



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



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Pedro Renedo Fernández, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Francisco J. Arroyo Redondo, con DNI 13130859, ha realizado el Trabajo final de Grado en Ingeniería Informática.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 14 de agosto de 2025

Vº. Bº. del Tutor:

D. Pedro Renedo Fernández

Resumen

Gestión de Contratos GeNomIn, es una aplicación web desarrollada en *Application Express (APEX)* 24.2 para el control y gestión de los pagos de contratos de Investigación que actualmente se gestionan a través de una hoja de cálculo. Estos contratos se vinculan a convocatorias y éstas a proyectos de investigación, pretendiendoificar todas estas actividades en un único aplicativo más sencillo de gestionar por el personal del Servicio y que permita a través de un informe de nómina mensual cotejar los datos remitidos por retribuciones para ese mismo mes, con los calculados por el Servicio.

Descriptores

Gestión de Pagos, Contratos, Convocatorias, Investigación, Oracle, Apex, GeNomIn

Abstract

GeNomIn Contract Management is a web application developed in APEX 24.2 for the control and management of research contract payments, currently managed through a spreadsheet. These contracts are linked to calls for proposals, and these are linked to research projects. The aim is to integrate all these activities into a single application that is easier for Service staff to manage and that allows, through a monthly payroll report, to compare the data submitted for remuneration for that same month with the data calculated by the Service.

Keywords

Payment Management, Contracts, Calls, Research, Oracle, Apex, GeNomIn

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Índice de Códigos	vii
1. Introducción	1
2. Objetivos del proyecto	3
2.1. Objetivos funcionales	3
2.2. Objetivos no funcionales	4
2.3. Objetivos personales	5
3. Conceptos teóricos	7
3.1. Casos de Uso	7
3.2. Diagramas de Secuencias	8
3.3. Normalización	9
3.4. Modelo E-R	10
3.5. Diseño Físico	10
3.6. Sistemas de Gestión de Bases de Datos (SGBD)	11
3.7. Agilidad-Scrum	13
3.8. Objetivos de Desarrollo Sostenible ODS	16
4. Técnicas y herramientas	19
4.1. Oracle23ai	20
4.2. Apex 24.02	21

4.3. ORDS	22
4.4. TestCafe	22
4.5. Oracle Cloud Free Tier	25
4.6. Apex Office Print	26
4.7. GitHub	26
4.8. Zube	28
4.9. TextStudio	28
5. Aspectos relevantes del desarrollo del proyecto	29
5.1. Oportunidad de negocio	29
5.2. ¿Por qué APEX 2402?	30
5.3. Creación de las Páginas	31
5.4. Programación	31
5.5. Despliegue en Oracle Cloud Free Tier	32
6. Trabajos relacionados	33
7. Conclusiones, valoración personal y trabajo futuro	35
7.1. Cumplimiento de objetivos funcionales	35
7.2. Cumplimiento de objetivos no funcionales	36
7.3. Valoración Personal	36
7.4. Trabajo futuro	36
7.5. Reflexión sobre el uso de IA	37
Bibliografía	43

Índice de figuras

3.1. Casos de Uso	8
3.2. Diagrama de secuencias	8
3.3. Detalle anualidad de un contrato	9
3.4. Diagrama de Entidad Relación	11
3.5. Kanban TFGUBU	14
3.6. Informe de test Allure	14
4.1. Esquema de Oracle23ai	20
4.2. Home Apex TFGUBU-GeNomIn	21
4.3. Configuración ORDS-HTTPS	22
4.4. Resultado del Test 25:comprueba el informe entre fechas	24
4.5. Base de Datos Autónoma en Oracle Cloud	25
4.6. Plantilla de informes-Informe AOP	26
4.7. GitHub del Proyecto GeNomIn	27
4.8. Vista de un release	28
5.1. Gestión de usuarios de APEX	30
5.2. Vista de las páginas APEX	31
5.3. Acciones dinámicas APEX y su código js	31
7.1. Logotipo de la aplicación: GeNomIn	35

