



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



## TFG del Grado en Ingeniería Informática

**Gestión de Contratos GeNomIn  
Aplicación APEX sobre Oracle, para la  
gestión de contratos del Personal  
contratado con cargo a proyectos de  
Investigación**



Presentado por Francisco J. Arroyo Redondo  
en Universidad de Burgos — Septiembre 2025  
Tutor: Pedro Renedo Fernández



---

# Índice general

---

<b>Índice general</b>	i
<b>Índice de figuras</b>	iv
<b>Índice de tablas</b>	vi
<b>Apéndice A Plan de Proyecto Software</b>	1
A.1. Introducción . . . . .	1
A.2. Plan temporal . . . . .	1
Kanban . . . . .	2
Milestone . . . . .	2
Sprints . . . . .	3
A.3. Estudio de Viabilidad . . . . .	7
Viabilidad Técnica . . . . .	7
Viabilidad Económica . . . . .	8
Viabilidad Legal . . . . .	8
Viabilidad de Implantación . . . . .	8
Conclusiones . . . . .	9
<b>Apéndice B Requisitos y Casos de Uso</b>	11
B.1. Introducción . . . . .	11
B.2. Requisitos Funcionales . . . . .	13
RF01: . . . . .	13
RF02: . . . . .	14
RF03, RF04, RF05 y RF06: . . . . .	15
RF07: . . . . .	16
RF08: . . . . .	17

RF09:	18
RF10:	19
RF11:	20
RF12:	21
RF13:	22
RF14:	23
RF15, RF16 y RF17:	24
RF18, RF19 y RF20:	25
RF21, RF22 y RF23:	26
RF24:	27
RF25:	28
RF26:	29
RF27:	30
<b>Apéndice C Diseño GeNomIn</b>	<b>33</b>
C.1. Introducción	33
C.2. App Builder	33
C.3. Página Proyectos	34
C.4. Página Responsables	35
C.5. Página Mantenimiento Convocatorias	35
Crear Convocatoria	35
C.6. Página de Solicitantes	37
Crear solicitante	37
C.7. Página de Contratos	39
Página Nuevo Contrato	39
Página Renovación	43
Página Renuncia al contrato	49
C.8. Página de Informes	53
Página Nómina-mes	54
Página Vencimientos	55
Página Contratos	56
Página Nóminas-Periodo	56
<b>Apéndice D Configuración</b>	<b>59</b>
D.1. Oracel 23ai	59
D.2. Oracle REST Data Service	61
D.3. Apex Office Print AOP	65
D.4. Oracle Cloud Infraestructura	67
<b>Apéndice E Manual de Usuario</b>	<b>71</b>
E.1. Tabla de contenido	71

<i>Índice general</i>	III
E.2. Objetivo y proceso . . . . .	72
E.3. Inicio de Sesión: . . . . .	73
E.4. Administración . . . . .	73
Responsables . . . . .	74
Proyectos . . . . .	75
E.5. Convocatorias . . . . .	75
Solicitante . . . . .	76
E.6. Contratos . . . . .	78
Crear . . . . .	78
Renovación . . . . .	80
Renuncia . . . . .	82
E.7. Informes . . . . .	83
Nómina-mes . . . . .	83
Vencimientos . . . . .	85
Contratos . . . . .	86
Nominas-Periodo . . . . .	86
E.8. Acerca de . . . . .	87
Guía de uso . . . . .	87
<b>Apéndice F Competencias de Sostenibilidad Curricular</b>	<b>89</b>
F.1. Competencias Adquiridas . . . . .	89
Uso sostenible de recursos . . . . .	89
Participación comunitaria . . . . .	90
Principios éticos . . . . .	90
F.2. Aplicación en el Proyecto GeNomIn . . . . .	90
Diseño y Funcionalidad . . . . .	90
Impacto Ambiental y Social . . . . .	90
Conciencia ambiental . . . . .	90
F.3. Conclusión . . . . .	91
<b>Bibliografía</b>	<b>97</b>

---

# Índice de figuras

---

A.1. Detalle del tablero kanban al inicio del proyecto . . . . .	2
A.2. Milestone: Kick-Off . . . . .	3
A.3. Milestone: Prototipo . . . . .	4
A.4. Mileston: MPV . . . . .	5
A.5. Milestone: Desarrollo Completo . . . . .	5
A.6. Milestone: Documentación . . . . .	6
A.7. Gráfica de Velocidad de los Sprints . . . . .	6
A.8. Tarea #2:Instalación de herramientas en GitHub . . . . .	7
B.1. Diagrama de secuencia del proceso General . . . . .	12
B.2. Informe Allure de Test . . . . .	31
C.1. Creación de app APEX . . . . .	34
D.1. Configuración conexión a FREEPDB1 . . . . .	60
D.2. Configuración Listener a FREEPDB1 . . . . .	60
D.3. Servicios activados de Oracle . . . . .	60
D.4. Verificación de Listner activo . . . . .	61
D.5. Directorio de instalación ORDS . . . . .	62
D.6. Fichero de configuración pool.xml . . . . .	63
D.7. Configuración del fichero settings.xml . . . . .	64
D.8. Imagen de acceso web a servicios ORDS . . . . .	64
D.9. Vista de la instalación de plug-ins de AOP . . . . .	65
D.10. Confiuguración de los plug-ins AOP . . . . .	66
D.11. Visualización de la plantilla AOP . . . . .	66
D.12. Asociación de la plantilla al informe . . . . .	67
D.13. Menu Oracle Database Infraestructure . . . . .	68
D.14. Creacion de la Instancia de BD-1 en Cloud . . . . .	68

*Índice de figuras*

v

D.15.Creacion de la Instancia de BD-1 en Cloud . . . . .	69
D.16.Creación de Instancia APEX en Cloud . . . . .	70

---

# Índice de tablas

---

A.1. Herramientas utilizadas en el desarrollo del proyecto . . . . .	7
A.2. Coste económico GeNomIn . . . . .	8
B.1. Caso de prueba US-01 . . . . .	13
B.2. Caso de prueba US-02 . . . . .	14
B.3. Caso de prueba US-03/06 . . . . .	15
B.4. Caso de prueba US-07 . . . . .	16
B.5. Caso de prueba US-08 . . . . .	17
B.6. Caso de prueba US-09 . . . . .	18
B.7. Caso de prueba US-10 . . . . .	19
B.8. Caso de prueba US-11 . . . . .	20
B.9. Caso de prueba US-12 . . . . .	21
B.10.Caso de prueba US-13 . . . . .	22
B.11.Caso de prueba US-14 . . . . .	23
B.12.Caso de prueba US-15, US-16 y US-17 . . . . .	24
B.13.Caso de prueba US-18, US-19 y US-20 . . . . .	25
B.14.Caso de prueba US-21, US-22 y US-23 . . . . .	26
B.15.Caso de prueba US-24 . . . . .	27
B.16.Caso de prueba US-25 . . . . .	28
B.17.Caso de prueba US-26 . . . . .	29
B.18.Caso de prueba US-27 . . . . .	30

---

# Índice de Códigos

---

C.1. Informe de todos los Proyectos . . . . .	34
C.2. Informe de todos los Responsables . . . . .	35
C.3. Informe de todas las Convocatorias . . . . .	35
C.4. Presenta de todos las Investigadores . . . . .	36
C.5. Presenta informe proyectos abiertos para el investigador seleccionado . . . . .	36
C.6. Control y aviso de proyecto abiertos de investigador seleccionado . . . . .	36
C.7. Informa de la duración restante del proyecto elegido . . . . .	36
C.8. Presenta informe de solicitantes . . . . .	37
C.9. Presenta informe de solicitantes . . . . .	37
C.10.Comprueba campo abierto de convocatoria . . . . .	38
C.11.Control y aviso de convocatoria cerrada . . . . .	38
C.12.Comprueba campo de titulación requerida . . . . .	38
C.13.Control y aviso de titulación insuficiente . . . . .	38
C.14.Comprueba campo de titulación requerida . . . . .	39
C.15.Carga fecha inicio proyecto . . . . .	40
C.16.Carga fecha fin proyecto . . . . .	40
C.17.Fuerza nueva petición de fechas si son erróneas . . . . .	40
C.18.Controla que la fecha inicio sea posterior al inicio del proyecto . . . . .	40
C.19.Controla que la fecha fin sea posterior a inicio y anterior al fin del proyecto . . . . .	41
C.20.Proceso que genera las nóminas relativas a un nuevo contrato . . . . .	41
C.21.Selección de contratos abiertos . . . . .	43
C.22.Obtención de datos del contrato a renovar . . . . .	43
C.23.Obtención de fecha fin del contrato a renovar . . . . .	44
C.24.Verifica la fecha de renovación se posterior a la actual y anterior a fin proyecto . . . . .	44

C.25.Reactiva fecha renovación se es errónea . . . . .	46
C.26.Genera las nuevas nóminas tras la renovación . . . . .	46
C.27.Guarda seguridad social en campo oculto . . . . .	49
C.28.Verifica las fecha de renuncia del contrato . . . . .	49
C.29.Genera las nuevas nóminas tras la renuncia . . . . .	52
C.30.Consulta informe nómina para un mes determinado . . . . .	54
C.31.Guarda y muestra el total del listado . . . . .	54
C.32.Muestra las referencias de proyectos del informe para seleccionar . . . . .	55
C.33.Muestra de contratos que vencen en periodo de fechas . . . . .	55
C.34.Consulta para todos los contratos . . . . .	56
C.35.Consulta de contratados para selección . . . . .	56
C.36.Consulta de nóminas para un contratado en un intervalo de tiempo . . . . .	57
D.1. Creación de Esquema TFGUBU . . . . .	59

## *Apéndice A*

---

# **Plan de Proyecto Software**

---

## **A.1. Introducción**

La realización de la planificación de este proyecto software ha sido gestionada a través de la **Metodología ágil** dividiendo el trabajo en hitos, **Milestones**, y éstos a su vez en **Sprints**.

Esta división del proyecto en tareas, permite desarrollo flexible y adaptativo, produciendo entregas parciales del producto en cada entrega de **Sprint**.

En particular en este proyecto se ha utilizado una mezcla de metodologías; **Scrum**, que prioriza el desarrollo en **Sprints**, con sus roles y eventos definidos y **Kanban**, que se centra en la visualización de los flujos de trabajo en un tablero con columnas, etiquetadas con las diferentes fases del desarrollo, en las cuales se van colocando tarjetas con las tareas que se están realizando y que se han asociado a los **Sprints**.

Para una mejor organización temporal, se han establecido una serie de hitos (**Milestones**), y éstos a su vez en **Sprints**, que agrupan un conjunto de **Sprints** reflejando las diferentes etapas del proyecto, como comprobamos en la planificación temporal siguiente:

## **A.2. Plan temporal**

Como se ha indicado anteriormente para una mejor gestión de tiempos se ha dividido el proyecto en diferentes hitos, **Sprints** y todo ello gestionado a través del **Kanban**:

## Kanban

El tablero **Kanban** ha sido una de las herramientas fundamentales para el desarrollo de esta aplicación. Gestionado por la plataforma **Zube** e integrado con **GitHub**, permite un control de las diferentes tareas. Para ello se dividen en varios estados, en el caso de este proyecto; **inbox** (entrada por defecto), **backlog** (lista inicial de trabajos a realizar por el equipo), **ready** (marca el inicio del **Sprint** para esa tarjeta), **in progress** (durante el desarrollo), **in review** (revisión del sprint) y finalmente **done**, cuando se cierra ese **Sprint**. Vemos un ejemplo del **Kanban** del proyecto.

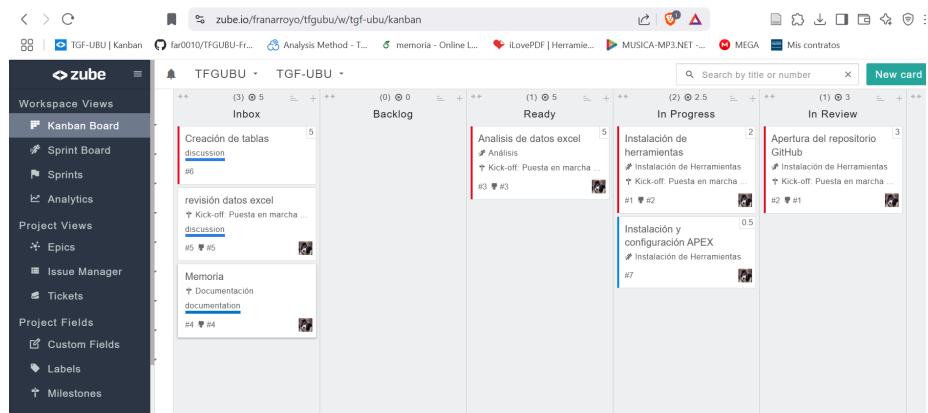


Figura A.1: Detalle del tablero kanban al inicio del proyecto

.6

## Milestone

- **Kick-off: Puesta en marcha del proyecto:** Completado el 16 de junio. Una vez reunido con el tutor y definidas las líneas generales del proyecto se procede a recopilar e instalar las herramientas necesarias para el desarrollo y la documentación. Este hito hubo que extender hasta el 23, por el upgrade a **APEX2402**
- **Prototipo:** Completado el 30 de junio. Aunque se han ido realizando diversas "entregas", se establece un prototipo que sirva de base para que el usuario "verifique si se está yendo en la dirección correcta en el desarrollo. Simplemente se muestran las primeras funcionalidades y los diversos menús que se podrán usar. Queda reflejado como el **1er Release**

- **MPV: Mínimo producto viable:** Completado el 14 de julio. Este **2º Release**, ya presentan prácticamente todas las funcionalidades de la aplicación, pudiendo observar los usuarios, si es ajustado a sus requerimientos.
- **Desarrollo completo:** Completado el 15 de agosto. Se ha finalizado la aplicación (**3er Release**) y despliega en Oracle cloud, siendo funcional a todos los efectos.
- **Documentación:** Se prepara, completa y revisa toda la documentación del proyecto; memoria, anexos, vídeos y el resto de documentos para presentar.

## Sprints

Para la ejecución del proyecto se han realizado los siguientes **Sprints**, algunos coincidentes en el tiempo dentro del mismo hito:

- **Kick-off: Puesta en marcha del proyecto hasta 16 de junio:**

**Análisis:** en este **Sprint** se realizaron las tareas de análisis de datos de la **hoja de cálculo excel** y su **Normalización**, **definición las entidades** y **sus relaciones** para finalmente crear las distintas **tablas** con sus campos, definiendo **PK** y **FK**

**Instalación de herramientas:** Se abrió el repositorio en **GitHub** para el proyecto, instalándose en modo local, **APEX 5.01** y **Oracle 9**, herramientas que se habían utilizado en la asignatura de *Sistemas de Gestión de Bases de Datos*. También se instaló **TestCafé** para la ejecución de test, con el complemento (Allure) así como el editor de LaTex, **TexStudio**.



Figura A.2: Milestone: Kick-Off

- Prototipo: hasta el 30/06:

**Creación pantalla de inicio:** Se crea la pantalla de **log-in** y la página de entrada principal, así como el primer módulo de Administración, que contendrá los datos de los **proyectos** de los **responsables**. Como los datos obrantes en el sistema son privados, es preciso la creación para las pruebas de **Datos MOCK**.

**Módulos y test:** En este **Sprint** se diseña un programa en Phyton para llenar las tablas de responsables y proyectos con datos de prueba y se llenan las tablas correspondientes para poder realizar los test. Se diseñan los primeros Test para los menús creados que son la base del prototipo. Se realiza la migración de **HTTP** a **HTTPS**, debiéndose obtener un certificado a través de **OpenSSL**

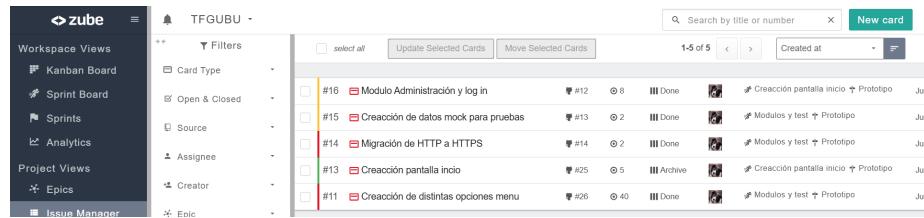


Figura A.3: Milestone: Prototipo

- MPV: hasta el 14/07:

**Creación Contrato.** Se crean la utilidad principal para efectuar un contrato a un solicitante. El proceso debe crear las nóminas correspondientes en la tabla **NOMINAS**.

**Creación de renovación y renuncias:** Se crean los módulos de renovación, que amplía la fecha de fin de contrato y añade las nóminas correspondientes y el de renuncia, que recorta la fecha de contrato y elimina la nómina desde esa fecha hasta el final del contrato anterior.

**Test de usabilidad:** Se testean estas nuevas funcionalidades, comprobando que se actualizan correctamente las fechas y las nóminas.

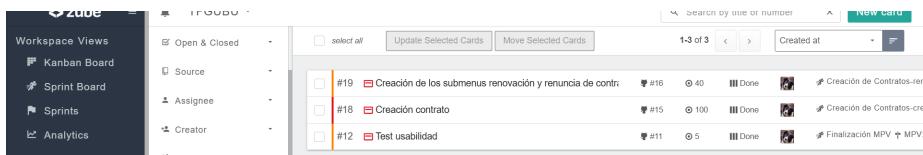


Figura A.4: Mileston: MPV

- Desarrollo Completo: hasta 15 de agosto:

**Creación de informes:** Se crean los informes pendientes, **Nóminas, Vencimientos y Nóminas periodo.**

**Creación de cuenta Oracle Cloud:** Se crea la cuenta de **OCI** y las instancias para la BD y **APEX**. Se efectúa el traspaso de la base de datos local a Cloud y se instala la aplicación MPV, para su prueba.

**Revisión de Test e informes:** Se crea e instala **APEX Office Print (AOP)**. Se realizan los test de estos últimos informes, para comprobar que se generan correctamente y se emite el informe en PDF a través de **AOP**.

**Creación de Oracle Cloud Infrastructure (OCI):** Se finaliza el desarrollo de la aplicación y se despliega en **OCI**. En este caso no hace falta certificado **SSL** ni para Cloud y **AOP**, ya que al ser el mismo proveedor confía en su entorno.

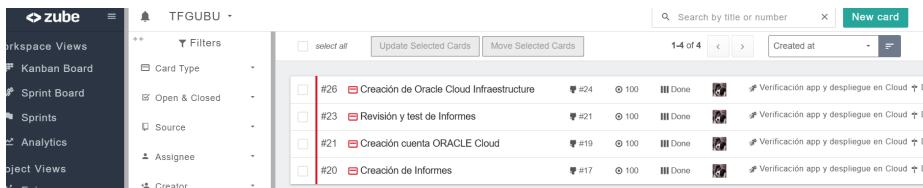


Figura A.5: Milestone: Desarrollo Completo

- Documentación: hasta el 5 de septiembre:

**Generación de documentación:** Se finaliza La memoria del proyecto, documento creciente que se va desarrollando a lo largo del proyecto, pero es en este punto final cuando se invierte la mayoría del tiempo en su finalización, plasmándose los **Objetivos de Desarrollo Sostenible (ODS)**. Para la realización de la memoria y los anexos ha sido imprescindible el aprendizaje de **LATEX**. Finalizado el proyecto se crea el manual de uso y se sube al **wiki** de GitHub. Para finalizar

la documentación se crean los vídeos de funcionamiento de la app **GeNomIn** y de presentación del **TFG**

Figura A.6: Milestone: Documentación

- Gráfica de Velocidad: en la siguiente gráfica podemos ver cómo se han ido realizando los diferentes **Sprints**, manteniendo una constancia relativa. Se puede observar, que la tarea de **verificación y despliegue** ha sido más laboriosa, ya que comprendía la finalización y trasvase de datos a Cloud.

