizar la compatibilidad total entre módulos.

En conjunto, los hallazgos confirman la necesidad de planificar una migración controlada y progresiva del modelo de datos, acompañada de una estrategia de refactorización sistemática que asegure la estabilidad funcional del sistema SISPE a largo plazo.

DIPER

	 Imagen 14.
12. VALOR DEL CONTRATO	Valor total: Valor: $56.100.000.00 Valor autorizado a pagar: $5.100.000.00

<table>
<tr><td>13. CUMPLIMIENTO PAGO SEGURIDAD SOCIAL</td><td>
<table>
<tr><th>Obligación</th><th>Entidad</th><th>Valor Pago</th></tr>
<tr><td>PENSIÓN</td><td>PORVENIR</td><td>$326.400</td></tr>
<tr><td>SALUD</td><td>SALUD TOTAL</td><td>$255.000</td></tr>
<tr><td>ARL</td><td>POSITIVA</td><td>$10.649</td></tr>
</table>
</td></tr>
<tr><td>14. DOCUMENTOS QUE HACEN PARTE DEL INFORME</td><td>1. Cuenta de cobro

2. Informe de gestión sobre el cumplimiento del objeto contractual, firmado por el CONTRATISTA y el recibido a satisfacción del supervisor del contrato, de acuerdo al formato establecido.

3. Planilla de seguridad social: copia discriminada de los soportes de pago de los aportes mensuales obligatorios al sistema general de seguridad social en salud y pensión, sobre un ingreso base de cotización del 40% del valor mensual del contrato, con aporte del 16% en pensiones y el 12,5% en salud, ARL 0,522%

Nota: Lo anterior, conforme lo establece la normatividad legal vigente, la reglamentación de la materia y de acuerdo con la afiliación efectuada a la A.R.L.

4. Comprobante de pago de la planilla de seguridad social

5. Pantallazo SECOP II – Numeral 7 actualizado</td></tr>
<tr><td>15. CONSTANCIAS</td><td>En el presente informe se deja constancia que el/la contratista cumplió con las obligaciones contractuales y legales reguladas para este tipo de contrato y como consecuencia se recibe a satisfacción de forma PARCIAL __X__ TOTAL

Así mismo, se deja constancia que durante la ejecución del contrato no se han configurado los presupuestos legales que regulan el contrato laboral.</td></tr>
<tr><td>16. RECOMENDACIONES</td><td>Se sugiere mantener los servicios profesionales de ingeniería de sistemas para dar soporte a los sistemas de información de talento humano que administra el Comando de Personal a través de la Dirección de Personal, en este caso el Sistema de Información ASTRA.</td></tr>
<tr><td>17. CONCLUSIONES</td><td>Teniendo en cuenta mejoras y ajustes realizados al Sistema ASTRA del Ejército Nacional, se requiere de los servicios profesionales de ingeniería de sistemas para atender los requerimientos técnicos con respecto a dicho Sistema de Información para los traslados del personal y el desarrollo de nuevos ajustes y mejoras para los usuarios funcionales de Ejército.</td></tr>
</table>

Para constancia se firma en, **(Bogotá),** en el mes de **NOVIEMBRE 2025**

FIRMA SUPERVISOR
Contrato No.125-CENACPERSONAL-2025
Grado y Nombre completo: **CT. ALVARADO MORATO JONATAN ANDRÉS**
Cargo: Oficial Administrador Informático
Resolución de nombramiento: Resolución de Delegación RES_00000352 del 07 de enero de 2025