Índice de tablas

3.1. <i>Objetivos de Desarrollo Sostenible (ODS)</i>	16
6.1. <i>Comparativa GeNomIn vs Universitas XXI</i>	33

Índice de Códigos

3.1. Generación de <i>Datos MOCK</i> para la tabla Responsables . . .	12
4.2. Código en js para prueba en TestCafe	23

1. Introducción

El Servicio de Gestión de la Investigación realiza diversas tareas en el ámbito universitario que van desde la tramitación de facturas, ayudas a la investigación, doctorado, gestión de proyectos, becas y contratos de investigación, pudiendo considerarse en sí misma una **mini-gestora** universitaria.

En este contexto cobran una gran trascendencia los contratos de investigación, ya que generan un gran volumen de trabajo e información. Así, para la realización de un contrato de investigación asociado a un proyecto, entran en liza diferentes ámbitos del servicio que pueden llegar a ser inmanejables tal y como se realiza actualmente.

Se tramitan aproximadamente 400 contratos de investigación, contabilizando su detalle a través de una hoja de cálculo, que contiene en sus campos los principales datos, junto con las retribuciones mensuales, tales como: orgánica, datos, contratado y cada una de las mensualidades del año, con su seguridad social.

Antes de finalizar el mes, se cotejan los pagos apuntados en esta tabla excel, con los datos remitidos por el Servicio de Retribuciones, lo cual, como es de suponer, genera no pocos errores.

Se pretende con este trabajo de fin de Grado, ya no sólo facilitar el que hacer diario, si no evitar errores que se producen fácilmente con el manejo tan voluminoso de hojas de cálculo.

Dicha migración se realizará a través de una aplicación en *Application Express (APEX)* 24.2 (sobre Oracle 23ie) con la que está familiarizada el personal de este servicio, ya que actualmente se ha migrado a esta plataforma, tanto el portal de investigación con la gestión económica.

2. Objetivos del proyecto

2.1. Objetivos funcionales

Estos objetivos se centran en las funcionalidades y características que debe tener la aplicación *GeNomIn* para satisfacer las necesidades y expectativas de los usuarios. A continuación se detallan los objetivos funcionales del proyecto:

- **Sustitución de la hoja de cálculo Excel:** Uno de los primeros y fundamentales pasos es la conversión de el libro de Excel tablas para la *ORACLE 23ai (ODB)* sin pérdida de datos a través del proceso de *Normalización*. En este proceso se irán generado *Forma Normal (FN)* 1, 2 y 3 que nos permitirán; la 1ª eliminar grupos repetidos de datos, y que cada columna tenga un solo valor, la 2ª eliminará redundancias en claves primarias y la 3ª eliminará dependencias transitivas.
- **Creación de las tablas en la *ORACLE 23ai (ODB)*:** Una vez normalizada el libro de Excel, se crearán las correspondientes tablas en la *ODB*, configurando correctamente las relaciones entre las mismas *Modelo Relacional*, de vital importancia, ya que una mala definición de las relaciones entre las distintas tablas puede dar al traste con todo el proyecto.
- **Creación de la aplicación sobre *Application Express (APEX)*** 24.2: Uno de los principales retos es la creación de dicha aplicación, su correcto funcionamiento y funcionalidad, que exige el aprendizaje de esta potente plataforma de desarrollo de aplicaciones web de bajo código que se ejecuta dentro de una base de datos Oracle. Esta versión

introduce nuevas funcionalidades, mejoras y actualizaciones, especialmente en el área de la inteligencia artificial generativa y la experiencia del desarrollador.