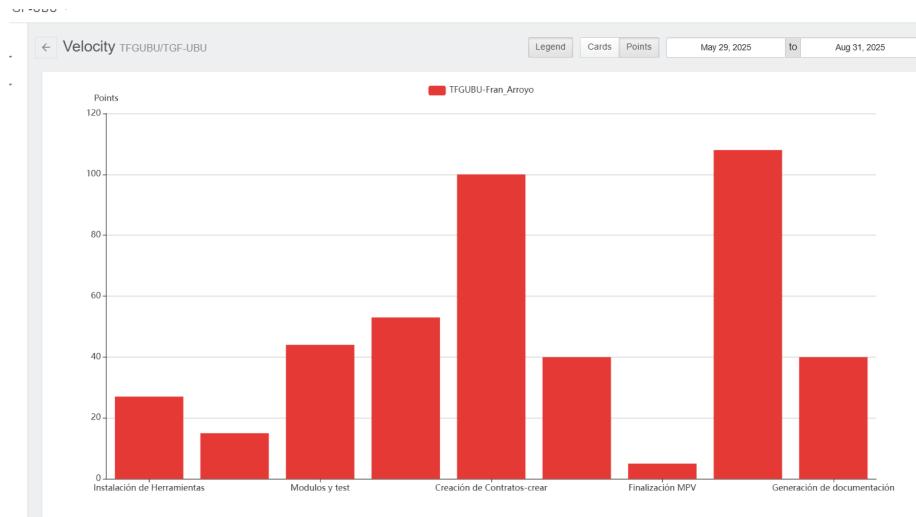


Figura A.7: Gráfica de Velocidad de los Sprints

Vemos también, en la imagen, como las tareas han sido sincronizadas con el repositorio de **GitHub**, incluyendo su **Sprint** y **Milestone** asignados.



Figura A.8: Tarea #2:Instalación de herramientas en GitHub

## A.3. Estudio de Viabilidad

### Viabilidad Técnica

Este proyecto ha sido realizado siguiendo la filosofía de software libre y coste cero, lo que garantiza su viabilidad técnica, ya que todas las herramientas utilizadas son open access, como vemos en la siguiente tabla:

Herramienta	Función	Licencia / Coste
Oracle 23ai	BD relacional	Gratis (Free Tier / Licencia UBU)
SQL Developer	Diseño BD	Gratis
Oracle APEX 24.02	Desarrollo web	Gratis (Free Tier / Licencia UBU)
AOP – Apex Office Print	Informes PDF personalizados	100 informes, luego 35 €/mes
TestCafe	Pruebas	Gratis
Oracle Cloud Free Tier	Despliegue en la nube	Gratis
TeXstudio	Redacción técnica	Gratis

Tabla A.1: Herramientas utilizadas en el desarrollo del proyecto

## Viabilidad Económica

Para el desarrollo del proyecto se estiman unas 250 horas a un precio estimado de 25 €/hora para un técnico cualificado, las licencias ya se ha indicado que son libres, salvo **APEX Office Print (AOP)** en producción. El alojamiento no es necesario en la nube, ya que se utilizaría en los servidores propios y las licencias existentes.

Concepto	Importe €
Horas de desarrollo (250 h x 25 €/h)	6.250 €
Hardware	0 (Entorno UBU)
Licencias Soft	0 (Salvo AOP en producción)
AOP – Apex Office Print	35 €/mes
Mantenimiento	0 (Entorno UBU)
<b>Total</b>	<b>6.670 €</b>

Tabla A.2: Coste económico GeNomIn

## Viabilidad Legal

- **Propiedad intelectual:** Este desarrollo ha sido realizado por el autor del **TFG**, sin restricciones.
- **Protección de datos:** Los datos son gestionados por la Universidad de Burgos, dentro de su marco legal, cumpliendo se con el **RGPD**.
- **licencias:** Todas las licencias, como se ha comentado son o libres o cubiertas por la propia Universidad, salvo **AOP**, si se supera el límite de informes.

## Viabilidad de Implantación

- **Entorno Local:** La Universidad dispone de las herramientas y capacidad para la instalación, con licencias de Oracle.
- **Escalabilidad:** como ya se ha demostrado al escalarlo al entorno de Oracle Cloud, no habría problema en el uso de servidores de la propia red.

- **Mantenimiento:** Las tecnologías usadas pueden ser mantenidas por el personal técnico interno.

## Conclusiones

: La aplicación web **GeNomIn**, es viable tanto técnica como legalmente, con un coste contenido y un despliegue seguro en los servidores de la Universidad, cumpliendo con el objetivo inicial de facilitar el trabajo al Servicio en cargado de la gestión de contratos con cargo a proyectos de Investigación.



## *Apéndice B*

---

# **Requisitos y Casos de Uso**

---

## **B.1. Introducción**

En este anexo B, se presentarán los requisitos de la aplicación, que definen su comportamiento, detallándolo a través de tablas y diagramas para facilitar su comprensión. Sin perder de vista el objetivo principal de esta aplicación **GeNomIn**, que es la de ofrecer una alternativa a al uso de una hoja de cálculo para la gestión de pagos en contratos vinculados a proyectos de investigación, vamos a ir desgranando cada una de las actuaciones para llegar a este objetivo.

Como podemos ver a la siguiente imagen siguiente, en el proceso de la realización de un contrato habría tres actores principales; **Investigador**, que es quien realiza la solicitud de contratación, **Solicitante**, que son las personas que presentan una solicitud para participar y los actores principales de la aplicación, los **Gestores** de **GeNomIn**, que son los que realizan todas las acciones en la aplicación.

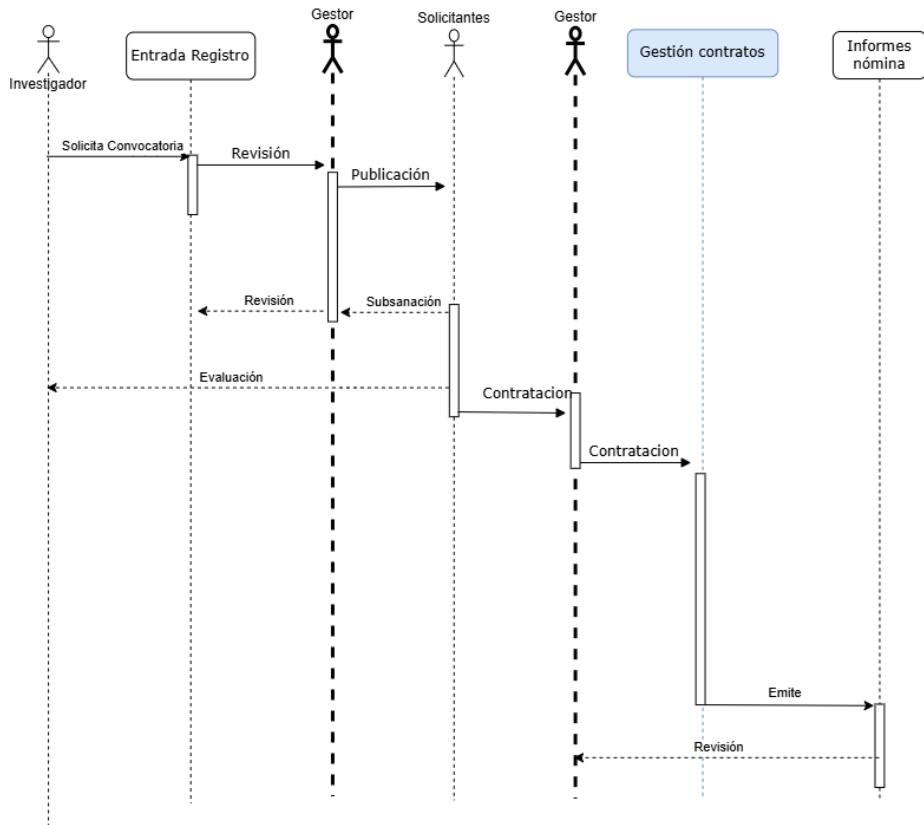


Figura B.1: Diagrama de secuencia del proceso General

Así, una vez realizada la solicitud y resuelta la convocatoria se realizan el contrato con el solicitante seleccionado y es aquí cuando se iniciaba el traslado de datos a la tabla de Excel, y que ahora, el **Gestor**, realizará a través de la aplicación y reflejamos en los requisitos, los casos de uso y test ejecutados siguientes.

## B.2. Requisitos Funcionales

### RF01:

El Gestor debe poder acceder al sistema, con su **usuario** y **contraseña**

<b>Identificador</b>	US-01
<b>Título</b>	Inicio de sesión con usuario "user01" y contraseña "user01"
<b>Requisito vinculado</b>	R.F.01
<b>Precondiciones</b>	El usuario "user01" ha sido registrado en el sistema.
<b>Postcondiciones</b>	El usuario accede a su cuenta de usuario.
<b>Proceso</b>	El usuario se loguea en el sistema.
<b>Datos de prueba</b>	Usuario: "user01" Contraseña: "user01"
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente.
<b>Estado</b>	Realizado: Test: t01_login.js

Tabla B.1: Caso de prueba US-01

**RF02:**

Un usuario sin estar dado de alta no puede acceder al sistema.

<b>Identificador</b>	US-02
<b>Título</b>	Inicio de sesión con usuario "no existe" y contraseña "ninguna"
<b>Requisito vinculado</b>	R.F.02
<b>Precondiciones</b>	El usuario "no existe" no ha sido registrado en el sistema.
<b>Postcondiciones</b>	El usuario no accede a su cuenta de usuario.
<b>Proceso</b>	El usuario intenta loguearse con usuario y contraseña erróneos
<b>Datos de prueba</b>	Usuario: "no existe" Contraseña: "ninguna"
<b>Resultados esperados</b>	El usuario no accede a la página de inicio de la aplicación correctamente y se presenta rechazado
<b>Estado</b>	Realizado: Test: t02_loginfail.js

Tabla B.2: Caso de prueba US-02

**RF03, RF04, RF05 y RF06:**

El Gestor accede a su usuario y contraseña y puede desplegar los distintos menús de la aplicación.

<b>Identificador</b>	US-03, US-04, US-05 y US-06
<b>Título</b>	El Gestor accede al sistema y despliega distintos menús
<b>Requisito vinculado</b>	RF03, RF04, RF06 y RF06
<b>Precondiciones</b>	El usuario "user01" ha sido registrado en el sistema.
<b>Postcondiciones</b>	El Usuario accede a su cuenta.
<b>Proceso</b>	El usuario se loguea y despliega Administración, Responsables, Proyectos, Solicitantes
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01"
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación y despliega correctamente los distintos menús
<b>Estado</b>	Realizado: Test: t03_menuAdmin.js, t04_logyresp.js, t05_logypro.js, t06_logysol.js

Tabla B.3: Caso de prueba US-03/06

**RF07:**

El Gestor accede a con su usuario y contraseña y puede crear un nuevo responsable.

<b>Identificador</b>	US-07
<b>Título</b>	El Gestor accede y crea uno nuevo responsable
<b>Requisito vinculado</b>	R.F.07
<b>Precondiciones</b>	El usuario "user01" ha sido registrado en el sistema.
<b>Postcondiciones</b>	Debe haber un responsable más
<b>Proceso</b>	El usuario se loguea y despliega Responsables pulsa en crea y añade el nuevo responsable
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", DNI: 00000005E, Pérez López, Juan
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Responsable y crea uno nuevo. Debe haber una fila más en la tabla de responsables.
<b>Estado</b>	Realizado: Test: t07_crearResp.js

Tabla B.4: Caso de prueba US-07

**RF08:**

El Gestor accede a con su usuario y contraseña y puede borrar un responsable.

<b>Identificador</b>	US-08
<b>Título</b>	El Gestor accede y borra el responsable creado
<b>Requisito vinculado</b>	R.F.08
<b>Precondiciones</b>	El usuario "user01" ha sido registrado en el sistema.
<b>Postcondiciones</b>	Debe haber un responsable menos
<b>Proceso</b>	El usuario se loguea y accede al menú de responsables, pulsa en el campo de búsqueda e introduce el NIF: 00000005E, pulsa en el lapisero y luego en el botón Delete.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", DNI: 00000005E
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Responsable y borra el responsable creado. Vuelve a buscar por DNI, debe obtener el mensaje Data not found.
<b>Estado</b>	Realizado: Test: t08_borrarResp.js

Tabla B.5: Caso de prueba US-08

**RF09:**

El Gestor accede con su usuario y puede crear una convocatoria.

<b>Identificador</b>	US-09
<b>Título</b>	El Gestor accede con su usuario y crea una Convocatoria
<b>Requisito vinculado</b>	R.F.09
<b>Precondiciones</b>	El usuario "user01" ha sido registrado en el sistema.
<b>Postcondiciones</b>	Debe haber una convocatoria nueva más
<b>Proceso</b>	El usuario se loguea accede a convocatorias y crea una nueva
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", Convocatoria: PRUEBA PARA BORRAR, Tit: Grado, abierta
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Convocatorias-Mantenimiento y crea la convocatoria. Debe haber una fila más en la tabla
<b>Estado</b>	Realizado: Test: t09_crearcConv.js

Tabla B.6: Caso de prueba US-09

**RF10:**

El Gestor accede a con su usuario y contraseña y puede borrar una convocatoria.

<b>Identificador</b>	US-10
<b>Título</b>	El Gestor accede y borra la convocatoria de prueba
<b>Requisito vinculado</b>	R.F.10
<b>Precondiciones</b>	El usuario "user01" ha sido registrado en el sistema y no hay ninguna convocatoria más abierta.
<b>Postcondiciones</b>	No Debe haber una convocatorias disponibles.
<b>Proceso</b>	El usuario se loguea y accede a convocatorias, pulsa en el icono del lápiz de la convocatoria a eliminar y pulsa en el botón Delete.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", Convocatoria: PRUEBA PARA BORRAR
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Convocatorias-Mantenimiento y borra la convocatoria. Sale el mensaje Data not found, ya que no quedan más filas.
<b>Estado</b>	Realizado: Test: t10_borrarConvSimple.js

Tabla B.7: Caso de prueba US-10

**RF11:**

No debe poderse asociar un solicitante a una convocatoria cerrada.

<b>Identificador</b>	US-11
<b>Título</b>	El Gestor accede e intenta crea un solicitante e intenta asociarlo a convocatoria cerrada
<b>Requisito vinculado</b>	R.F.11
<b>Precondiciones</b>	El usuario "user01" debe haber una convocatoria cerrada para poder probar.
<b>Postcondiciones</b>	No se asocia el solicitante a ninguna convocatoria.
<b>Proceso</b>	El usuario se loguea he intenta asociar un solicitante a una convocatoria cerrada
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01",solicitante: "00001FAKE", Pérez López, Juan, Convocatoria: Prueba convocatoria 3 cerrada
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Convocatorias-Solicitantes crea el solicitante, pero sin asignar convocatoria
<b>Estado</b>	Realizado: Test: t11_crearSol_Con_Close.js

Tabla B.8: Caso de prueba US-11

**RF12:**

No puede crear y asociarse un solicitante a una convocatoria sin tener la titulación adecuada.

<b>Identificador</b>	US-12
<b>Título</b>	Inicio de sesión con usuario "user01" contraseña "user01" crear solicitante e intentar asociar a convocatoria sin tener la titulación cerrada
<b>Requisito vinculado</b>	R.F.12
<b>Precondiciones</b>	El usuario "user01" debe haber una convocatoria para poder probar.
<b>Postcondiciones</b>	No se asocia el solicitante a ninguna convocatoria.
<b>Proceso</b>	Loguea el usuario y accede e intenta asignar una convocatoria con titulación superior a la que posee
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01",Solicitante: "00001FAKE", Pérez López, Juan -Diplomado, Convocatoria: prueba de convocatoria 1- grado
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Convocatorias-Solicitantes edita el solicitante, pero sin asignar convocatoria por la titulación incorrecta.
<b>Estado</b>	Realizado: Test: t12_modSol_No_Tit.js

Tabla B.9: Caso de prueba US-12

**RF13:**

Se debe poder asociar un solicitante con titulación adecuada a una solicitud.

<b>Identificador</b>	US-13
<b>Título</b>	Inicio de sesión con usuario user01z contraseña user01z modificar solicitante e intentar asociar a convocatoria con tit. correcta
<b>Requisito vinculado</b>	R.F.13
<b>Precondiciones</b>	El usuario "user01" debe haber una convocatoria para poder probar y solicitante con titulación adecuada.
<b>Postcondiciones</b>	Se le asocia al solicitante la convocatoria.
<b>Proceso</b>	Una vez logueado el usuario selecciona Solicitantes y pulsa en el lápiz del usuario a modificar la convocatoria adecuada y lo asigna
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01",Solicitante: "00001FAKE", Pérez López, Juan Convocatoria: CONV017
<b>Resultados esperados</b>	Al solicitante Juan, se le asigna la convocatoria: CONVO017
<b>Estado</b>	Realizado: Test: t13_modSol_OK_Tit.js

Tabla B.10: Caso de prueba US-13

**RF14:**

Se debe poderse borrar un solicitante existente.

<b>Identificador</b>	US-14
<b>Título</b>	Borra el solicitante creado
<b>Requisito vinculado</b>	R.F.14
<b>Precondiciones</b>	Debe existir un solicitante para borrar.
<b>Postcondiciones</b>	Se elimina el solicitante.
<b>Proceso</b>	Accedemos a la página de inicio de sesión log, se accede a solicitantes y se pulsa en botón editar del solicitante, luego se pulsa en borrar y se confirma.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01",Solicitante: "00001FAKE", Pérez López
<b>Resultados esperados</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Convocatorias-Solicitantes edita el solicitante, y elimina el solicitante antes creado, mostrando el mensaje Action Processed.
<b>Estado</b>	Realizado: Test: t14_delSolicitante.js

Tabla B.11: Caso de prueba US-14

**RF15, RF16 y RF17:**

Se debe poderse crear un contrato para un solicitante asignado correctamente a una convocatoria, siempre y cuando las fechas estén dentro del periodo del proyecto. Deben generarse las nóminas correspondientes

<b>Identificador</b>	US-15, US-16 y US-17
<b>Título</b>	Crear un contrato a un solicitante
<b>Requisito vinculado</b>	RF.15
<b>Precondiciones</b>	Debe existir solicitante asignado a una convocatoria y que el contrato se encuentre entre las fechas del proyecto.
<b>Postcondiciones</b>	Se crea el contrato y se generan las nóminas correspondientes.
<b>Proceso</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Contratos-Crear, ingresa los datos solicitados y se crea el contrato.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01" Contratado: Juncal Álvarez Leal Fechas: 01-NOV-2025 a 31-OCT-2027, Ret total: 49000, mes: 1500, ss: 500, indemnización: 1000, reserva: 49000, observaciones: PRUEBA TEST
<b>Resultados esperados</b>	Debe haberse creado el contrato en la tabla correspondiente y 25 nóminas para ese DNI, en la tabla nómina
<b>Estado</b>	Realizado: Test: t15_crearcContrato.js, t16_valFechasContrato.js y t17_compNominas.js

Tabla B.12: Caso de prueba US-15, US-16 y US-17

**RF18, RF19 y RF20:**

Se debe poder renovar un contrato, siempre y cuando la fecha fin no exceda del fin del proyecto. Además deben añadirse las nóminas correspondientes.