- **Despliegue de la aplicación:** Inicialmente se contemplaron las opciones de compartir el trabajo a través de **VirtualBox** o **Kubernetes**, pero finalmente se escogió la opción de **Oracle Cloud Infrastructure (OCI)** por su aproximación a un despliegue y uso real, ya que ofrece prácticamente todas las opciones en su version **Free Tier**, tanto para el alojamiento de la **BDR**, como de la aplicación en **Application Express (APEX)**

2.2. Objetivos no funcionales

Los objetivos no funcionales se refieren a los desafíos y metas que se deben abordar para desarrollar el software. Estos objetivos abarcan aspectos como la arquitectura del sistema, las tecnologías a utilizar y las metodologías de desarrollo. A continuación se detallan los objetivos no funcionales del proyecto:

- **Gestión de las diversas herramientas para la creación de la aplicación GeNomIn:** Durante la asignatura de *Sistemas de Gestión de BD (SGBD)*, se han visto someramente las herramientas **APEX** 5.1 y Oracle 9. Teniendo el objetivo de desarrollar una aplicación funcional y entendiendo que ha de desplegarse en un servidor **HTTPS** con **Secure Sockets Layer**, se prefirió como reto personal, el desarrollo sobre **Application Express (APEX)** 23.2 sobre Oracle 23ai. También durante la asignatura de *Validación y Pruebas (VYP)*, se han utilizado herramientas de prueba como Selenium y Katalom, se ha querido salir de la zona de confort y utilizar **textcafe** para comprobar la usabilidad de la aplicación.
- **Usabilidad de GeNomIn:** La aplicación no debe ser sólo útil, como principal objetivo, si no que debe permitir un manejo fácil e intuitivo por el personal que debe manejarla. Como se ha comentado en la introducción la familiaridad del personal con **Application Express (APEX)** en las distintas aplicaciones ya usadas (portales Económico e Investigación), favorece una mejor asimilación de este nuevo entorno en comparación con la tediosa hoja de cálculo.

2.3. Objetivos personales

- **Capacidad de desarrollo:** el principal objetivo es comprobar si el esfuerzo realizado durante todos estos años es aplicable a la "vida profesional" tiene aplicación directa en un caso real.
- **Reto de aprendizaje:** Lo sencillo hubiera sido realizar este trabajo con herramientas conocidas y se ha querido llevar un paso más allá, utilizando versiones muy superiores a las utilizadas en prácticas (**APEX5.1**) y software totalmente distinto para la ejecución de pruebas y la realización de un despliegue en un entorno de trabajo profesional como **OCI**. La realización de esta memoria con **LATEX**, también supone un cambio de paradigma sobre el uso de los procesadores de texto convencionales como Word.
- **Reto personal:** la realización del **TFG** supone el final del camino iniciado hace ya más de 20 años cuando inicié la Ingeniería Técnica Informática y por motivos personales no pude concluir.

3. Conceptos teóricos

En este capítulo se definen los conceptos teóricos que se ha utilizado para la conversión de un hoja de cálculo Excel en una *Base de Datos Relacional (BDR)*, y la consecución de la aplicación web **GeNomIn**.

3.1. Casos de Uso

La parte fundamental a la hora de desarrollar una aplicación es la entrevista con el usuario, de nada sirve una aplicación visualmente perfecta si no cumple con la función que requiere el usuario final. Los *Caso de Uso* sirven para modelar el sistema, entender las funcionalidades y establecer los requisitos que luego serán testados.

Los **actores** identificados en el análisis previo corresponden a uno principal, el **Gestor**, usuario final que realiza las acciones en la aplicación, y cuyo objetivo es tener una serie de informes de los contratos de nómina, para poder controlar los pagos, y otros dos "secundarios" **Investigador**, que realiza la solicitud de una convocatoria y el **Interesado** o solicitante, que es al final el beneficiario del contrato y sobre los que recaen las nóminas. Éstos últimos no tienen especial relevancia en el desarrollo de la aplicación ya que sus datos son gestionados por el Gestor, por lo que la aplicación también deberá permitir, la gestión de sus datos, la solicitud de convocatorias y creación de nóminas. Vemos un resumen en la siguiente imagen de la entrevista previa, en la que, en una primera toma de contacto podamos entender qué quiere el usuario:

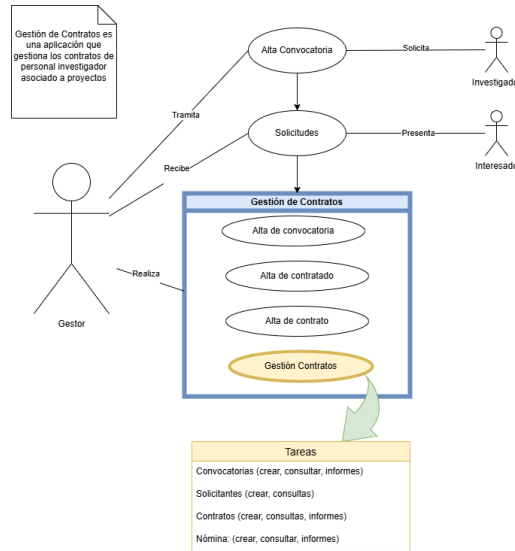


Figura 3.1: Casos de Uso

3.2. Diagramas de Secuencias

Los diagramas de secuencias son representaciones visuales que muestran cómo interactúan diferentes componentes del sistema para realizar una tarea, como vemos en la siguiente imagen, que recrea el proceso general que se quiere desarrollar con la aplicación:

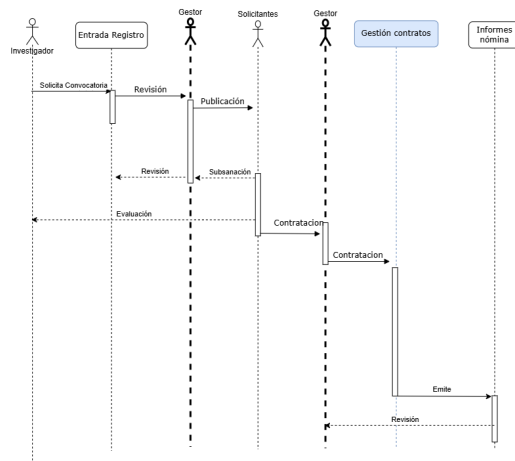


Figura 3.2: Diagrama de secuencias

cada Responsable, pudiendo así determinar cuantos proyectos son dirigidos por un responsable, sin duplicidad de datos.

- **3ªFN**: Para finalizar, eliminaremos las dependencias transitivas de los campos de cada tabla, es decir, si un atributo no depende directamente de su **PK**, deberá ir en otra tabla. En nuestro caso, no tiene sentido que cada mensualidad esté con el nombre de la persona, siendo más lógico crear su propia tabla.

Finalizada esta fase obtenemos varias tablas para organizar los datos recogidos en la hoja de cálculo; **Responsables**, **Departamentos**, **Proyectos**, **Convocatorias**, **Solicitante**, **Contratos** y **Nomina**.

3.4. Modelo E-R

En esta segunda fase del diseño se establece cómo se conectan entre sí los diferentes objetos (tablas) del sistema. Esto permite organizar visualmente la estructura de la información antes de crear la **BDR** definitiva y entender cómo se almacenan y relacionan los datos. Estas tablas ya contienen los distintos atributos que las definen unívocamente así como las **FK** que permiten establecer las relaciones con otras entidades. Hablamos aquí del concepto de **Cardinalidad**, por el que básicamente se indica cuántos elementos de una entidad puede relacionarse con los de otra, siendo una característica fundamental en el diseño de la **BDR**. Los tipos de relaciones **1:1** uno a uno, **1:N** uno a muchos, **N:M** muchos a muchos.

Hay que tener en cuenta que la **Cardinalidad**, es un elemento fundamental en el diseño ya que optimiza las consultas, ayuda a mantener la integridad y evita la redundancia de datos garantizando la consistencia de la información.