<b>Identificador</b>	US-18, US-19 y US-20
<b>Título</b>	Renovación de un contrato para el solicitante, se validan las fechas de la nueva renovación y se actualiza la fecha fin del contrato y las nuevas cantidades de nómina, seg. Social e indemnización
<b>Requisito vinculado</b>	RF.18, RF.19, RF20
<b>Precondiciones</b>	Debe existir solicitante con contrato que no haya llegado a la fecha fin del proyecto.
<b>Postcondiciones</b>	Se efectúa la renovación del contrato modificando su fecha fin y se generan las nóminas nuevas nóminas correspondientes.
<b>Proceso</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Contratos selecciona el contrato a renovar e ingresa los nuevos datos solicitados.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01" Contrato: CONT123, Fecha fin nueva: 30-sep-2026
<b>Resultados esperados</b>	El contrato indicado tiene que haberse renovado hasta la fecha indicada con las nuevas cantidades, además de incluirse en Observaciones: RENOVADO. Se deben haber generado 16 nuevas nóminas adicionales.
<b>Estado</b>	Realizado: Test: t18_renovarContratoFechas copy.js, t19_renovarContrato.js, t20_compRenNominas.js

Tabla B.13: Caso de prueba US-18, US-19 y US-20

**RF21, RF22 y RF23:**

Se debe poder renunciar a un contrato, siempre y cuando la fecha no sea posterior al fin del mismo y eliminando las nóminas sobrantes.

<b>Identificador</b>	US-21, US-22 y US-23
<b>Título</b>	Renuncia del contrato CONT141 con fecha 31-12-2025, eliminando nóminas y fecha fin.
<b>Requisito vinculado</b>	RF.21, RF.22, RF23
<b>Precondiciones</b>	Debe existir solicitante con contrato en vigor.
<b>Postcondiciones</b>	Se efectúa la renuncia del contrato modificando su fecha fin y se eliminan las nóminas correspondientes.
<b>Proceso</b>	El usuario accede a la página de inicio de la aplicación correctamente y despliega Contratos selecciona el contrato a renunciar e indica la nueva fecha fin correcta.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", DNI =12345678M, fecha 21-12-2025
<b>Resultados esperados</b>	El contrato indicado tiene que haberse reducido hasta la fecha indicada, además de incluirse en Observaciones: RENUNCIA. Se deben haber eliminado las nóminas sobrantes.
<b>Estado</b>	Realizado: Test: t21_renunciaContrato.js, t22_renunciaContratoFechas.js, t23_compRenunnNominas.js

Tabla B.14: Caso de prueba US-21, US-22 y US-23

**RF24:**

Se debe poder generar un informe de la nómina de un mes determinado y luego imprimirla en PDF.

<b>Identificador</b>	US-24
<b>Título</b>	Comprobación si se genera informe de la nómina de un mes determinado y se puede imprimir dicho mes pdf.
<b>Requisito vinculado</b>	RF.24
<b>Precondiciones</b>	Deben existir contratos en vigor para el mes/años solicitados.
<b>Postcondiciones</b>	Se genera el informe en PDF
<b>Proceso</b>	El usuario accede a la sección de informes Nómina-mes, selecciona un mes y un año y pulsa “consultar”. Una vez listado podrá escoger filtrar por orgánica y volviendo a pulsar “consultar” Pulsando Generar PDF descarga el informe personalizado.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", Mes: "Diciembre" Año "2025"
<b>Resultados esperados</b>	Se lista los datos de nómina correspondientes al mes y año. Se puede realizar filtro por orgánica descargar un pdf personalizado.
<b>Estado</b>	Realizado: Test: t24_infNominaMesyPDF.js

Tabla B.15: Caso de prueba US-24

**RF25:**

Se debe poder comprobar los vencimientos de contratos entre fechas.

<b>Identificador</b>	US-25
<b>Título</b>	Comprobación de vencimientos de contratos entre fechas.
<b>Requisito vinculado</b>	RF.25
<b>Precondiciones</b>	Deben existir contratos en vigor entre las fechas solicitadas.
<b>Postcondiciones</b>	Se visualiza el informe
<b>Proceso</b>	El usuario accede a la sección de informes Vencimientos y selecciona intervalo de fechas y pulsa botón Consulta
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01", 01-01-2020 a 31-12-2025
<b>Resultados esperados</b>	Se listan los datos de vencimientos de contrato en el periodo solicitado.
<b>Estado</b>	Realizado: Test: t25_infVencimientos.js

Tabla B.16: Caso de prueba US-25

**RF26:**

Se debe poder obtener un listado con todos los datos generales de contratos y aplicar filtros.

<b>Identificador</b>	US-26
<b>Título</b>	Listado de contratos.
<b>Requisito vinculado</b>	RF.26
<b>Precondiciones</b>	Deben existir contratos en vigor.
<b>Postcondiciones</b>	Se visualiza el informe
<b>Proceso</b>	El usuario accede a la sección de informes Listado de contratos.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01"
<b>Resultados esperados</b>	Se listan los datos de vencimientos de contrato en el periodo solicitado.
<b>Estado</b>	Realizado: Test: t26_listaContratosFull.js

Tabla B.17: Caso de prueba US-26

**RF27:**

Se debe poder obtener un listado con las nóminas de un contratado en un periodo indicado. Debe ofrecer el total de remuneración en ese periodo.

<b>Identificador</b>	US-27
<b>Título</b>	Listado de nóminas de contratado por periodo
<b>Requisito vinculado</b>	RF.27
<b>Precondiciones</b>	Deben existir contratos en vigor para un solicitante en el periodo.
<b>Postcondiciones</b>	Se visualiza el informe
<b>Proceso</b>	El usuario accede a la sección de informes Nóminas-periodo. Selecciona un contratado de la selección e indica un intervalo de fechas y pulsa buscar.
<b>Datos de prueba</b>	usuario: "user01" Contraseña: "user01" Contratado: Juncal Álvarez Pérez, periodo: 01-01-2025 a 31-12-2025
<b>Resultados esperados</b>	Se ofrece un listado de las nóminas de ese contratado en el periodo solicitado junto con su total para comprobación = 4600.25
<b>Estado</b>	Realizado: Test: t27_infNomPeriodo.js

Tabla B.18: Caso de prueba US-27

En la siguiente tabla vemos a través del informe de Allure, los test pasados (informe completo [aquí](#)) El código de los test en JS pueden verse: [aquí](#)

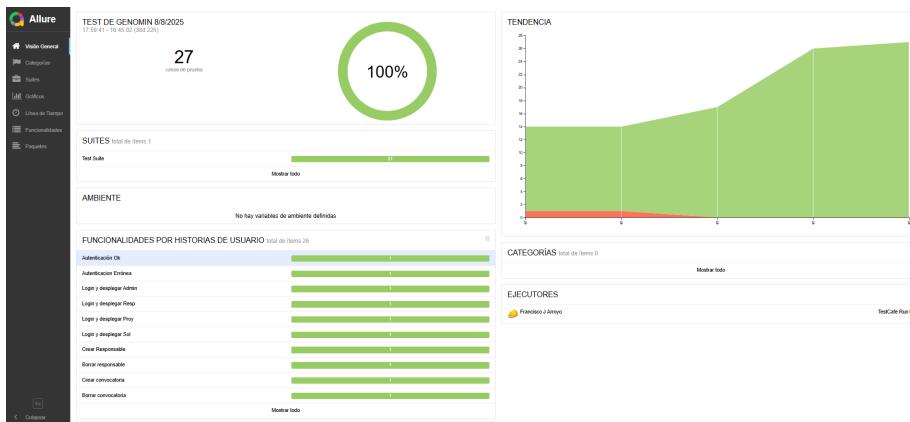


Figura B.2: Informe Allure de Test



## *Apéndice C*

---

# Diseño GeNomIn

---

## C.1. Introducción

En este Anexo C, voy a explicar como se ha realizado la aplicación **GeNomIn** a través de la plataforma *Application Express (APEX)* 2402. Se datallará como se han implementado las diferentes páginas para obtener las funcionalidades y requisitos exigidos por los usuarios.

Como ya se indicó en el **apartado 4.2** de la memoria, una vez instalado **APEX** y la base de datos *ORACLE 23ai (ODB)*, es preciso crear los usuarios que utilizarán la aplicación, en este caso a parte del **Administrador**, se crearon un **desarrollador** y un **usuario final**.

## C.2. App Builder

La herramienta para construcción de aplicaciones de apex [1], es un acrshortIDE con poco código, a través de páginas y componentes compartidos (**shared components**). Inicialmente podemos crear nuestra apliación a través de varias fuentes; con ficheros (XML, CSV XLSX, JSON, SQL), copiando aplicaciones existentes o directamente con el asistente, que nos creará una página de lógin y nos irá permitiendo añadir páginas y componentes para personalizar nuestro diseño.

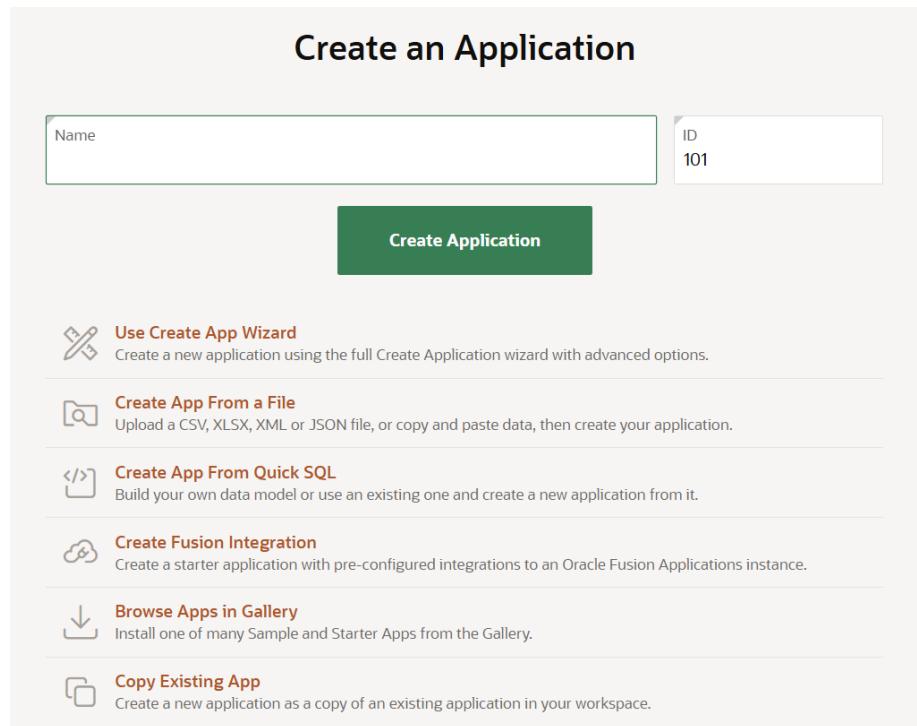


Figura C.1: Creación de app APEX

En los siguientes apartados se irán detallando los aspectos fundamentales para el desarrollo de las páginas necesarias e importantes en este proyecto y el código asociado, ya sea en **SQL**, **PL/SQL** o **JS**. Todo este código también está disponible en el repositorio [Código de GeNomIn](#)

### C.3. Página Proyectos

En esta página se detallan todos los proyectos que existen en la base de datos a través de una consulta simple **SQL**. El gestor de **acciones** de **APEX**, permite ejecutar búsquedas y diversos filtros, además de añadir, eliminar registros y actualizar datos.

Código C.1: Informe de todos los Proyectos

```
SELECT
    p.ORGANICA,
    p.TITULO,
    r.NOMBRE || ' ' || r.ape1 || ' ' || r.ape2 AS REPONSABLE,
    p.fecha_ini,
```

```

p.fecha_fin
FROM proyecto p
JOIN responsable r ON p.responsable = r.dni;

```

## C.4. Página Responsables

Al igual que la página anterior usa una consulta **SQL**, para mostrar todos los responsables, tengan o no, proyectos abiertos. También permite realizar **Acciones**.

Código C.2: Informe de todos los Responsables

```

SELECT
R."DNI",
R."APE1",
R."APE2",
R."NOMBRE",
D."NOMBRE" AS "DEPARTAMENTO"
FROM RESPONSABLE R
JOIN DEPTOS D
ON R."DEPTO" = D."REF";

```

## C.5. Página Mantenimiento Convocatorias

En esta página se hace una consulta **SQL** que devuelve el detalle de todas las convocatorias y su responsable asociado, incluyendo **Acciones**.

Código C.3: Informe de todas las Convocatorias

```

SELECT C.REFERENCIA,
C.TITULO,
C.TIT_REQUERIDA,
CASE C.ABIERTO WHEN 1 THEN 'S\'{i}' ELSE 'No' END AS ABIERTO,
R.APE1 || ' ' || R.APE2 || ', ' || R.NOMBRE AS INVESTIGADOR,
C.REF_PROY,
C.NUM_PLAZAS
FROM CONVOCATORIA C
LEFT JOIN RESPONSABLE R ON C.REF_INV = R.DNI

```

### Crear Convocatoria

Para crear una nueva convocatoria ha sido preciso el uso de; primero, una consulta **SQL**, que no muestra todos los investigadores:

Código C.4: Presenta de todos los Investigadores

```
SELECT APE1 || ' ' || APE2 || ' ' || NOMBRE AS DISPLAY_VALUE
FROM RESPONSABLE
ORDER BY APE1, APE2
```

segundo, una consulta que una vez escogido el investigador, muestre los proyectos abiertos:

Código C.5: Presenta informe proyectos abiertos para el investigador seleccionado

```
SELECT P.TITULO, P.ORGANICA
FROM PROYECTO P
WHERE P.RESPONSABLE = :P9_REF_INV
AND (P.FECHA_FIN IS NULL OR P.FECHA_FIN >= TRUNC(SYSDATE))
ORDER BY P.TITULO;
```

Y dos acciones dinámicas en JS, en la que se controla: primera, si el investigador tiene proyectos abiertos desde la fecha actual, si no, muestra mensaje de aviso:

Código C.6: Control y aviso de proyecto abiertos de investigador seleccionado

```
apex.message.clearErrors();

// Esperamos brevemente a que el combo termine de cargarse
setTimeout(function() {
    const opciones = $("#P9_REF_PROY_option");
    const seleccionado = $v("P9_REF_PROY");

    // Si solo hay una opcion la nula o ninguna
    if (opciones.length <= 1 || !opciones[1].value) {
        apex.message.alert("AVISO: El investigador");
    }
}, 300);
```

y la segunda, nos muestra el tiempo restante del proyecto para tener en cuenta en la duración de contratos.

Código C.7: Informa de la duración restante del proyecto elegido

```
apex.message.clearErrors();

var duracion = $v("P9_DURACION");
if (duracion && duracion.trim() != "") {
```

```

        apex.message.alert("AVISO: La duracion del proyecto es:
}

```

## C.6. Página de Solicitantes

Al igual que las anteriores, muestra una búsqueda **SQL**, simple, con todos los solicitantes y si tienen o no una convocatoria asignada:

Código C.8: Presenta informe de solicitantes

```

select "DNI_SOL",
       "APE1_SOL",
       "APE2_SOL",
       "NOM_SOL",
       "TIT_SOL",
       "REF_CON"
  from "#OWNER#"."SOLICITANTE"

```

### Crear solicitante

Para la creación de un nuevos solicitantes, hay que tener en cuenta varios aspectos, como son; si quedan plazas disponibles en la convocatoria, esto se ha resuelto con una búsqueda **SQL** que cuenta los solicitantes de las convocatorias y las muestra si son mayores de 0:

Código C.9: Presenta informe de solicitantes

```

SELECT C.TITULO, C.REFERENCIA
  FROM CONVOCATORIA C
LEFT JOIN (
    SELECT REF_CON, COUNT(*) AS NUM_SOLICITANTES
      FROM SOLICITANTE
     GROUP BY REF_CON
) S ON C.REFERENCIA = S.REF_CON
 WHERE NVL(S.NUM_SOLICITANTES, 0) < C.NUM_PLAZAS
ORDER BY C.TITULO, C.REFERENCIA;

```

Luego es preciso verificar si la convocatoria está abierta y que envíe un mensaje de aviso en caso de estar cerrada. Esto se ha resuelto con una primera **acción dinámica**, con consulta **SQL** que guarda el campo abierto en un **textbox** oculto. Para ello, se utilizan las opciones de la página **APEX**: **items to submit** y **Affected Elements**, que nos permiten enviar y guardar los datos.

Código C.10: Comprueba campo abierto de convocatoria

```
SELECT ABIERTO
FROM CONVOCATORIA
WHERE REFERENCIA = :P7_REF_CON
```

Una vez guardado el resultado en el campo oculto **P7\_CON\_OPEN**, comprobamos con la siguiente parte de esta primera acción, a través de **JS**:

Código C.11: Control y aviso de convocatoria cerrada

```
apex.message.clearErrors();
```

```
const abierto = $v('P7_CON_OPEN');
```

```
if (abierto == '0') {
    const mensaje = 'Esta convocatoria esta cerrada';
    apex.message.alert("AVISO:" +mensaje);
    $s('P7_REF_CON', '');
} // Limpia la convocatoria par
```

El último paso es para comprobar si el solicitante tiene la titulación que requiere la convocatoria, procediendo de forma similar, con otra **acción dinámica**. Primero una consulta sencilla **SQL** que guarda la titulación requerida en un campo oculto, **P7\_TIT-REQ**.

Código C.12: Comprueba campo de titulación requerida

```
SELECT TIT_REQERIDA
FROM CONVOCATORIA
WHERE REFERENCIA = :P7_REF_CON
```

y luego mostramos un aviso con **JS**, en el caso de que la titulación del solicitante no sea igual o superior a la requerida en la convocatoria.

Código C.13: Control y aviso de titulación insuficiente

```
apex.message.clearErrors();
if ($v('P7_CON_OPEN') == '1'){
    // compruebo que la convocatoria esta abierta
    const valReq = $v('P7_TIT-REQ');
    const valSol = $v('P7_TIT_SOL');

    if (valReq !== '' && valSol !== '') {
        const tit_requerido = parseInt(valReq, 10);
        const tit_usuario = parseInt(valSol, 10);
```

```

        if (tit_usuario < tit_requerido) {
            // Buscar el texto asociado al valor requerido
            const select = document.getElementById('P7_REF_CON');
            const textoRequerido = Array.from(select.options);
            const mensaje = 'Nivel\u00e1cademico\u00e1insuficiente';
            apex.message.alert("AVISO:" +mensaje);
            $s('P7_REF_CON', '');
        }
    }
}

```

## C.7. Página de Contratos

Este primer **informe interactivo**, no requiere de consultas adicionales ya que **APEX**, realiza esta función son solo indicarle la tabla que queremos mostrar, en este caso CONTRATOS.