3.5. Diseño Físico

Una vez finalizada la parte lógica y conceptual de nuestro diseño es el momento de definir cómo serán nuestras tablas físicamente, qué campo será la **PK**, tipos de datos y longitud de los mismos, como se relacionarán las tablas entre sí, qué campos servirán de enlace o **FK**. Así, vemos en la imagen siguiente como queda nuestro diseño.

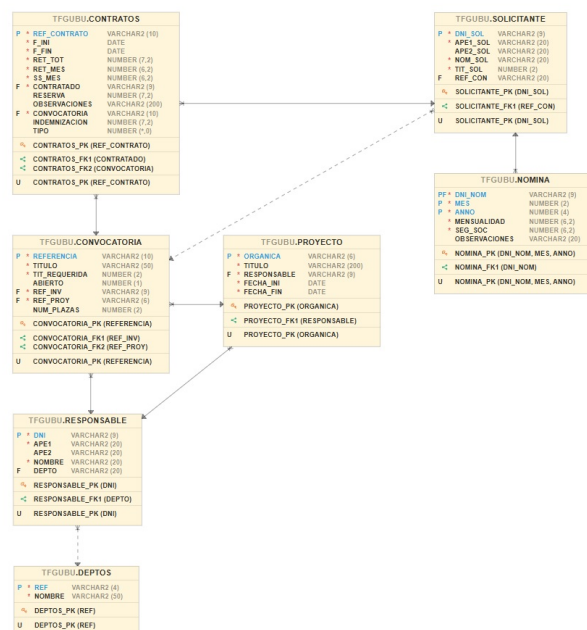


Figura 3.4: Diagrama de Entidad Relación

3.6. Sistemas de Gestión de Bases de Datos (SGBD)

Los **SGBD** son software que permite crear, gestionar y acceder a la Base de Datos interactuando entre datos y aplicaciones, permitiendo a los Administradores de las mismas una gestión eficaz del sistema. Se denomina así, al conjunto formado por la BD, el *Sistemas de Gestión de BD (SGBD)* y los programas de aplicación que dan servicio a la entidad o empresa [8][13]. Las principales funciones son:

- **Definición de la BD**, mediante un *Lenguaje de Definición de Datos (DDL)*, que permite especificar la estructura, tipo y restricciones de los datos.
- **Inserción, supresión, consulta y actualización de datos**, mediante un *Lenguaje de Manipulación de Datos (DML)*. Este tipo de lenguajes **no procedurales**, es decir, que no operan sobre los registros, solamente lo que quieren obtener, son los utilizados por los **SGBD**, siendo el más utilizado y estándar de facto *Structured Query Language (SQL)*.

- **Seguridad**, mediante un acceso controlado.
- **Integridad y consistencia** de la BD.
- **Control de concurrencia**, permitiendo el acceso compartido a la BD.
- **Diccionario de datos**, que contiene la descripción de los datos de la BD.

Entre los **SGBD** más utilizados podemos destacar **MySQL**, **PostgreSQL**, **SQL Server** y el escogido para este trabajo **Oracle**, principalmente por su conexión con la herramienta de desarrollo de la aplicación **APEX**, y por ser uno de los más extendidos en entornos comerciales, utilizado por la propia Universidad y con una versión gratuita como veremos en mas detalle posteriormente.

Una vez elegida la herramienta de gestión es preciso la carga de datos en las tablas que inician el proceso de contratación y que de una manera deben ser más o menos estáticas (aunque en el desarrollo se permita su modificación), como son **Responsables** y **Proyectos**. Para rellenar estas tablas se ha utilizado un generador de **Datos MOCK** en python:

```

1  # Listas de nombres y apellidos base
2  nombres = ["Laura", "Carlos", "Ana", "Miguel", "Lucia", "
        Pedro", "Maria", "Javier"]
3  apellidos = ["Garcia", "Lopez", "Martinez", "Sanchez", "
        Ramirez", "Torres", "Vega", "Diaz"]
4  departamentos = ["V103", "V104", "V105", "V110", "V112", "
        V118", "V121", "V122", "V123"]
5
6  # Crear archivo CSV
7  with open('datos_personas.csv', 'w', newline='') as archivo:
8      writer = csv.writer(archivo)
9      writer.writerow(['DNI', 'APE1', 'APE2', 'NOMBRE', 'DEPTO'])
10
11  for i in range(1, 101):
12      dni = f"{i:08d}A"
13      # Nos aseguramos de que no se repitan nombre y apellidos
        exactamente igual
14      while True:
15          ape1 = random.choice(apellidos)
16          ape2 = random.choice(apellidos)
17          nombre = random.choice(nombres)
18          if not (ape1 == ape2 == nombre):
19              break
20      depto = random.choice(departamentos)

```



```
21 | writer.writerow([dni, ape1, ape2, nombre, depto])
```

Código 3.1: Generación de *Datos MOCK* para la tabla Responsables

Para los **Departamentos**, se ha establecido una lista de valores estática, ya que éstos son predeterminados por la entidad y no pueden ser modificados por el usuario.

3.7. Agilidad-Scrum

Scrum es un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos [14]. La agilidad se fundamenta en los **sprints**, que son eventos de longitud fija para crear consistencia. Durante los mismos se produce el trabajo establecido durante la planificación del mismo.

El equipo Scrum (Scrum Team)

Para el desarrollo de la metodología ágil, es preciso la confección de un equipo de trabajo compuesto por; (**Product Owner**, propietario del producto, que es más bien un enlace entre la empresa y el equipo y puede representar a varias intervinientes externo, **Scrum Master**, es el líder del equipo y se encarga de que la metodología se desarrolle correctamente, ayuda al equipo y sirve de nexo de unión entre todos los demás componentes y los **Desarrolladores** que se encargan de crear cualquier incremento útil de un sprint. Evidentemente en el desarrollo de este **TFG**, varios roles son asumidos por el Tutor, siendo el desarrollo a cargo del alumno.

Planificación

Para realizar toda esta planificación se ha usado un tablero **kanban** que permite organizar y priorizar las historias de usuario. Estas tareas se vincularon al repositorio **GitHub** que permite obtener una trazabilidad entre el código y el sprint, aplicándose una integración continua manual.

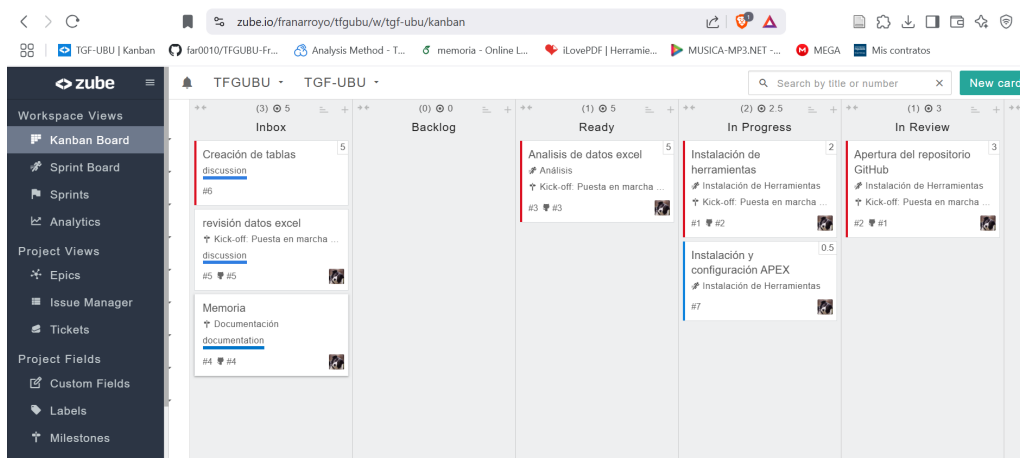


Figura 3.5: Kanban TFGUBU

Revisión del sprint

Tras cada **sprint** el código es analizado por **SonarCube**, para detectar errores, vulnerabilidades, problemas de estilo y garantizando la calidad. Así mismo, cada funcionalidad añadida por un sprint ha sido validada por **TestCafe** cuyos resultados se presentan visualmente en los informes generados por **Allure**.