### Página Nuevo Contrato

Para la realización de un nuevo contrato tendremos que realizar diversas comprobaciones. Inicialmente ofrecemos las convocatorias disponibles, **APEX** también nos ofrece esta opción sin necesidad de programación adicional a través de **List of Values: CONVOCATORIA.TITULO**, en (shared componentes). En una primera consulta **SQL**, obtendremos los solicitantes que participan en la convocatoria elegida anteriormente:

Código C.14: Comprueba campo de titulación requerida

```

SELECT NOM_SOL || ' ' || APE1_SOL || ' ' || APE2_SOL AS DISPLAY_VALUE
      , DNI_SOL AS RETURN_VALUE
   FROM SOLICITANTE S
 WHERE REF_CON = :P12_CONVOCATORIA
   AND NOT EXISTS (
   SELECT 1
     FROM CONTRATOS C
    WHERE C.CONTRATADO = S.DNI_SOL
      AND C.F_FIN >= SYSDATE -- compruebo si esta activo
   )
 ORDER BY APE1_SOL;

```

Y ahora necesitamos tres **acciones dinámicas**, una que va a cargar las fechas de inicio y fin del proyecto, en campos ocultos, P12\_F\_INI\_OCULTO y P12\_F\_FIN\_OCULTO a través de dos consultas **SQL**:

Código C.15: Carga fecha inicio proyecto

```
SELECT TO_CHAR(P.FECHA_INI, 'YYYY-MM-DD') AS FECHA_INI_PROYE
FROM CONVOCATORIA C
JOIN PROYECTO P ON P.ORGANICA = C.REF_PROY
WHERE C.REFERENCIA = :P12_CONVOCATORIA
```

Código C.16: Carga fecha fin proyecto

```
SELECT TO_CHAR(P.FECHA_FIN, 'YYYY-MM-DD') AS FECHA_FIN_PROYE
FROM CONVOCATORIA C
JOIN PROYECTO P ON P.ORGANICA = C.REF_PROY
WHERE C.REFERENCIA = :P12_CONVOCATORIA
```

y **JS**, para forzar de nuevo la comprobación en caso de error:

Código C.17: Fuerza nueva petición de fechas si son erróneas

```
apex.event.trigger('#P12_F_INI', 'change');
apex.event.trigger('#P12_F_FIN', 'change');
```

Una vez cargadas las fechas de inicio y fin de los proyectos es preciso cotejarlas con las introducidas por el usuario, para comprobar si están dentro del margen del proyecto a través de dos **acciones dinámicas** en **JS**:

Código C.18: Controla que la fecha inicio sea posterior al inicio del proyecto

```
const fechaContratoStr = $v('P12_F_INI');
const fechaProyectoStr = $v('P12_F_INI_OCULTO');
const fechaContratoIni = new Date(fechaContratoStr);
const fechaProyectoIni = new Date(fechaProyectoStr);
console.log('INI_OCULTO:', $v('P12_F_INI_OCULTO'));
console.log('FIN_OCULTO:', $v('P12_F_FIN_OCULTO'));
```

```
if (!isNaN(fechaContratoIni) && !isNaN(fechaProyectoIni)) {
    if (fechaContratoIni < fechaProyectoIni) {
        const mensaje = 'La fecha de inicio debe ser posterior a la fecha de fin';
        fechaProyectoIni.toLocaleDateString('es-ES');
        apex.message.alert("AVISO:" + mensaje);
        $s('P12_F_INI', '');
        $('#P12_F_INI').trigger('change');
```

```

        }
    }
```

Y ahora comprobamos si la fecha de fin de contrato, es posterior a la de inicio y si sobrepasa al proyecto:

Código C.19: Controla que la fecha fin sea posterior a inicio y anterior al fin del proyecto

```

const fechaContratoFinStr = $v('P12_F_FIN');
const fechaContratoIniStr = $v('P12_F_INI');
const fechaProyectoFinStr = $v('P12_F_FIN_OCULTO');
const fechaContratoIni = new Date(fechaContratoIniStr);
const fechaContratoFin = new Date(fechaContratoFinStr);
const fechaProyectoFin = new Date(fechaProyectoFinStr);

if (!isNaN(fechaContratoFin) && !isNaN(fechaContratoIni)) {
    if (fechaContratoFin < fechaContratoIni) {
        const mensaje1 = 'La fecha de fin debe ser igual o posterior a la de inicio';
        apex.message.alert("AVISO:" + mensaje1);
        $s('P12_F_FIN', $v('P12_F_INI'));
        $('#P12_F_FIN').trigger('change');
    }
}
if (!isNaN(fechaContratoFin) && !isNaN(fechaProyectoFin)) {
    if (fechaContratoFin > fechaProyectoFin) {
        const mensaje2 = 'La fecha de fin debe ser igual o anterior a la de fin del proyecto';
        apex.message.alert("AVISO:" + mensaje2);
        $s('P12_F_FIN', '');
        $('#P12_F_FIN').trigger('change');
    }
}
```

Una vez realizadas las comprobaciones, es preciso el paso más importante generar las nóminas para el contrato en la tabla nómina. Para ello se requiere un **proceso** en **PL/SQL**:

Código C.20: Proceso que genera las nóminas relativas a un nuevo contrato

```

DECLARE
    v_fecha_ini DATE := :P12_F_INI;
    v_fecha_fin DATE := :P12_F_FIN;
    v_total_meses      PLS_INTEGER;
    v_meses_generados PLS_INTEGER := 0;
```

```

v_mes_actual           DATE;
v_ret_mes NUMBER := :P12_RET_MES;
v_ss_mes NUMBER := :P12_SS_MES;
v_indemnizacion NUMBER := :P12_INDEMNIZACION;

BEGIN
EXECUTE IMMEDIATE 'ALTER SESSION SET NLS_NUMERIC_CHARACTERS
                     .';

v_total_meses := MONTHS_BETWEEN(TRUNC(v_fecha_fin, 'MM'), TRUNC(v_fecha_ini, 'MM'));

WHILE v_meses_generados < v_total_meses LOOP
  v_mes_actual := ADD_MONTHS(TRUNC(v_fecha_ini, 'MM'), v_meses_generados);

  INSERT INTO NOMINA (
    dni_nom, mes, anno, mensualidad, seg_soc, observaciones
  ) VALUES (
    :P12_CONTRATADO,
    TO_NUMBER(TO_CHAR(v_mes_actual, 'MM')) ,
    TO_NUMBER(TO_CHAR(v_mes_actual, 'YYYY')) ,
    v_ret_mes,
    CASE
      WHEN v_meses_generados = 0 THEN 0
      ELSE v_ss_mes
    END,
    NULL
  );

  v_meses_generados := v_meses_generados + 1;
END LOOP;

-- Mes extra con indemnizacion
v_mes_actual := ADD_MONTHS(TRUNC(v_fecha_ini, 'MM'), v_total_meses);

  INSERT INTO NOMINA (
    dni_nom, mes, anno, mensualidad, seg_soc, observaciones
  ) VALUES (
    :P12_CONTRATADO,
    TO_NUMBER(TO_CHAR(v_mes_actual, 'MM')) ,
    TO_NUMBER(TO_CHAR(v_mes_actual, 'YYYY')) ,
    v_indemnizacion ,
    CASE
      WHEN v_meses_generados = 0 THEN 0
      ELSE v_ss_mes
    END
  );

```

```

    v_ss_mes ,
    'indemnizacion',
);

EXCEPTION
WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR(-20099, 'Error_interno:' || SQLERRM);
END;

```

## Página Renovación

La renovación de un contrato requiere de varios controles. Inicialmente tendremos que seleccionar los contratos con fecha fin posterior a la actual, a través de una consulta **SQL**, en el campo P13\_SEL\_CONT

Código C.21: Selección de contratos abiertos

```

SELECT
    C.REF_CONTRATO || ' - ' || S.NOM_SOL || ',' || S.APE1_SOL || ',' ||
    C.REF_CONTRATO AS RETURN_VALUE
FROM CONTRATOS C
JOIN SOLICITANTE S ON C.CONTRATADO = S.DNI_SOL
WHERE C.F_FIN >= TRUNC(SYSDATE)
ORDER BY C.REF_CONTRATO

```

Seguidamente a través de una **acción dinámica** mostraremos los campos de ese contrato seleccionado anteriormente con otra consulta **SQL**:

Código C.22: Obtención de datos del contrato a renovar

```

SELECT
    C.REF_CONTRATO,
    S.NOM_SOL || ',' || S.APE1_SOL || ',' || S.APE2_SOL AS NOMBRE_COMPI,
    TO_CHAR(C.F_INI, 'DD-MM-YYYY') AS F_INI,
    TO_CHAR(C.F_FIN, 'DD-MM-YYYY') AS F_FIN,
    TO_CHAR(C.RET_MES, '999G999G990D00') AS RET_MES,
    TO_CHAR(C.SS_MES, '999G999G990D00') AS SS_MES,
    TO_CHAR(C.INDEMNIZACION, '999G999G990D00') AS INDEMNIZACION
FROM CONTRATOS C
JOIN SOLICITANTE S ON C.CONTRATADO = S.DNI_SOL
WHERE C.REF_CONTRATO = :P13_SEL_CONT

```

Y posteriormente guardaremos dentro de esta misma acción dinámica la fecha fin en un campo oculto P13\_NEW\_FECHA, para su posterior comparación.

Como en anteriores ocasiones esto es posible por la propiedad de **APEX**, Affected elements.

Código C.23: Obtención de fecha fin del contrato a renovar

```
SELECT P.FECHA_FIN
FROM CONTRATOS C
JOIN CONVOCATORIA V ON C.CONVOCATORIA = V.REFERENCIA
JOIN PROYECTO P ON V.REF_PROY = P.ORGANICA
WHERE C.REF_CONTRATO = :P13_SEL_CONT
```

Una vez obtenidos los datos del contrato en vigor, se introduce la fecha de la renovación, que tiene que comprobarse a través de otra **acción dinámica** asociada a un botón **verifica-fecha**, que cambiará a color verde si es correcta y mostrará error en caso contrario. Esta acción dinámica es verificada por dos códigos **JS**: Este primer código es bastante más complejo, ya que ha requerido la conversión de fechas.

Código C.24: Verifica la fecha de renovación se posterior a la actual y anterior a fin proyecto

```
// parsear fechas en formato 'DD-MM-YYYY' o 'DD-MON-YYYY'
function parseFechaFlexible(fechaStr) {
    if (!fechaStr) {
        console.warn('Fecha_vacia_o_null:', fechaStr);
        return null;
    }

    const partes = fechaStr.trim().split(' ');
    if (partes.length !== 3) {
        console.error('Formato_inesperado:', fechaStr);
        return null;
    }

    const dia = parseInt(partes[0], 10);
    const año = parseInt(partes[2], 10);

    // Detectar si el mes es numero o texto
    const mesTexto = partes[1].toUpperCase();
    const meses = {
        'ENE': 0, 'FEB': 1, 'MAR': 2, 'ABR': 3,
        'MAY': 4, 'JUN': 5, 'JUL': 6, 'AGO': 7,
        'SEP': 8, 'OCT': 9, 'NOV': 10, 'DIC': 11
    };
}
```

```

let mes;
if (!isNaN(parseInt(mesTexto, 10))) {
    mes = parseInt(mesTexto, 10) - 1; // formato numero
} else {
    mes = meses[mesTexto]; // formato texto
}

if (mes === undefined || isNaN(dia) || isNaN(año)) {
    console.error('Problema al interpretar partes de la fecha');
    return null;
}

return new Date(año, mes, dia);
}

// Leer valores
const fechaInicioStr          = $v('P13_F_INI');
const fechaContratoFinStr      = $v('P13_F_FIN');
const nuevaFechaFinStr         = $v('P13_NEW_F_FIN');
const fechaProyectoFinStr     = $v('P13_NEW_FECHA');

// Parsear las fechas
const fechaInicio              = parseFechaFlexible(fechaInicioStr);
const fechaContratoFin         = parseFechaFlexible(fechaContratoFinStr);
const nuevaFechaFin             = parseFechaFlexible(nuevaFechaFinStr);
const fechaProyectoFin         = parseFechaFlexible(fechaProyectoFinStr);

// semaforo
let estadoVerificacion = 'ok';
// nueva fecha debe ser posterior a actual
if (nuevaFechaFin && fechaContratoFin) {
    if (nuevaFechaFin <= fechaContratoFin) {
        apex.message.alert("AVISO: La nueva fecha debe ser posterior a la fecha de contrato");
        $s('P13_NEW_F_FIN', '');
        $('#P13_NEW_F_FIN').trigger('change');
        estadoVerificacion = 'fail';
    }
}

// nueva fecha no debe pasar la fecha fin de proyecto
if (nuevaFechaFin && fechaProyectoFin) {
}

```

```

        if (nuevaFechaFin > fechaProyectoFin) {
            apex.message.alert("AVISO: La nueva fecha es posterior a la fecha de proyecto final");
            $s('#P13_NEW_F_FIN', '');
            $('#P13_NEW_F_FIN').trigger('change');
            estadoVerificacion = 'fail';
        }
    }

    // cambio color si ok
    const $boton = $('#btn_verifica');

    if ($boton.length) {
        // Elimina clases anteriores
        $boton.removeClass('t-Button--success t-Button--hot');

        // Aplica color segun estado
        if (estadoVerificacion === "ok") {
            $boton.addClass('t-Button--success');
        }
    }
}

```

Como en anteriores casos, si la fecha fuera errónea, habría que reactivar el campo de fecha:

Código C.25: Reactiva fecha renovación se es errónea

```
apex.event.trigger('#P13_NEW_F_FIN', 'change');
```

Finalmente como en el caso inicial de creación de nóminas, necesitamos ejecutar un **procedimiento** en **PL/SQL**, que genere las nuevas nóminas adicionales:

Código C.26: Genera las nuevas nóminas tras la renovación

```

DECLARE
    v_fecha_ini DATE := :P13_F_FIN;
    v_fecha_fin DATE := :P13_NEW_F_FIN;
    v_total_meses      PLS_INTEGER;
    v_meses_generados PLS_INTEGER := 0;
    v_mes_actual       DATE;

    v_ret_mes NUMBER := :P13_NEW_SAL;
    v_ss_mes NUMBER := :P13_NEW_SS;
    v_indemnizacion NUMBER := :P13_NEW_IND;

```

```

v_dni VARCHAR2(9);

v_ultimo_mes DATE;

BEGIN
EXECUTE IMMEDIATE 'ALTER SESSION SET NLS_NUMERIC_CHARACTERS=.,.';

-- Obtengo el DNI desde la tabla CONTRATO
SELECT CONTRATADO
INTO v_dni
FROM CONTRATOS
WHERE REF_CONTRATO = :P13_SEL_CONT;

-- Actualizo la nueva fecha fin del contrato e importes anado obs
UPDATE CONTRATOS
SET
F_FIN = :P13_NEW_F_FIN,
RET_MES = :P13_NEW_SAL,
SS_MES = :P13_NEW_SS,
INDEMNIZACION = :P13_NEW_IND,
OBSERVACIONES = 'Renovado'
WHERE REF_CONTRATO = :P13_SEL_CONT;

-- Buscamos el ultimo mes insertado anteriormente
SELECT TO_DATE(anno || LPAD(mes, 2, '0'), 'YYYYMM')
INTO v_ultimo_mes
FROM (
SELECT mes, anno
FROM NOMINA
WHERE dni_nom = v_dni
ORDER BY anno DESC, mes DESC
)
WHERE ROWNUM = 1;

-- Actualizar nueva retribucion del primer nuevo mes
UPDATE NOMINA
SET mensualidad = v_ret_mes
WHERE dni_nom = v_dni

```

```

AND mes = TO_NUMBER(TO_CHAR(v_ultimo_mes , 'MM')) )
AND anno = TO_NUMBER(TO_CHAR(v_ultimo_mes , 'YYYY')) );

-- Calculo de nuevos meses a generar
v_total_meses := MONTHS_BETWEEN(TRUNC(v_fecha_fin , 'MM')) , A

-- Insercion nuevos meses con salario y SS
WHILE v_meses_generados < v_total_meses LOOP
  v_mes_actual := ADD_MONTHS(TRUNC(v_ultimo_mes , 'MM')) , v_me

INSERT INTO NOMINA (
  dni_nom , mes , anno , mensualidad , seg_soc , observaciones
) VALUES (
  v_dni ,
  TO_NUMBER(TO_CHAR(v_mes_actual , 'MM')) ,
  TO_NUMBER(TO_CHAR(v_mes_actual , 'YYYY')) ,
  v_ret_mes ,
  v_ss_mes ,
  NULL
);

v_meses_generados := v_meses_generados + 1;
END LOOP;

-- Insertar el ultimo mes adicional solo con la indemnizacion
v_mes_actual := ADD_MONTHS(TRUNC(v_ultimo_mes , 'MM')) , v_tot

INSERT INTO NOMINA (
  dni_nom , mes , anno , mensualidad , seg_soc , observaciones
) VALUES (
  v_dni ,
  TO_NUMBER(TO_CHAR(v_mes_actual , 'MM')) ,
  TO_NUMBER(TO_CHAR(v_mes_actual , 'YYYY')) ,
  v_indemnizacion ,
  v_ss_mes ,
  'Indemnizacion_final'
);
END;

```

## Página Renuncia al contrato

Para realizar la renuncia de un contrato, primero se seleccionan aquellos contratos que tienen una fecha posterior a la actual con una **acción dinámica**, tal y como en el caso anterior de la **renovación** a través de una consulta **SQL**, guardando el valor de la seg. social en un campo oculto **P14\_SS\_OCULTA**, ya que es un valor que necesitaremos después, mediante la segunda parte de esta acción en **JS**:

Código C.27: Guarda seguridad social en campo oculto  
`$s("P14_SS_OCULTA", $v("P14_SEG_SOG"));`

Esta primera parte nos presentará los datos del contrato a renunciar, debiendo introducir después, la fecha de renuncia, que se verificará a través de otra acción dinámica asociada al botón, **Verificar Fecha**, que comprueba si la fecha de renuncia introducida es menor que la actual fecha fin y si es posterior a la actual, a través del siguiente código **JS**:

Código C.28: Verifica las fecha de renuncia del contrato

```
// parsear fechas en formato 'DD-MM-YYYY' o 'DD-MON-YYYY'
function parseFechaFlexible(fechaStr) {
    if (!fechaStr) {
        console.warn('Fecha vacia o null:', fechaStr);
        return null;
    }

    const partes = fechaStr.trim().split(' - ');
    if (partes.length !== 3) {
        console.error('Formato inesperado:', fechaStr);
        return null;
    }

    const dia = parseInt(partes[0], 10);
    const año = parseInt(partes[2], 10);

    // Detectar si el mes es numero o texto
    const mesTexto = partes[1].toUpperCase();
    const meses = {
        'ENE': 0, 'FEB': 1, 'MAR': 2, 'ABR': 3,
        'MAY': 4, 'JUN': 5, 'JUL': 6, 'AGO': 7,
        'SEP': 8, 'OCT': 9, 'NOV': 10, 'DIC': 11
    };
}
```

```

let mes;
if (!isNaN(parseInt(mesTexto, 10))) {
    mes = parseInt(mesTexto, 10) - 1; // formato numerico
} else {
    mes = meses[mesTexto]; // formato texto
}

if (mes === undefined || isNaN(dia) || isNaN(año)) {
    console.error('Problema al interpretar partes de la fecha');
    return null;
}

return new Date(año, mes, dia);
}

// Leer valores
const fechaInicioStr      = $v('P14_F_INI');
const fechaContratoFinStr  = $v('P14_F_FIN');
const nuevaFechaFinStr     = $v('P14_F_RENUNCIA');
// obtenemos la fecha del actual
const hoy = new Date();
// Extrae año, mes y dia
const añoHoy = hoy.getFullYear();
const mesHoy = hoy.getMonth() + 1;
const diaHoy = hoy.getDate();
const fechaHoy = new Date(añoHoy, mesHoy - 1, diaHoy);

// Parsear las fechas
const fechaInicio          = parseFechaFlexible(fechaInicioStr);
const fechaContratoFin     = parseFechaFlexible(fechaContratoFinStr);
const nuevaFechaFin         = parseFechaFlexible(nuevaFechaFinStr);

// Logs para verificar valores
console.log('Fecha_INICIO(raw):', fechaInicioStr);
console.log('Fecha_FIN_actual(raw):', fechaContratoFinStr);
console.log('Nueva_fecha_FIN(raw):', nuevaFechaFinStr);

console.log('INICIO(obj):', fechaInicio?.toLocaleDateString('es-ES'));
console.log('FIN_actual(obj):', fechaContratoFin?.toLocaleDateString('es-ES'));