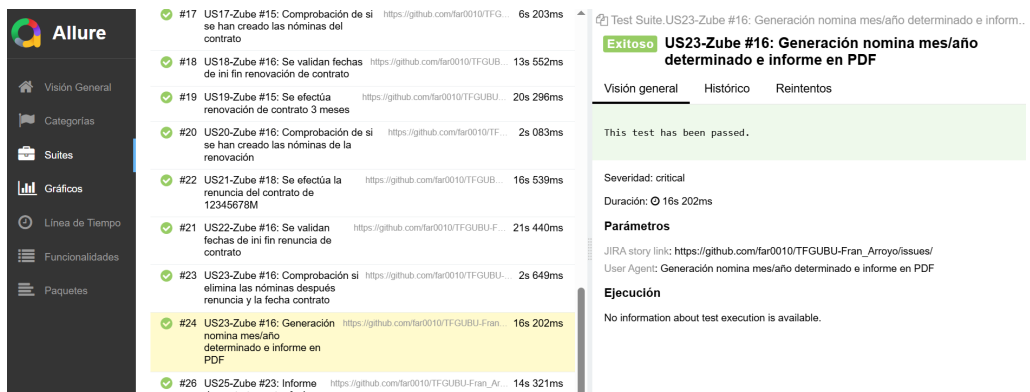


Figura 3.6: Informe de test Allure

Integración y pruebas

Debido a que el proyecto se ha desarrollado en **APEX**, que es una plataforma **low-code** que corre sobre Oracle Database y se gestiona desde

un entorno web, es imposible realizar la integración automatizada a través de GitHub Actions, ya que no permite fácilmente levantar el entorno local, por lo cual se optó por una estrategia de **integración manual continua**, teniendo así control del software generado. Una vez desplegada la aplicación sí sería posible implementar test automatizados, pero se perdería el fin del método de desarrollo para este trabajo.

Releases

Durante el desarrollo del proyecto se han realizado tres entregas (**Release**), **Prototipo**, **MPV** y la versión final, con todas las funcionalidades implementadas.

Milestones

Aunque en **Scrum**, no es un elemento obligatorio la creación de **Milestone** "hitos", para un mejor control de tiempos se ha considerado su inclusión dentro de la definición del **kanban**, incluyendo dentro de los mismos los **sprints** implicados, quedando así definidos:

- **Kick-off: Puesta en marcha del proyecto 09-06-25:** Tras las primeras conversaciones con el Tutor del proyecto se inicia esta fase de puesta en marcha, que se basa principalmente en recopilar información, e instalación de las herramientas necesarias.
- **Prototipo 01-07-25:** Finalizada la fase anterior se comienza el desarrollo del prototipo de la aplicación, que es una imagen visual de lo que se quiere llevar a cabo.
- **MPV 14-08-25:** Realizadas las operaciones más importantes se lanza la versión mínima para enseñar a los usuarios y comprobar que cumplen los requisitos y en caso contrario hacer modificaciones.
- **Desarrollo completo 31-08-25:** La aplicación ha quedado finalizada y se realiza el despliegue en Oracle cloud para su visualización.
- **Dokumentación 06-09-25:** Este último hito da por finalizado el desarrollo del **TFG**, y consiste en generar toda la documentación pertinente. Dicha documentación se ha ido elaborando durante la vida del proyecto.