```

```

console.log('NuevaFIN(obj):', nuevaFechaFin?.toLocaleDateString('es-ES'))
console.log('fechaHoy(obj):', nuevaFechaFin?.toLocaleDateString('es-ES'))
// semaforo
let estadoVerificacion = 'ok';
// nueva fecha debe ser posterior a actual
if (nuevaFechaFin && fechaContratoFin) {
    if (nuevaFechaFin > fechaContratoFin) {
        apex.message.alert("AVISO: La fecha de renuncia debe ser posterior a la de inicio y a la actual");
        $('#P14_F_RENUNCIA').trigger('change');
        estadoVerificacion = 'fail';
    }
}
// debe ser posterior a la de inicio y a la actual
if (nuevaFechaFin && fechaContratoFin) {
    if (nuevaFechaFin < fechaInicio || nuevaFechaFin < fechaHoy) {
        apex.message.alert("AVISO: La fecha de renuncia debe ser posterior a la de inicio y a la actual");
        $('#P14_F_RENUNCIA').trigger('change');
        estadoVerificacion = 'fail';
    }
}

// cambio color si ok
const $boton = $('#btn_verifica2');

if ($boton.length) {
    // Elimina clases anteriores
    $boton.removeClass('t-Button--success t-Button--hot t-Button--dark');

    // Aplica color segun estado
    if (estadoVerificacion === "ok") {
        $boton.addClass('t-Button--success');
        // HAY QUE CAMBIAR EL FORMATO DE LA SS MOSTRADA
        var ssRaw = $v("P14_SS_SOG");
        var ssClean = ssRaw.replace('E', ',')
            .replace(/\s/g, "");
        $s("P14_SS_OCULTA", ssClean);
    }
}

```

```

    }
}
```

Una vez verificada la fecha, con el botón en verde, y pulsado **Renunciar**, se ejecuta un proceso **PL/SQL**, que reduce las nóminas hasta la nueva fecha fin, y también cambia la de fin del contrato:

Código C.29: Genera las nuevas nóminas tras la renuncia

```

DECLARE
    v_fecha_ini      DATE := :P14_F_FIN;
    v_fecha_fin      DATE := :P14_F_RENUNCIA;
    v_total_meses   PLS_INTEGER;
    v_meses_generados PLS_INTEGER := 0;
    v_mes_actual     DATE;
    v_ss_mes         NUMBER := :P14_SS_OCULTA;
    v_indemnizacion NUMBER := :P14_NEW_IND;

    v_dni            VARCHAR2(9);
    v_ultimo_mes     DATE;

BEGIN
EXECUTE IMMEDIATE 'ALTER SESSION SET NLS_NUMERIC_CHARACTERS
                     .';

-- Obtengo el DNI
SELECT CONTRATADO
INTO v_dni
FROM CONTRATOS
WHERE REF_CONTRATO = :P14_SEL_CONT;
DBMS_OUTPUT.PUT_LINE('DNI obtenido: ' || v_dni);
-- Actualizo la fecha fin del contrato
UPDATE CONTRATOS
SET
    F_FIN = :P14_F_RENUNCIA,
    INDEMNIZACION = v_indemnizacion ,
    OBSERVACIONES = 'Renuncia';
WHERE REF_CONTRATO = :P14_SEL_CONT;
-- elimino nominas posteriores a la renuncia
DELETE FROM NOMINA
WHERE dni_nom = v_dni
AND TO_DATE(anno || LPAD(mes, 2, '0'), 'YYYYMM') > TRUNC(v_
```

```

-- veo cual es ultimo mes de nomina
SELECT TO_DATE(anno || LPAD(mes, 2, '0'), 'YYYYMM')
INTO v_ultimo_mes
FROM (
  SELECT mes, anno
  FROM NOMINA
  WHERE dni_nom = v_dni
  ORDER BY anno DESC, mes DESC
)
WHERE ROWNUM = 1;

-- Actualizo ultima numina con su nueva indemnizacion
UPDATE NOMINA
SET mensualidad = v_indemnizacion ,
seg_soc = v_ss_mes ,
observaciones = 'Ultimo_mes_con_indemnizacion'
WHERE dni_nom = v_dni
AND TO_DATE(anno || LPAD(mes, 2, '0'), 'YYYYMM') = TRUNC(v_fecha)

-- Annado el siguiente mes con solo ss
v_mes_actual := ADD_MONTHS(TRUNC(v_fecha_fin, 'MM'), 1);

INSERT INTO NOMINA (
dni_nom, mes, anno, mensualidad, seg_soc, observaciones
) VALUES (
v_dni,
TO_NUMBER(TO_CHAR(v_mes_actual, 'MM')) ,
TO_NUMBER(TO_CHAR(v_mes_actual, 'YYYY')) ,
0,
v_ss_mes,
'ultima_ss'
);

END;

```

## C.8. Página de Informes

En las siguientes páginas se muestran los informes realizados.

## Página Nómina-mes

Este informe es el objetivo final de la aplicación, para ello se solicitan primeramente un mes y un año y se realiza una consulta **SQL**, al pulsar el botón **Consultar** y devuelve la suma total del mes para todas las aplicaciones en :

Código C.30: Consulta informe nómina para un mes determinado

```

SELECT
o.ORGANICA,
s.NOM_SOL || ' ' || s.APE1_SOL || ' ' || s.APE2_SOL AS NOMBRE,
n.MENSUALIDAD,
n.SEG_SOC,
n.MENSUALIDAD + n.SEG_SOC AS TOTAL,
n.OBSERVACIONES,
:P18_MES AS MES_PEDIDO,
:P18_ANNO AS ANNO_PEDIDO,
(
SELECT SUM(n2.MENSUALIDAD + n2.SEG_SOC)
FROM PROYECTO o2
JOIN CONVOCATORIA c2 ON c2.REF_PROY = o2.ORGANICA
JOIN CONTRATOS ct2 ON ct2.CONVOCATORIA = c2.REFERENCIA
JOIN SOLICITANTE s2 ON s2.DNI_SOL = ct2.CONTRATADO
JOIN NOMINA n2 ON n2.DNI_NOM = s2.DNI_SOL
WHERE n2.MES = :P18_MES
AND n2.ANNO = :P18_ANNO
AND (:P18_FILTRO IS NULL OR o2.ORGANICA = :P18_FILTRO)
) AS TOTAL_GENERAL
FROM PROYECTO o
JOIN CONVOCATORIA c ON c.REF_PROY = o.ORGANICA
JOIN CONTRATOS ct ON ct.CONVOCATORIA = c.REFERENCIA
JOIN SOLICITANTE s ON s.DNI_SOL = ct.CONTRATADO
JOIN NOMINA n ON n.DNI_NOM = s.DNI_SOL
WHERE n.MES = :P18_MES
AND n.ANNO = :P18_ANNO
AND (:P18_FILTRO IS NULL OR o.ORGANICA = :P18_FILTRO)

```

De este informe se guarda el total en P18\_T\_GEN, a través de una **acción dinámica** en **JS**. Esto lo podría hacer el propio informe dinámico, pero nos interesa para imprimir el informe en pdf:

Código C.31: Guarda y muestra el total del listado

```

$( "td[headers='TOTAL_GENERAL']" ).first().text().trim();
total = total.replace(",","");

```

```
$s("P18_T_GEN", total);
```

Mostrando el informe, podemos filtrar la orgánica que queramos y que vuelva a calcular el importe total. Primero mostramos todas las referencias que salen en el informe del mes, con una consulta **SQL**:

Código C.32: Muestra las referencias de proyectos del informe para seleccionar

```
SELECT NULL AS display_value, NULL AS return_value
UNION
SELECT DISTINCT o.ORGANICA AS display_value, o.ORGANICA AS return_value
FROM PROYECTO o
JOIN CONVOCATORIA c ON c.REF_PROY = o.ORGANICA
JOIN CONTRATOS ct ON ct.CONVOCATORIA = c.REFERENCIA
JOIN SOLICITANTE s ON s.DNI_SOL = ct.CONTRATADO
JOIN NOMINA n ON n.DNI_NOM = s.DNI_SOL
WHERE n.MES = :P18_MES
AND n.ANNO = :P18_ANNO
ORDER BY 1
```

Y con la acción dinámica anterior se vuelve a refrescar el importe total.

Para poder imprimir el informe a PDF a través de **APEX Office Print (AOP)**, es precisa otra acción dinámica que dispare el **plug-in UC-APEX OfficePrint(AOP)-DA**, al pulsar el botón **Generar PDF**. Indicándole cual es nuestra plantilla personalizada, guardada en shared components: **plantilla\_informe\_nomina.docx**

## Página Vencimientos

En este informe interactivo, se presentarán los vencimientos de contratos entre dos fechas, para su control, a través de una consulta **SQL**, al pulsar el botón **Consulta** que activará la acción dinámica de submit al informe.

Código C.33: Muestra de contratos que vencen en periodo de fechas

```
SELECT
S.APE1_SOL || ' ' || S.APE2_SOL || ', ' || S.NOM_SOL AS CONTRATADO,
CT.REF_CONTRATO AS CONTRATO,
CT.F_FIN AS FIN,
P.ORGANICA AS ORGANICA
FROM SOLICITANTE S
INNER JOIN CONTRATOS CT ON CT.CONTRATADO = S.DNI_SOL
INNER JOIN CONVOCATORIA CV ON CV.REFERENCIA = CT.CONVOCATORIA
```

```
INNER JOIN PROYECTO P ON P.ORGANICA = CV.REF_PROY
WHERE CT.F_FIN BETWEEN TO_DATE(:P17_FDESDE, 'DD/MM/YYYY') AND
```

## Página Contratos

Esta página es un informe interactivo simple que muestra los datos de todos los contratos a través de una consulta **SQL**, pudiendo realizar los filtros predeterminados que ofrece **Actions**:

Código C.34: Consulta para todos los contratos

```
SELECT
    S.APE1_SOL || ' ' || S.APE2_SOL || ', ' || S.NOM_SOL AS CONTRATO,
    CT.REF_CONTRATO AS CONTRATO,
    CT.F_INI AS INICIO,
    CT.F_FIN AS FIN,
    CV.REFERENCIA AS REF_CONV,
    P.ORGANICA AS ORGANICA,
    P.FECHA_INI ASINI_PROY,
    P.FECHA_FIN AS FIN_PROY
FROM SOLICITANTE S
INNER JOIN CONTRATOS CT ON CT.CONTRATADO = S.DNI_SOL
INNER JOIN CONVOCATORIA CV ON CV.REFERENCIA = CT.CONVOCATORIA
INNER JOIN PROYECTO P ON P.ORGANICA = CV.REF_PROY
```

## Página Nóminas-Periodo

Aquí se presentan para un contratado las nóminas que se han pagado en un periodo de tiempo solicitado. Para ello se realiza una primera consulta **SQL**, que muestra los contratados:

Código C.35: Consulta de contratados para selección

```
SELECT
    NOM_SOL || ' ' || APE1_SOL || ' ' || APE2_SOL AS display_value,
    DNI_SOL AS return_value
FROM SOLICITANTE
JOIN CONTRATOS C ON C.CONTRATADO = DNI_SOL
ORDER BY NOM_SOL;
```

Una vez seleccionado el contratado e introducidas las fechas de inicio y fin se realiza otra consulta **SQL**, para llenar el informe y mostrando el total pagado en ese periodo. En este caso se ha suprimido la opción **Actions**, ya que el informe es único:

Código C.36: Consulta de nóminas para un contratado en un intervalo de tiempo

```
SELECT
    n.MES,
    n.ANNO AS ANIO,
    n.MENSUALIDAD AS MENSUALIDAD,
    n.SEG_SOC AS SEG_SOC,
    n.MENSUALIDAD + n.SEG_SOC AS TOTAL,
    n.OBSERVACIONES
FROM SOLICITANTE s
JOIN CONTRATOS ct ON s.DNI_SOL = ct.CONTRATADO
JOIN NOMINA n ON n.DNI_NOM = s.DNI_SOL
WHERE s.DNI_SOL=:P19_CONTRATADO AND ct.F_INI>= :P19 DESDE AND ct.F_
```



## *Apéndice D*

---

# Configuración

---

En esta sección se indicarán los parámetros principales para la configuración, tanto de la **BDR**, **ORDS**, **APEX**, **AOP** y **Oracle Cloud**.

## D.1. Oracle 23ai

El proceso de instalación se realiza siguiendo las instrucciones de Oracle 23ai [3]. Esto contiene una **Plugable Data Base (PBD)**, **FREEPDB1** donde creamos el esquema de trabajo para nuestro proyecto **TFGUBU** en "*C:/app/User/product23ai/oradata/FREE/FREEPDB1*"

Una vez instalado y a través de **SQL plus**, procedemos a crear nuestro entorno conectados como sys as sysdba/1234.

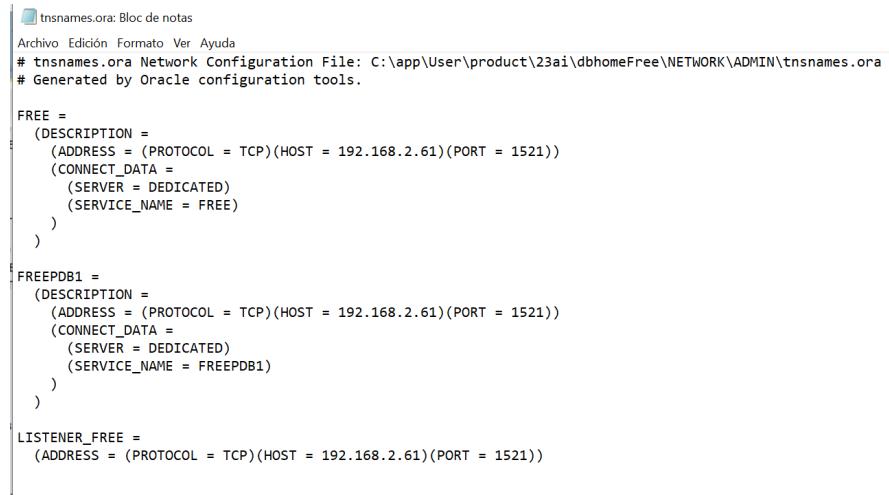
Código D.1: Creación de Esquema TFGUBU

```
-- cambiamos de contenedor
ALTER SESSION SET CONTAINER = FREEPDB1;

-- creamos nuestro usuario-esquema
CREATE USER TFGUBU IDENTIFIED BY "tu passwd"
DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP
QUOTA UNLIMITED ON USERS;

-- concedemos permisos
GRANT CONNECT , RESOURCE TO TFGUBU;
GRANT CREATE SESSION TO TFGUBU;
GRANT CREATE TABLE, CREATE VIEW, CREATE PROCEDURE TO TFGUBU;
```

vemos en la imagen cómo queda configurada la conexión, en nuestro caso para **FREEPDB1**:



```

tnsnames.ora: Bloc de notas
Archivo Edición Formato Ver Ayuda
# tnsnames.ora Network Configuration File: C:\app\user\product\23ai\dbhomeFree\NETWORK\ADMIN\tnsnames.ora
# Generated by Oracle configuration tools.

FREE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.2.61)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = FREE)
  )
)

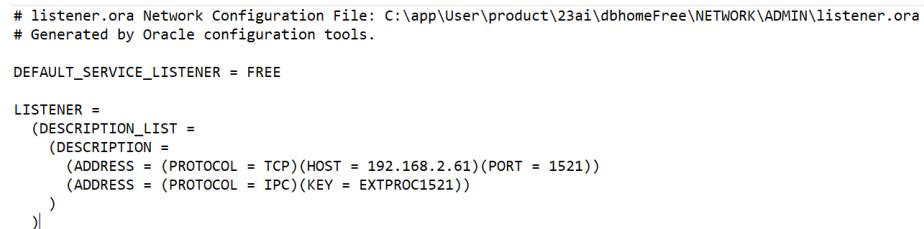
FREEPDB1 =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.2.61)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = FREEPDB1)
  )
)

LISTENER_FREE =
(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.2.61)(PORT = 1521))

```

Figura D.1: Configuración conexión a FREEPDB1

Este archivo permite definir un alias de conexión para acceder al servicio y en la siguiente imagen, vemos como quedaría la configuración del **listener** que gestiona las solicitudes de conexión entrantes:



```

# listener.ora Network Configuration File: C:\app\user\product\23ai\dbhomeFree\NETWORK\ADMIN\listener.ora
# Generated by Oracle configuration tools.

DEFAULT_SERVICE_LISTENER = FREE

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.2.61)(PORT = 1521))
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
  )
)
```

Figura D.2: Configuración Listener a FREEPDB1

Hay que tener en cuenta que los servicios de Oracle y Listener deben estar activos, si no, no se podrá realizar la conexión.



Figura D.3: Servicios activados de Oracle

Lo que podremos comprobar con el estatus del linter con **lsnrctl status**:

```
C:\Users\User>lsnrctl status

LSNRCTL for 64-bit Windows: Version 23.0.0.0.0 - Production on 21-AGO-2025 17:21:36
Copyright (c) 1991, 2024, Oracle. All rights reserved.

Conectándose a (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.2.61)(PORT=1521)))
ESTADO del LISTENER
-----
Alias LISTENER
Versión TNSLSNR for 64-bit Windows: Version 23.0.0.0.0 - Production
Fecha de Inicio 21-AGO-2025 16:31:42
Tiempo Actividad 0 días 0 hr. 49 min. 53 seg.
Nivel de Rastreo off
Seguridad ON: Local OS Authentication
SNMP OFF
Servicio por Defecto FREE
Parámetros del Listener C:\app\user\product\23ai\dbhomeFree\network\admin\listener.ora
Log del Listener C:\app\user\product\23ai\diag\tnslsnr\DESKTOP-UFQIAI9\listener\alert\log.xml
Recibiendo Resumen de Puntos Finales...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.2.61)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\.\pipe\EXTPROC1521ipc))

Resumen de Servicios...
El servicio "378ef548113d40859fe5c0cb202b023c" tiene 1 instancia(s).
  La instancia "free", con estado READY, tiene 2 manejador(es) para este servicio...
El servicio "FREE" tiene 1 instancia(s).
  La instancia "free", con estado READY, tiene 2 manejador(es) para este servicio...
El servicio "FREEXDB" tiene 1 instancia(s).
  La instancia "free", con estado READY, tiene 1 manejador(es) para este servicio...
El servicio "freedb1" tiene 1 instancia(s).
  La instancia "free", con estado READY, tiene 2 manejador(es) para este servicio...
El comando ha terminado correctamente
```

Figura D.4: Verificación de Listener activo

A partir de este momento podríamos empezar a crear nuestras tablas con **SQL Developer**, con la conexión indicada, nuestro usuario: **TFGUBU** y la contraseña establecida.

## D.2. Oracle REST Data Service

**Oracle REST Data Services (ORDS)** es una herramienta desarrollada por Oracle que permite acceder a sus bases de datos a través de servicios (RESTFUL) **Representational State Transfer**, esto permite acceder a la BD como si fuera una web sin usar **SQL**, si no con peticiones **HTTPS**, pudiendo realizar las operaciones **CRUD**, fácilmente.

Una vez realizada la descarga e instalación (**Oracle REST**), lo instalamos localmente, en mi caso:

```

Directorio de C:\ords-latest

04/08/2025 13:46 <DIR> .
04/08/2025 13:46 <DIR> ..
14/06/2025 10:58 <DIR> apex
14/06/2025 08:12 <DIR> bin
19/06/2025 21:08 <DIR> certs
19/06/2025 17:55 <DIR> config
13/06/2025 23:23 <DIR> databases
21/05/2025 17:21 <DIR> docs
12/05/2025 19:42 <DIR> examples
13/06/2025 23:23 <DIR> global
04/08/2025 13:46 145.948 hs_err_pid10020.log
12/05/2025 19:42 <DIR> icons
12/05/2025 19:42 <DIR> lib
12/05/2025 19:42 5.993 LICENSE.txt
21/05/2025 17:21 <DIR> linux-support
12/05/2025 19:42 210 NOTICE.txt
21/05/2025 17:20 137.823.838 ords.war
12/05/2025 19:42 <DIR> scripts
12/05/2025 19:42 365 THIRD-PARTY-LICENSES.txt
      5 archivos 137.976.354 bytes
     14 dirs 75.532.910.592 bytes libres

```

Figura D.5: Directorio de instalación ORDS

.6

En el cual destacamos:

- **ords.war**: Archivo ejecutable principal
- **dir bin**: Scripts de instalación
- **dir databases**:
  - dir default: Contiene la configuración por defecto, el subdirectorio **images** que contiene las imágenes de *Application Express (APEX)* y el fichero de configuración de la conexión a la BD **pool.xml**

Así la conexión entre ORDS y la base de datos se define en este fichero **pool.xml** donde especificamos el puerto **1521**

```
*pool.xml: Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Saved on Fri Jun 13 22:02:36 UTC 2025</comment>
<entry key="db.connectionType">basic</entry>
<entry key="db.hostname">192.168.2.61</entry>
<entry key="db.port">1521</entry>
<entry key="db.servicename">FREEPDB1</entry>
<entry key="db.username">ORDS_PUBLIC_USER</entry>
<entry key="feature.sdw">true</entry>
<entry key="plsql.gateway.mode">proxied</entry>
<entry key="restEnabledSql.active">true</entry>
<entry key="security.requestValidationFunction">ords_util.authorize_plsql_gateway</entry>
</properties>
```

Figura D.6: Fichero de configuración pool.xml

Para el despliegue de la aplicación y una vez que se decidió hacerlo en **Oracle Cloud Infrastructure (OCI)**, fue necesaria la configuración del **HTTPS**. Así se generó un certificado autofirmado privado, con OpenSSL, para ello seguimos los siguientes pasos:

- Descargar OpenSSL
- Generar la clave privada: **openssl genrsa -out key.pem 2048**
- Crear solicitud de certificado: **openssl req -new -key key.pem -out cert.csr**
- Generar el cert autofirmado: **openssl x509 -req -in cert.csr -signkey key.pem -out cert.pem -days 365**
- Crear un archivo combinado: **cat cert.pem key.pem > combina-do.pem** (Requerido por APEX)

Una vez hecho esto tendremos que configurar el fichero **settings.xml**, para que tome este certificado y el puerto 8443.

```

settings.xml: Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Saved on Thu Jun 19 16:54:29 UTC 2025</comment>
<entry key="database.api.enabled">true</entry>
<entry key="standalone.doc.root">C:/ords-latest/apex/images</entry>
<!--entry key="standalone.doc.root">C:/ords-latest/bin\global\doc_root</entry> -->
<entry key="standalone.https.cert">C:/ords-latest/certs/cert.pem</entry>
<entry key="standalone.https.cert.key">C:/ords-latest/certs/key.pem</entry>
<!--entry key="standalone.https.key">C:/ords-latest/certs/key.pem</entry> -->
<entry key="standalone.https.port">8443</entry>
</properties>

```

Figura D.7: Configuración del fichero settings.xml

Una vez configurado todo podemos acceder a través del navegador a los servicios ORDS con **HTTPS**:

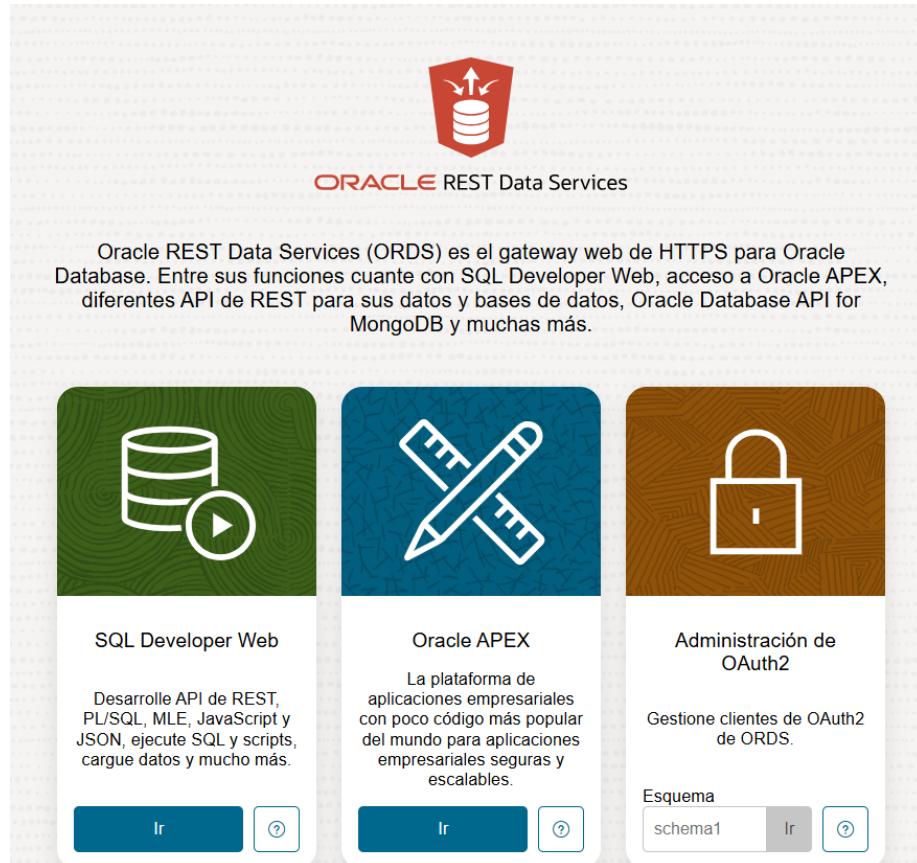
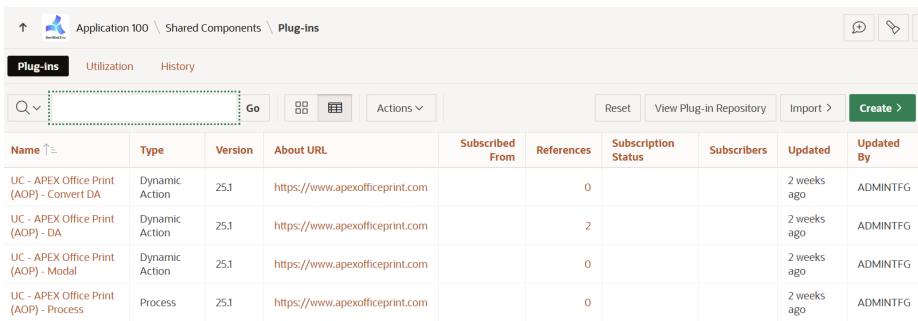


Figura D.8: Imagen de acceso web a servicios ORDS

## D.3. Apex Office Print AOP

Como uno de los objetivos de la aplicación era crear un informe personalizado desde **APEX** en formato **PDF**, se procedió a instalar la herramienta creada por Oracle para este propósito. [2] Una vez descargada desde **Web AOP**, tendremos que instalar los plug-ins necesarios en nuestra aplicación **GeNomIn**. Para ello accedemos a **Shared Components/Plug-ins** y procedemos a importar los ficheros:



Name	Type	Version	About URL	Subscribed From	References	Subscription Status	Subscribers	Updated	Updated By
UC - APEX Office Print (AOP) - Convert DA	Dynamic Action	25.1	<a href="https://www.apexofficeprint.com">https://www.apexofficeprint.com</a>		0			2 weeks ago	ADMINTFG
UC - APEX Office Print (AOP) - DA	Dynamic Action	25.1	<a href="https://www.apexofficeprint.com">https://www.apexofficeprint.com</a>		2			2 weeks ago	ADMINTFG
UC - APEX Office Print (AOP) - Modal	Dynamic Action	25.1	<a href="https://www.apexofficeprint.com">https://www.apexofficeprint.com</a>		0			2 weeks ago	ADMINTFG
UC - APEX Office Print (AOP) - Process	Process	25.1	<a href="https://www.apexofficeprint.com">https://www.apexofficeprint.com</a>		0			2 weeks ago	ADMINTFG

Figura D.9: Vista de la instalación de plug-ins de AOP

Una vez instalados, deberemos proceder a la configuración de cada plug-in para su correcto funcionamiento, en **Shared Components/Component Settings** escogeremos cada plug-in y e introduciremos la web (en este caso al ser local http) y la **API key**, que nos identifica en el servicio y el modo de AOP, en este caso ya en Producción para que no inserte marca de agua.

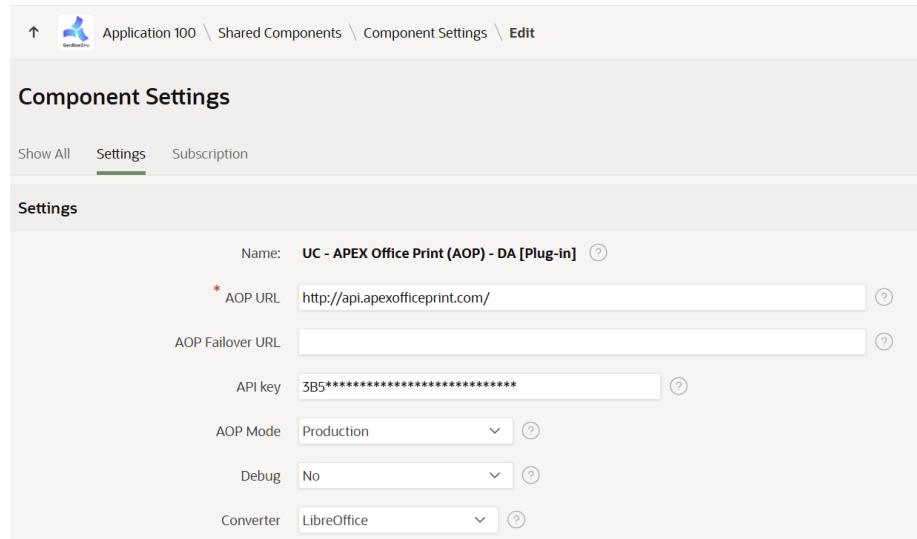


Figura D.10: Confiuguración de los plug-ins AOP

Solamente nos quedaría realizar nuestra plantilla personalizada en **Word**, indicando los campos que deben aparecer:



Figura D.11: Visualización de la plantilla AOP

y luego relacionándolo en le informe correspondiente (**Informe-nomina**)

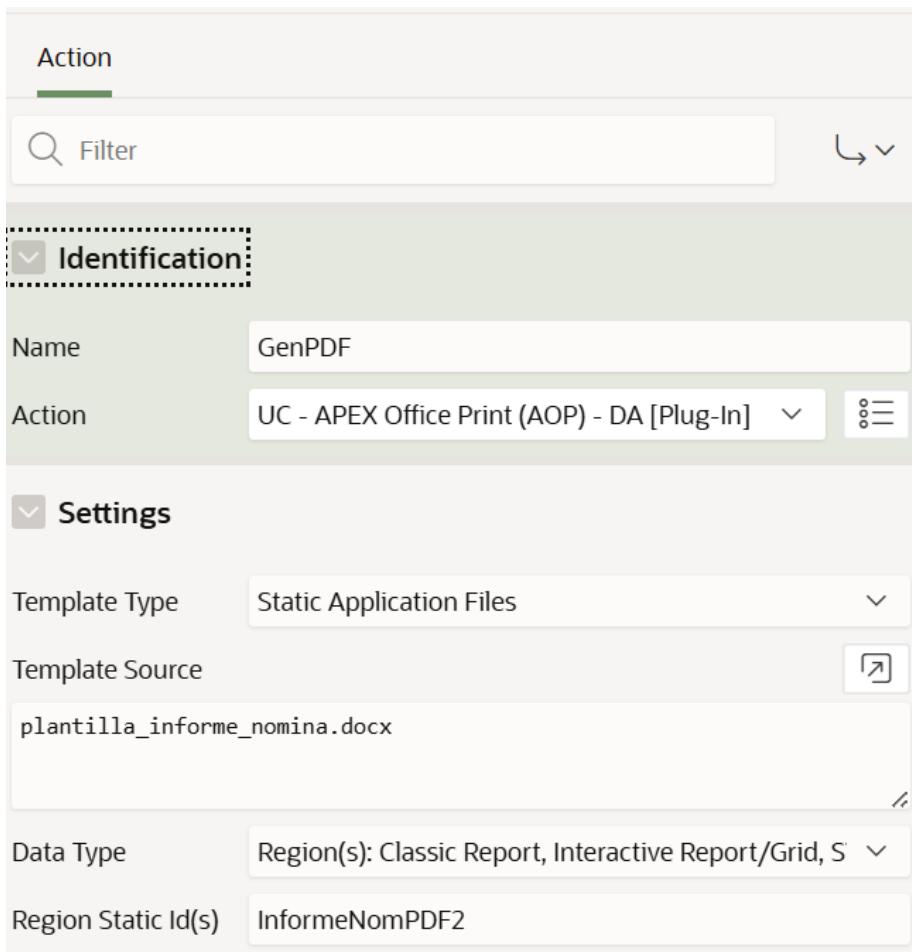


Figura D.12: Asociación de la plantilla al informe

## D.4. Oracle Cloud Infraestructura

Como ya se comentó en el **Capítulo 4.5** de la memoria, el mayor problema de **OCI**, es el registro, que no lleva asociada más capacidad intelectual que la paciencia. Una vez conseguido este propósito, procederíamos a crear nuestra instancia de la base de datos como **Base de datos autónoma** [?]

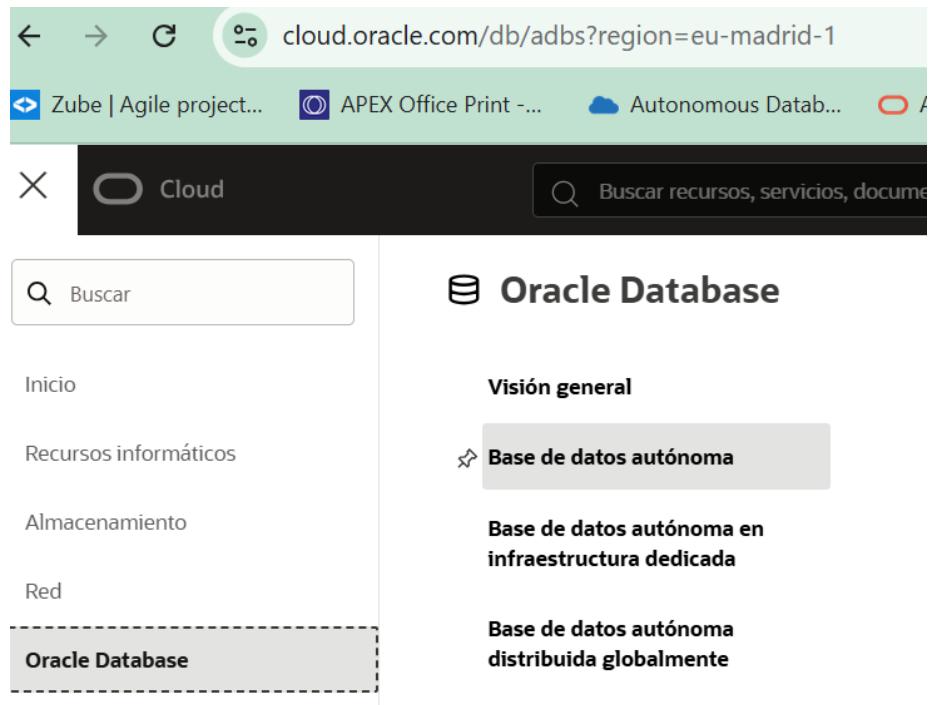


Figura D.13: Menu Oracle Database Infraestructure

En el siguiente paso deberemos configurar los datos principales, teniendo especial cuidado con la región de creación, en nuestro caso **Spain Central (Madrid)**. Estableceremos el nombre de la base de datos y la carga de trabajo. Como nosotros vamos a utilizar **APEX**, es lo que escogemos.

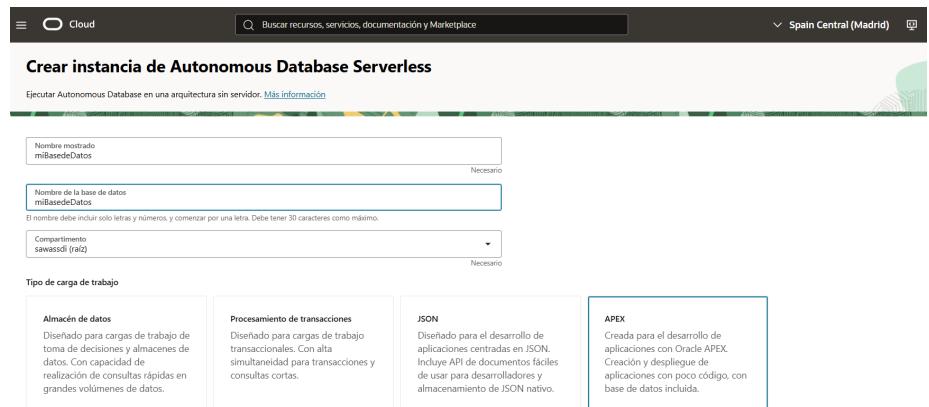


Figura D.14: Creacion de la Instancia de BD-1 en Cloud

Finalizaremos la configuración escogiendo la opción **siempre gratis**, seleccionando la versión **23ai**, como en nuestro modo local, introduciendo la contraseña de Administrador y el modo **Acceso seguro desde cualquier lugar**, para permitir a los usuarios con credenciales acceso desde Internet.

The screenshot shows the Oracle Cloud Infrastructure (OCI) console interface for creating a new database instance. The process is divided into several sections:

- Configuración de base de datos**: This section includes a "Siempre gratis" button, a note about free resources, and a "Desarrollador" checkbox. A dropdown menu shows "23ai" selected under "Seleccionar versión de base de datos".
- Opciones Avanzadas**: A collapsed section containing "Creación de credenciales de administrador" (Administrator Credential Creation).
- Creación de credenciales de administrador**: Fields for "Nombre de usuario" (ADMIN) and two password fields ("Contraseña" and "Confirmar contraseña").
- Acceso de Red**: A section for "Tipo de acceso" (Network Type). It shows three options: "Acceso seguro desde cualquier lugar" (selected), "Acceso seguro solo desde direcciones IP y VCN permitidas", and "Solo acceso de punto final privado".
- Contactos para notificaciones y anuncios operativos**: A section for "Correos electrónicos de contacto" (Email contacts) with a "Agregar contacto de cliente" (Add customer contact) button.

Figura D.15: Creacion de la Instancia de BD-1 en Cloud

Una vez creada la base de datos, procederemos a crear la Instancia de **APEX**, con similares parámetros:

The screenshot shows the 'Crear servicio APEX' (Create APEX Service) configuration interface. It includes fields for service name, database name, compartment, and service type, along with optional developer and version selection.

Configuración	Valor	Obligatorio
Nombre mostrado	miInstanciaAPEX	Necesario
Nombre de la base de datos	miInstanciaAPEX	
El nombre debe incluir solo letras y números, y comenzar por una letra. Debe tener 30 caracteres como máximo.		
Compartimiento	sawassdi (raíz)	Necesario
Siempre gratis	<input checked="" type="checkbox"/>	
Los recursos siempre gratis se pueden crear tanto en las cuentas gratuitas como en las de pago. Estos recursos nunca generarán costo alguno. <a href="#">Más información</a> sobre la base de datos autónoma siempre gratis.		
Desarrollador	<input type="checkbox"/>	
Seleccionar versión de base de datos	23ai	Necesario

Figura D.16: Creación de Instancia APEX en Cloud

A partir de este momento podríamos iniciar la instancia de **APEX**, importar nuestra base de datos y la aplicación locales y se habría realizado el despliegue. Hay que tener en cuenta que los plug-ins de **AOP**, hay que volver a instalar y configurarlos.

## *Apéndice E*

---

# **Manual de Usuario**

---



## **E.1. Tabla de contenido**

Objetivo y proceso 2

Administración 2

Responsables 2

Proyectos 3

Convocatorias 4

Solicitante 5

Contratos 6

Crear 6

Renovación 7

Renuncia 8

Informes 10

Nómina-mes 10

Vencimientos 11

Contratos 11

Nominas-Periodo 12

Acerca de 12

Guía de uso 12

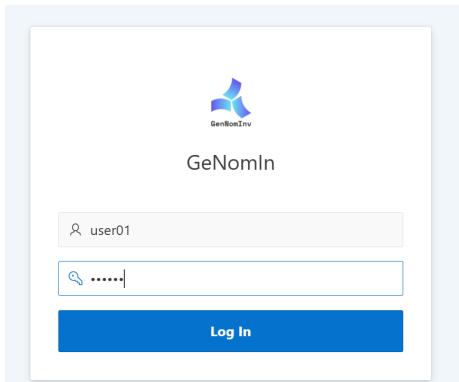
## **E.2. Objetivo y proceso**

El objetivo de la aplicación **GeNomIn**, es la creación, mantenimiento y gestión de las nóminas vinculadas a proyectos de investigación generados por el Servicio de Investigación de la Universidad de Burgos.

Para ello se ha establecido un proceso por el cual se irán desarrollando los distintos elementos que conformar un contrato, desde el mantenimiento de **Proyectos e Responsables** (Investigadores), **Convocatorias**, **Solicitantes** de las mismas, creación de **Contratos**, renovación y renuncia, y el objetivo final que es la generación de **Informes** de nómina mensuales, vencimientos de contratos y controles de nómina, para el cotejo con los informes emitidos por el Servicio de Retribuciones que actualmente se llevan con tablas Excel.

En esta guía de uso repasamos el proceso completo desde la administración de proyectos y responsables hasta el informe final de nómina mensual.

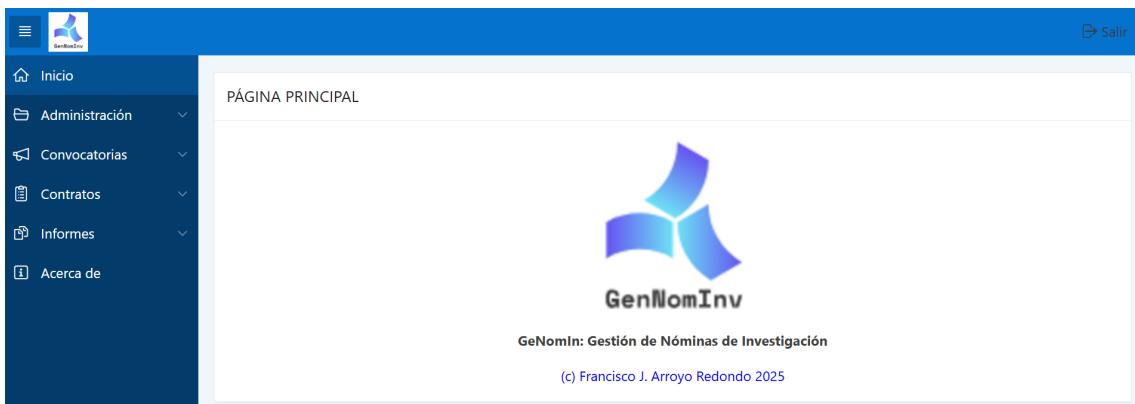
## E.3. Inicio de Sesión:



Una vez accedida a la aplicación nos pedirá las credenciales de entrada, en este caso accederemos con user01, user01.

Entrando al Menú principal, encontramos las siguientes opciones en el Menú Lateral, **Aministración**, **Convocatorias**, **Contratos**, **Informes** y **Acerca de**, que pasamos a detallar.

También podremos volver al log inicial pulsando en el botón **Salir**, de la derecha de la pantalla.

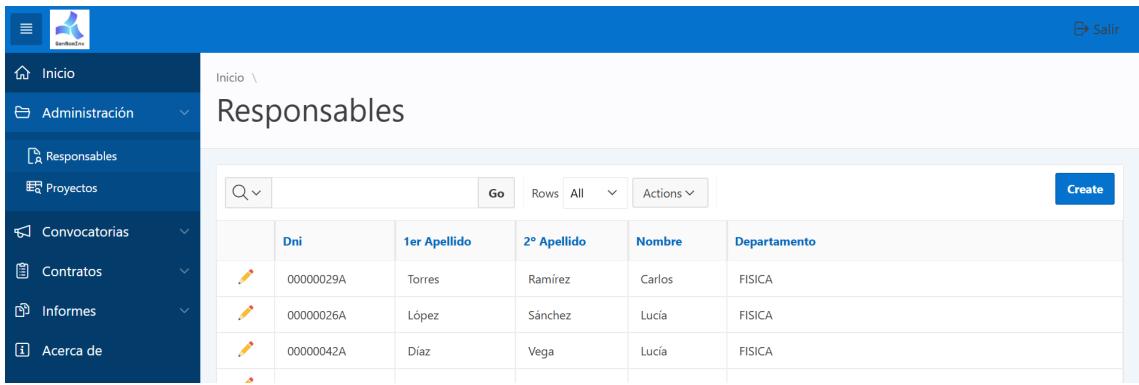


## E.4. Administración

Dentro del menú de **Administración**, podremos realizar la gestión de los **Proyectos** y de los **Responsables** de los mismos

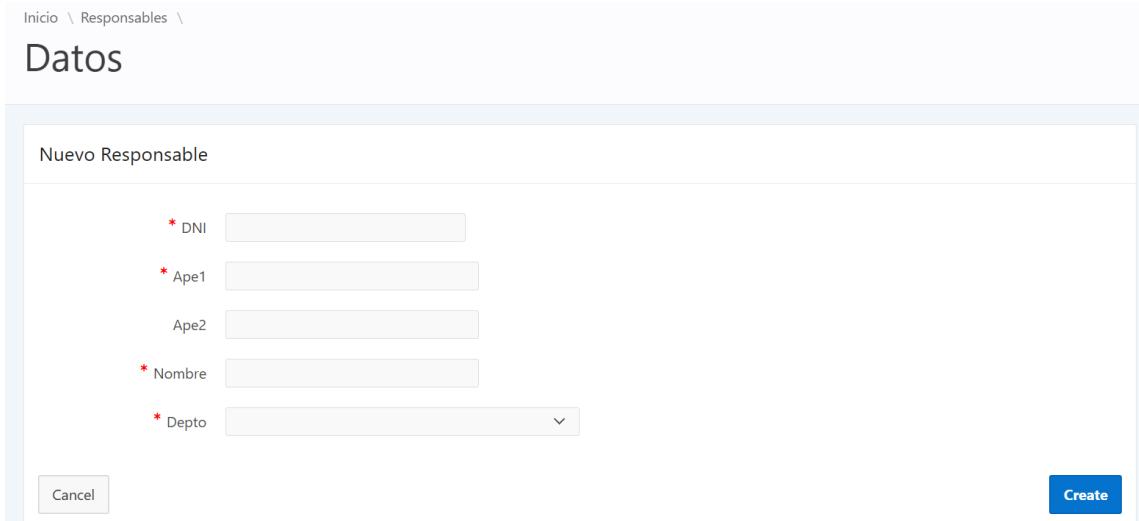
## Responsables

En esta tabla podemos ver y gestionar los Responsables de los proyectos, indicar que todos estos datos, han sido generados aleatoriamente y no corresponden a la realidad. Como vemos en la imagen, podremos Buscar por campos, (pulsando en el icono de la lupa) , editar los datos (pulsando en el icono del lápiz) , crear, agrupaciones, filtros, informes y descargarlos, pulsando en el botón **Actions**.



	Dni	1er Apellido	2º Apellido	Nombre	Departamento
	00000029A	Torres	Ramírez	Carlos	FISICA
	00000026A	López	Sánchez	Lucía	FISICA
	00000042A	Díaz	Vega	Lucía	FISICA

Si lo que queremos es crear un nuevo responsable pulsaremos en la opción **Create** de la derecha, que nos mostrará el siguiente menú, donde podremos incluir los datos necesarios, DNI, Apellidos, Nombre y Departamento (el cual podremos escoger en la cortinilla desplegable)

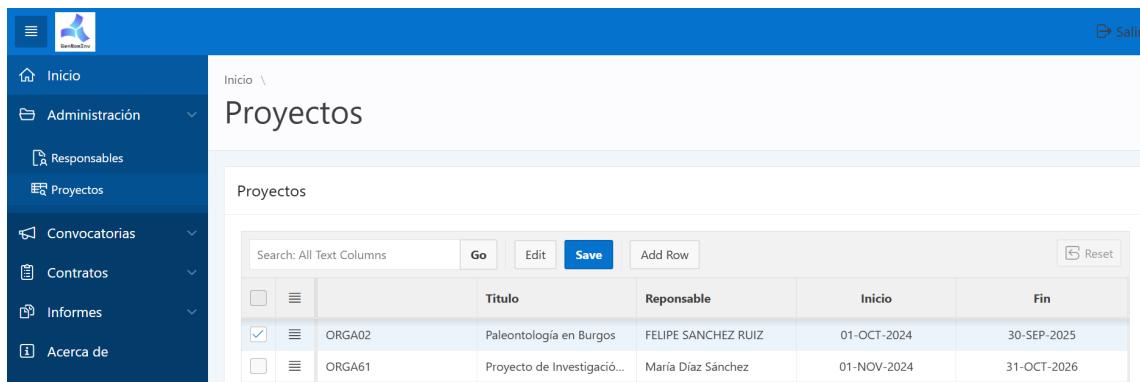


Nuevo Responsable

* DNI	<input type="text"/>
* Ape1	<input type="text"/>
Ape2	<input type="text"/>
* Nombre	<input type="text"/>
* Depto	<input type="text"/>

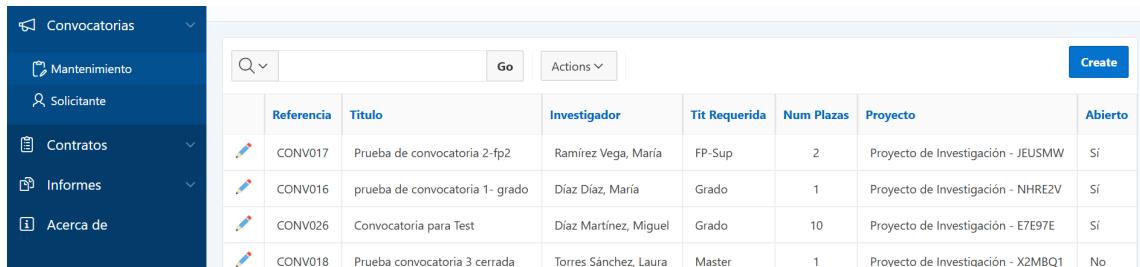
## Proyectos

Para los proyectos se ha creado un Grid Interactivo, que muestra los proyectos existentes (éstos también han sido generados de forma aleatoria), realizar búsquedas por Orgánica, Titulo y Responsable, editar campos, añadir nuevos proyectos (**Add Row**) y desde el botón **Actions**, generar filtros e informes.



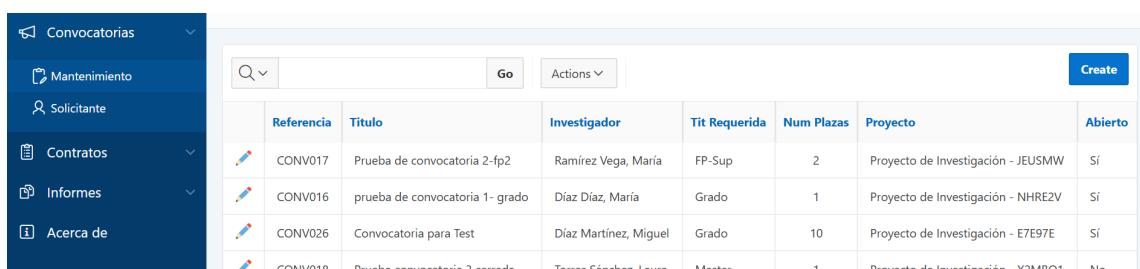
		Titulo	Responsable	Inicio	Fin
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ORGANIZACIÓN DE LA INVESTIGACIÓN EN BURGOS	FELIPE SÁNCHEZ RUIZ	01-OCT-2024	30-SEP-2025
<input type="checkbox"/>	<input type="checkbox"/>	Proyecto de Investigación	María Díaz Sánchez	01-NOV-2024	31-OCT-2026

## E.5. Convocatorias



	Referencia	Título	Investigador	Tit Requerida	Num Plazas	Proyecto	Abierto
<input type="checkbox"/>	CONV017	Prueba de convocatoria 2-fp2	Ramírez Vega, María	FP-Sup	2	Proyecto de Investigación - JEUSMW	Sí
<input type="checkbox"/>	CONV016	prueba de convocatoria 1- grado	Díaz Díaz, María	Grado	1	Proyecto de Investigación - NHRE2V	Sí
<input type="checkbox"/>	CONV026	Convocatoria para Test	Díaz Martínez, Miguel	Grado	10	Proyecto de Investigación - E7E97E	Sí
<input type="checkbox"/>	CONV018	Prueba convocatoria 3 cerrada	Torres Sánchez, Laura	Master	1	Proyecto de Investigación - X2MBQ1	No

Desde el menú convocatorias, podremos realizar el **Mantenimiento** de las convocatorias que solicite un Responsable de un Proyecto, inicialmente nos mostrará todas las que tenemos pudiendo editarlas, pulsando el icono del lapicero



	Referencia	Título	Investigador	Tit Requerida	Num Plazas	Proyecto	Abierto
<input type="checkbox"/>	CONV017	Prueba de convocatoria 2-fp2	Ramírez Vega, María	FP-Sup	2	Proyecto de Investigación - JEUSMW	Sí
<input type="checkbox"/>	CONV016	prueba de convocatoria 1- grado	Díaz Díaz, María	Grado	1	Proyecto de Investigación - NHRE2V	Sí
<input type="checkbox"/>	CONV026	Convocatoria para Test	Díaz Martínez, Miguel	Grado	10	Proyecto de Investigación - E7E97E	Sí
<input type="checkbox"/>	CONV018	Prueba convocatoria 3 cerrada	Torres Sánchez, Laura	Master	1	Proyecto de Investigación - X2MBQ1	No

También podremos hacer búsquedas por los distintos campos, pulsando como en anteriores ocasiones en la lupa y las mismas acciones.

Para generar una nueva convocatoria pulsaremos en el botón **Create**:

Tendremos que llenar los campos que se indican obligatoriamente con un \*

En el desplegable **Tit Requerida**, podremos seleccionar una de las titulaciones mínima que se requerirán para poder optar, por los solicitantes a cada convocatoria.

En el desplegable **Investigador**, podremos seleccionar cualquiera de los investigadores que estén dados de alta en el sistema, como hemos visto anteriormente en la pantalla **Responsables**. Este punto es importante, ya que en siguiente desplegable **Ref. Proy**, solo nos mostrará los proyectos del investigador seleccionado, indicándonos después el tiempo que queda para la finalización de este proyecto.

Al confirmar con **OK** podremos pulsar en el botón **Create**, generando una nueva convocatoria.

Nueva Convocatoria	
* Titulo	prueba
* Tit Requerida	Diplomado
Num Plazas	1
Abierto	Sí
* Investigador	Díaz Díaz, M.
* Ref Proy	Proyecto de Investigación - NHRE2V

**AVISO:** La duración del proyecto es: 3 años, 5 meses, 4 días restantes

**OK**

## Solicitante

Esta pantalla nos mostrará los solicitantes existentes para las convocatorias creadas, pudiendo realizar las mismas operaciones que anteriormente (búsquedas por campos y diversas acciones)

También podremos dar de alta un nuevo solicitante de una convocatoria abierta, pulsando en el botón **Create**.

Inicio \ Solicitante \

## Datos

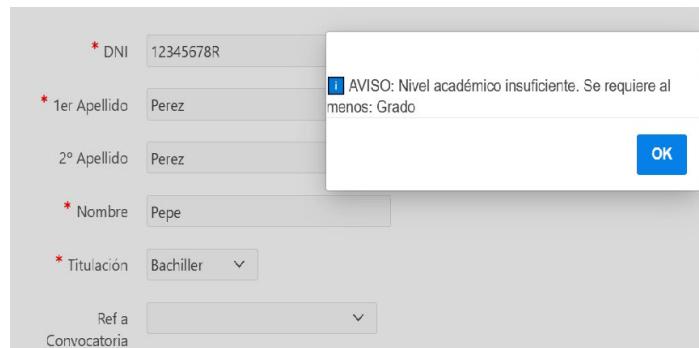
Nuevo Solicitud

* DNI	12345678R
* 1er Apellido	Perez
2º Apellido	Perez
* Nombre	Pepe
* Titulación	Bachiller
Ref a Convocatoria	<div style="border: 1px solid #ccc; padding: 5px; width: 150px;">           ▾            Convocatoria para Test            Prueba convocatoria 3 cerrada            Prueba de convocatoria 2-fp2         </div>
<input type="button" value="Cancel"/>	<input type="button" value="Create"/>

Tendremos que llenar los campos que se indican y seleccionar en el desplegable **Titulación**, una de las opciones que se indican correspondiente al nivel académico del solicitante. Luego escogeremos la convocatoria en la que quiere participar, y al seleccionarla, el sistema nos informará si puede o no participar, según el nivel requerido para la misma. En este caso con un nivel de Bachiller, en la convocatoria para Test, nos avisa de la incidencia.

* DNI	12345678R
* 1er Apellido	Perez
2º Apellido	Perez
* Nombre	Pepe
* Titulación	Bachiller
Ref a Convocatoria	<div style="border: 1px solid #ccc; padding: 5px; width: 150px;">           ▾            AVISO: Nivel académico insuficiente. Se requiere al menos: Grado         </div>

También, nos indicará si la convocatoria ya está cerrada, es el caso de si escogemos la opción de convocatoria cerrada.



Si la convocatoria fuera adecuada a su titulación pulsaríamos el botón **Create**

## E.6. Contratos

### Crear

Dentro de esta opción podremos **Crear**, un contrato para un **Solicitante**, vinculado a una **Convocatoria**

Ref Contrato	Convocatoria	Contratado	Tipo	F Ini	F Fin	Retribución Total	Mensualidad	Indemnización	SS Mes	Reserva
CONT086	Prueba de convocatoria 2-fp2	María Martínez Sánchez	PARCIAL	01-OCT-2025	31-DIC-2025	€4.700,00	€1.000,00	€800,00	€300,00	€4.700,00
CONT141	Convocatoria para Test	Pedro Picapiedra Dura	INDEFINIDO	01-AGO-2025	31-DIC-2025	€15.640,00	€1.200,00	€1.450,00	€120,00	€15.640,00

Inicialmente vemos los contratos dados de alta, pudiendo realizar las distintas tareas como en anteriores ocasiones, tales como, búsquedas por campos y diversos filtros e informes. Además, pulsando en el icono de edición podremos modificar datos de un contrato.

Para generar un nuevo contrato, pulsaremos en el botón **Create**

Inicialmente nos solicitará para qué convocatoria queremos el contrato, mostrando aquellos solicitantes de esa convocatoria.

### Nuevo Contratado

\* Convocatoria: Convocatoria para Test

\* Contratado: Juncal Álvarez Leal

\* Tipo: Juncal Álvarez Leal

\* Fecha Inicio: Test Solicitante de Prueba

Luego nos irá solicitando los diversos datos para realizar el contrato.

En el desplegable **Tipo**, podremos escoger entre los distintos tipos de contrato en vigor.

Después deberemos introducir las fechas de inicio y fin, las cuales deberán estar comprendidas entre las fechas del **Proyecto** vinculado, lanzando sendos avisos si no se cumpliera este requisito:



Una vez introducidos los datos requeridos y pulsando en el botón **Create**, nos pedirá confirmación para la creación del nuevo contrato y de la generación las nóminas correspondientes en la tabla nómina automáticamente para los meses correspondientes y uno adicional de seguridad social, que no tiene retribución.

TGFUBU.NOMINA							
Columnas	Insertar	Suprimir selección	✓ Confirmación	Deshacer todo			
Datos		DNI_NOM	MES	ANNO	MENSUALIDAD	SEG_SOC	OBSERVACIONES
Restricciones	1	12345678J		10	2025	1000	0 (nulo)
Permisos	2	12345678J		11	2025	1000	300 (nulo)
Estadísticas	3	12345678J		12	2025	1000	300 (nulo)
Disparadores	4	12345678J		1	2026	800	300 indemnización
Dependencias	5	12345678M		1	2026	0	120 ultima ss
Detalles	6	12345678M		8	2025	1200	0 (nulo)
Particiones	7	12345678M		9	2025	1200	120 (nulo)
índices	8	12345678M		10	2025	1200	120 (nulo)
	9	12345678M		11	2025	1200	120 (nulo)
	10	12345678M		12	2025	1450	120 Último mes con inder
	11	12345678H		11	2025	1500	0 (nulo)
	12	12345678H		12	2025	2100.25	500 Último mes con inder
	13	12345678H		1	2026	0	500 ultima ss
	14	12345678I		7	2025	2100.25	0 Último mes con inder
	15	12345678I		8	2025	0	525 ultima ss

## Renovación

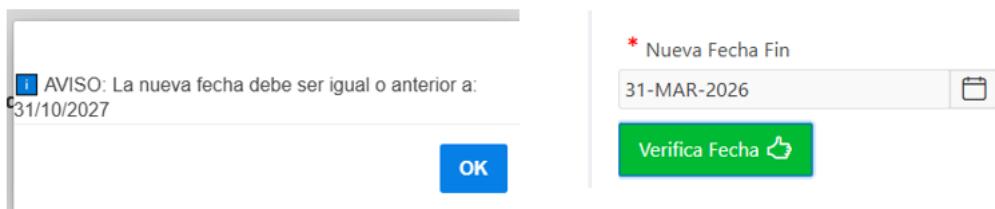
En este apartado podremos realizar la renovación de un contrato, siempre y cuando no sobrepase la fecha fin del **Proyecto**, vinculado.

Así, al seleccionar un contrato, nos ofrecerá los datos del mismo en la parte izquierda:

Pudiendo introducir los datos en la parte derecha.

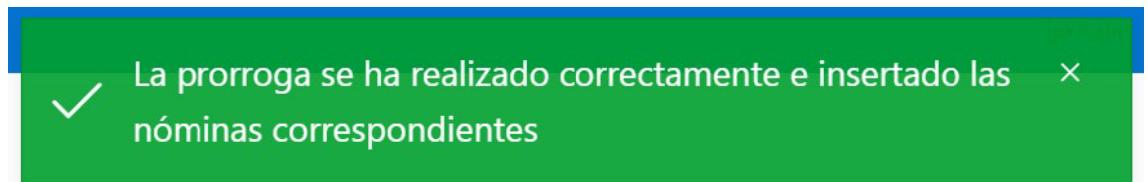
Para la fecha de renovación deberemos verificar la misma, pulsando en el botón **Verifica Fecha**, indicándonos si por ejemplo la fecha es anterior a la actual:

Si la fecha excede del fin del contrato o si la fecha fuese correcta, el botón cambiará a color verde.



Una vez rellenos los datos correspondientes y pulsado el botón **Renovar**, nos solicitará confirmación, como en el caso de creación inicial y generará las nuevas nóminas correspondientes y cambiará la fecha fin del mismo.

Realizando la confirmación:



## Renuncia

En esta sección un solicitante podrá renunciar a un contrato en una fecha determinada.

Como en el caso de la **Renovación**, tendremos que seleccionar un contrato:

### Renuncia

The screenshot shows a modal window titled "Renuncia". On the left, under "Datos del contrato:", a dropdown menu is open showing "CONT141 - Pedro Picapiedra Dura". Below it, the following details are listed: REFERENCIA: CONT141, NOMBRE: Pedro Picapiedra Dura, INICIO: 01-08-2025, FIN: 31-03-2026, MENSUALIDAD: 1.200,00 €, SEG.SOCIAL: 125,00 €, and INDEMNIZACIÓN: 1.500,00 €. On the right, under "Datos renovación:", there is a "Renunciar" button. Below it, fields for "Fecha de Renuncia" (with a calendar icon) and "Indemnización Total" (with a value of 0) are shown, along with a "Verifica Fecha" button.

El sistema nos mostrará los datos del contrato y la fecha fin actual, debiendo introducir en la parte de la derecha los datos de renuncia, y como en el caso anterior, comprobando si la renuncia es posterior a la fecha actual.

AVISO: La fecha de renuncia debe ser posterior a la fecha actual

OK

Y si es anterior a la fecha fin actual.

AVISO: La fecha de renuncia debe ser posterior a la fecha actual

OK

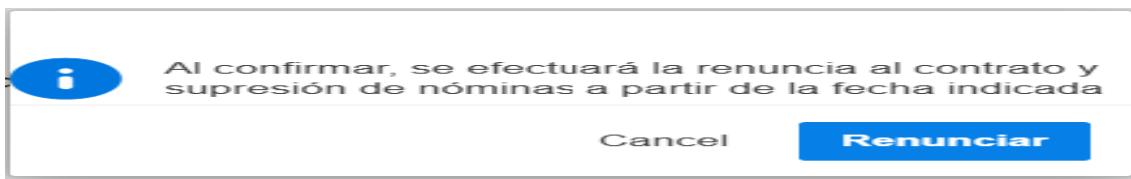
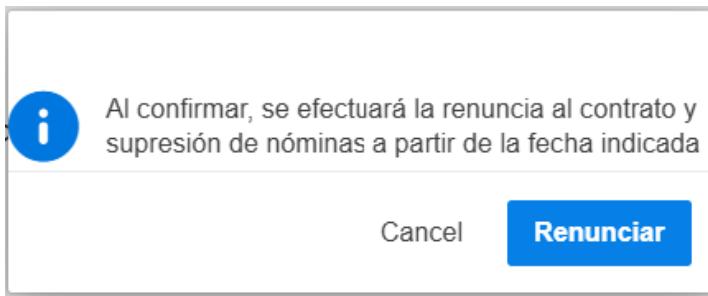
En el caso de haber introducido una fecha de renovación correcta, se verifica, con el botón en verde y al completar el resto de datos podemos pulsar en el botón **Renunciar**

Datos renovación:

\* Fecha de Renuncia  
31-ENE-2026

\* Indemnización Total  
1600

En este momento se nos pedirá confirmación para realizar la acción, que eliminará las nóminas correspondientes y cambiará la fecha fin del contrato.



## E.7. Informes

### Nómina-mes

Con este informe podremos ver la nómina correspondiente al mes seleccionado, es el informe principal de la aplicación, ya que se compara con la

sábana de retribuciones y se coteja con los datos obrantes en el servicio. Para ello accedemos a la sección de informes Nómina-mes y muestra la pantalla siguiente, en la que podemos escoger mes y año y al pulsar en Consultar obtendremos las nóminas a pagar ese mes.

Inicio \

## Informe Nomina Mes

Mes:  Año:  Consultar

Enero  
Tot  
Febrero  
Marzo  
Se Abril  
Mayo

Total General:

Una vez mostrados los datos, podremos filtrar por orgánica y volver a pulsar Consultar para aplicar ese filtro.

Inicio \

## Informe Nomina Mes

Mes:  Año:  Consultar

Enero  
Tot  
Febrero  
Marzo  
Se Abril  
Mayo

Total General:

Una vez conforme la vista pulsando en el botón generar, nos descargará un informe personalizado en pdf a través del servicio online Apex Oficce Print (AOP).



Genomin: 02/08/2025 9:08:12

#### **INFORME NÓMINA: (11 / 2025)**

ORGANICA	NOMBRE	MENSUALIDAD	SEC-SOCIAL	TOTAL	OBSERVACIONES
ORGA33	Maria Martínez Sánchez	1000	300	1300	
ORGA55	Pedro Picapiedra Dura	1200	120	1320	
ORGA55	Juncal Álvarez Leal	1500	0	1500	
<b>TOTAL GENERAL</b>				<b>4120</b>	

## Vencimientos

Desde aquí podremos consultar los vencimientos de los contratos, para ello introduciremos la fecha inicio y fin y pulsaremos el botón Consulta:

Inicio \ Vencimientos

Intervalo de Fechas

Desde: 01-AGO-2025      Hasta: 31-AGO-2025

Q Go Actions ▾      Consulta

Que nos mostrará los contratos que vencen entre las fechas indicadas:

Intervalo de Fechas

Desde: 01-ENE-2025      Hasta: 31-DIC-2025

Q Go Actions ▾      Consulta

Contratado	Contrato	Organica	Fin
Martínez Sánchez, María	CONT086	ORGA33	31-DIC-2025
Álvarez Leal, Juncal	CONT122	ORGA55	31-DIC-2025
Pérez López, Luis	CONT123	ORGA21	30-JUL-2025

Pulsando en la lupa podremos filtrar por campos y en el botón Actions, nos permite realizar varias opciones; como columnas a mostrar, filtros avanzados, ordenar datos y operar, guardar este informe para otras ocasiones (si lo hemos modificado) y descargar el informe en varios formatos

## Contratos

Este informe ofrece un listado de todos los datos de los contratos que hay en vigor

Intervalo de Fechas			
Desde:	01-ENE-2025	Hasta:	31-DIC-2025
<input type="button" value="Q v"/> <input type="button" value="Go"/> <input type="button" value="Actions v"/>		<input type="button" value="Consulta"/>	
Contratado	Contrato	Organica	Fin
Martínez Sánchez, María	CONT086	ORGA33	31-DIC-2025
Álvarez Leal, Juncal	CONT122	ORGA55	31-DIC-2025
Pérez López, Luis	CONT123	ORGA21	30-JUL-2025

1 - 3

Como en el anterior informe se pueden filtrar campos con la lupa y acceder a otras opciones y filtros avanzados con el botón Actions.

## Nominas-Periodo

Este informe presenta las nóminas que tiene un contratado en un periodo que se indica, una vez accedido al informe tendremos que seleccionar el contratado y el periodo desde, hasta.

### Vencimientos

Intervalo de Fechas			
* Desde:	<input type="text"/>	* Hasta:	<input type="text"/>
<input type="button" value="Q v"/> <input type="button" value="Go"/> <input type="button" value="Actions v"/>		<input type="button" value="Consulta"/>	
<hr/>			

Una vez pulsado el botón Buscar, se nos muestran las nóminas y el total de ese periodo. Este informe es usado para cotejar el importe pagado en un periodo a un contratado.

Selección Contratado					
Nombre:	Desde:		Hasta:		
Juncal Álvarez Leal	01-ENE-2025	<input type="button" value=""/>	31-DIC-2026	<input type="button" value=""/>	
Mes	Año	Mensualidad	Seg Soc	Total	Observaciones
1	2026	0	500	500	última ss
11	2025	1500	0	1500	
12	2025	2100,25	500	2600,25	Último mes con indemnización
				<b>4600,25</b>	

## E.8. Acerca de

Se muestran datos de la aplicación:

Acerca De



UNIVERSIDAD DE BURGOS  
ESCUELA POLÍTÉCNICA SUPERIOR  
Grado en Ingeniería Informática

TFG GeNomIn: Gestión de Nóminas de Investigación

Autor: Francisco J. Arroyo Redondo.

Tutor: Pedro Renedo Martínez

Burgos, septiembre 2025



## Guía de uso

Inicio \ Acerca de \

User Guide

Manual de Usuario





## *Apéndice F*

---

# **Competencias de Sostenibilidad Curricular**

---

Como ya se expuso brevemente en el capítulo 3.8 de la memoria, durante el desarrollo de este trabajo y la aplicación **GeNomIn**, que ha sido creada para modernizar un proceso administrativo tedioso. Desarrollada en **APEX** y desplegada en **OCI**, se ha intentado integrar los *Objetivos de Desarrollo Sostenible (ODS)* en el ámbito tecnológico [4].

Este proyecto me ha permitido comprender y aplicar estas competencias en sostenibilidad curricular, que se alinean con el **uso responsable de recursos**, la **conciencia ambiental**, la **participación comunitaria** y los **principios éticos**.

### **F.1. Competencias Adquiridas**

Este **TFG**, ha permitido reflexionar sobre la transformación que está sufriendo la sociedad, en particular las administraciones, influidas por la tecnología. En particular vemos como la digitalización reduce el impacto ambiental, mejora el acceso a la información y es mucho más eficiente que el mero uso del papel.

#### **Uso sostenible de recursos**

Según los datos aportados por (**Oracle y sostenibilidad**) sus centros utilizan energía renovable y diseño circular. Además, la eliminación de procesos manuales y en papel, supone una reducción de un 90 % de éstos, haciendo el proceso más sostenible en un entorno en la nube.

### **Participación comunitaria**

En el desarrollo del proyecto, como no podía ser de otra forma, se involucró al personal del Servicio para un mejor control del proceso administrativo seguido en el proyecto. Esto promueve una transformación digital transparente dentro de las instituciones.

### **Principios éticos**

El uso, por parte de Oracle, de herramientas **Ambientales, Sociales y de Gobernanza (ESG)**, refuerza el compromiso ético con la sostenibilidad, alineado con los objetivos **ODS, 12: (producción y consumo responsables), 13: Acción por el clima y 16: Paz, justicia e instituciones sólidas**

## **F.2. Aplicación en el Proyecto GeNomIn**

### **Diseño y Funcionalidad**

**GeNomIn** ha sido diseñada para sustituir hojas de cálculo en almacenamiento compartido por una solución segura, escalable e integral. Esta automatización puede reducir los errores hasta en un 70 % y un ahorro temporal de más del 30 %

### **Impacto Ambiental y Social**

La digitalización de procesos administrativos es uno de los objetivos **ODS** (9), que complementa al objetivo (13), **acción climática**, y a las **instituciones sólidas** (16). Así, esta migración a la nube supone, con servicios optimizados con gran eficiencia energética, un impacto ambiental mínimo.

### **Conciencia ambiental**

Durante la realización del proyecto se reflexionó sobre la ideoneidad de la migración a un entorno en la nube como Oracle y el consumo de sus servidores. Pero, evidentemente, el impacto ambiental es mucho mayor en el entorno de la propia Universidad que no dispone de los recursos de Oracle, aunque tampoco se ha podido valorar esta diferencia.

## F.3. Conclusión

Lo cierto es que antes de la iniciación de este **TFG**, no se había reflexionado sobre el impacto que puede tener la tecnología en el medio ambiente. Todos oímos hablar del coste (económico-ambiental) de los grandes proveedores de Google, pero no caemos en la cuenta de nuestro propio trabajo. Analizando el uso anterior, a través de hojas de cálculo almacenamientos, impresiones en papel y procesos obsoletos, nos damos cuenta de la importancia de adquirir estas competencias, no solo en el ámbito de la informática, si no, en la vida cotidiana.

**GeNomIn** es solo una gota de agua en la transformación de la administración hacia procesos más modernos y eficientes, pero me ha servido de aprendizaje para comprender el impacto que tienen las decisiones técnicas que se alinean con los **ODS** y la agenda 2030.



---

# Siglas

---

**AOP** APEX Office Print. 5, 8, 55, 59, 70

**APEX** *Application Express.* 2, 3, 5, 33, 34, 37, 39, 44, 59, 62, 65, 68, 69, 70, 89

**BDR** *Base de Datos Relacional.* 59

**ESG** Ambientales, Sociales y de Gobernanza. 90

**FK** *Foreing Key-Clave Externa.* 3

**HTTPS** *Secure Hipertext Transfer Protocol.* 4, 61, 63, 64

**JS** JavaScript. 31, 34, 36, 38, 40, 44, 49, 54

**MPV** *Mínimo Producto Viable.* 3

**OCI** Oracle Cloud Infrastructure. 5, 63, 67, 89

**ODB** *ORACLE 23ai.* 33

**ODS** *Objetivos de Desarrollo Sostenible.* 5, 89, 90, 91

**ORDS** Oracle REST Data Services. 59, 61, 64

**PBD** Plugable Data Base. 59

**PK** *Primary Key-Clave Primaria.* 3

**PL/SQL** Procedural Language/Structured Query Language. 34, 41, 46, 52

**RGPD** Reglamento General de Protección de Datos. [8](#)

**SQL** *Structured Query Language.* [34](#), [35](#), [37](#), [38](#), [39](#), [40](#), [43](#), [49](#), [54](#), [55](#), [56](#), [61](#)

**SSL** Secure Sockets Layer. [5](#)

**TFG** *Trabajo de Fin de Grado.* [6](#), [8](#), [89](#), [91](#)

---

## Glosario

---

**Datos MOCK** Datos ficticios para probar aplicaciones, bases de datos, interfaces, sin necesidad de usar datos reales o confidenciales. [4](#)

**Kanban** Método de gestión visual del flujo de trabajo que se caracteriza por el uso de tableros visuales y tarjetas para representar las tareas y su progreso a través de diferentes etapas, promoviendo la transparencia y la mejora continua.. [1](#), [1](#), [2](#)

**LATEX** Sistema de composición de textos orientado a la creación de documentos escritos, generalmente científicos que presenten una alta calidad tipográfica. [5](#)

**Metodología ágil** Enfoque iterativo e incremental para la gestión de proyectos, especialmente popular en el desarrollo de software, que se centra en la flexibilidad, la colaboración y la entrega continua de valor. En lugar de adherirse rígidamente a un plan predefinido, agile abraza el cambio y la adaptación a medida que avanza el proyecto.. [1](#)

**Milestone** Punto de control o evento significativo dentro de un proyecto que marca la finalización de una fase importante o la consecución de un objetivo específico. [1](#), [1](#), [2](#), [6](#)

**Normalización** Proceso de estructuración los datos en una BD relacional, eliminando redundancia de datos y manteniendo la integridad a través de Formas Normales. [3](#)

**Release** Versión específica de un software que ha sido completada, probada y puesta a disposición para su uso por los usuarios finales.. [2](#), [3](#)

**Representational State Transfer** Estilo de arquitectura para diseñar servicios web que se comunican a través de HTTP. [61](#)

**Scrum** Marco de trabajo ágil diseñado para gestionar y completar proyectos de manera eficiente y colaborativa.. [1](#)

**Sprint** Período de tiempo corto y fijo (generalmente de una a cuatro semanas) durante el cual un equipo trabaja para completar un conjunto específico de tareas.. [1](#), [2](#), [3](#), [4](#), [6](#)

---

## Bibliografía

---

- [1] Oracle APEX Administration Guide.
- [2] Overview | APEX Office Print (AOP).
- [3] Santanu Datta, Malai Stalin, Gordon Larimer, Sudeepthi Siddavatam, Gerald Venzl, Christopher Jones, Sivaselvam Narayanasamy, Sritej Puvvada, Vasundhara Venkatasubramanian, Alex Keh, Subrahmanyam Kodavaluru, and Sunil Surabhi. Installing Oracle Database Free.
- [4] Urtzi Markiegi, Inigo Aldalur, and Alain Pérez. Integrando los ODS en el grado de Ingeniería Informática.