3.8. Objetivos de Desarrollo Sostenible **ODS**

Los Objetivos de Desarrollo Sostenible son 17 objetivos globalmente acordados adoptados por la Asamblea General de las Naciones Unidas en 2015 como parte de la Agenda 2030 para el Desarrollo Sostenible. Los objetivos abordan de forma integral las tres esferas del desarrollo sostenible: la ambiental, la social y la económica. Además, abarcan áreas críticas como la pobreza, la desigualdad, la inclusión social, la energía sostenible, el cambio climático, la educación de calidad y la innovación tecnológica. [12]

En respuesta a modernizar la administración se desarrolla esta herramienta **GeNomIn** sobre la plataforma digital **APEX** y desplegada en Oracle Cloud, con el objetivo de sustituir procesos basados en hojas de cálculo Excel y almacenamiento compartido por una solución más integral, escalable y alienada con los citados principios y en particular:

ODS	Aplicación en el sistema	Impacto esperado
ODS 9: Industria, innovación e infraestructura	Sustitución de procesos manuales por digitalización completa.	Aumento de la eficiencia, innovación operativa
ODS 12: Producción y consumo responsables	Eliminación del papel y archivos locales	Reducción de residuos físicos y duplicidades
ODS 13: Acción por el clima	Uso de Oracle Cloud con enfoque verde	Disminución de la huella energética institucional
ODS 16: Paz, justicia e instituciones sólidas	Control documental, trazabilidad y acceso seguro	Transparencia organizacional y fortalecimiento de la gobernanza

Tabla 3.1: *Objetivos de Desarrollo Sostenible (ODS)*

Se puede hacer una estimación del impacto estimado con el seguimiento:

- **Reducción del uso de papel:** hasta 90
- **Ahorro de tiempo administrativo:** entre 30–50
- **Minimización de errores manuales:** hasta un 70 por ciento por validaciones automatizadas.

- **Migración energética eficiente:** reemplazo de equipos físicos por infraestructura cloud.
- **Consolidación documental:** simplificación del entorno de trabajo en una única plataforma digital.

Este compromiso de Oracle con **ODS**, se muestra en la infraestructura utilizada, en la que;

- sus centros son **100 por ciento de Energía renovable**,
- su arquitectura hardware se realiza con **diseño circular y reciclaje certificado**,
- sus servicios autónomos están optimizados para **eficiencia energética**
- y el uso de herramientas Oracle ESG, para medición del **impacto ambiental**

(**Oracle y sostenibilidad**)

4. Técnicas y herramientas

Otro de los aspectos a tener en cuenta en la realización de cualquier proyecto, ya sea de albañilería o de desarrollo informático, son las herramientas con las que se realiza, ya que de ello dependerá mucho tanto el resultado como la planificación de la ejecución. En este capítulo se detallarán las herramientas escogidas para el desarrollo de la aplicación **GeNomIn** y su justificación.

Puesto que la aplicación se desarrolla sobre BD, la primera elección tenía que estar condicionada por qué **SGBD** elegir. Como se indica en el capítulo anterior (3.6) entre los más usados comercialmente están; **MySQL**, **PostgreSQL**, **SQL Server** y **Oracle**. Puesto que este **TFG** pretende ser un reflejo de las competencias adquiridas durante el Grado y dado que en la asignatura de **SGBD**, se usa también Oracle 9 parece una opción elegible. También se estudió el sistema empleado por la Universidad de Burgos para el almacenamiento de sus BD, siendo también Oracle, con lo cual fue la opción elegida. Siguiendo este mismo criterio de continuidad, para el desarrollo de la aplicación se escogió **APEX** 5.1, ya que era la plataforma utilizada en el Grado.

Expuesto el proyecto al Tutor y puesto que los objetivos de los **TFG** deben ser también una mejora en las competencias adquiridas, se optó por una versión más moderna que permitiera el despliegue completo en cloud, por lo que definitivamente se escogieron como principales herramientas Oracle23ai Free y **APEX**2402.

Con esta decisión también fue necesaria la configuración de **ORDS** y **HTTPS**, como se detallará más adelante.

4.1. Oracle23ai

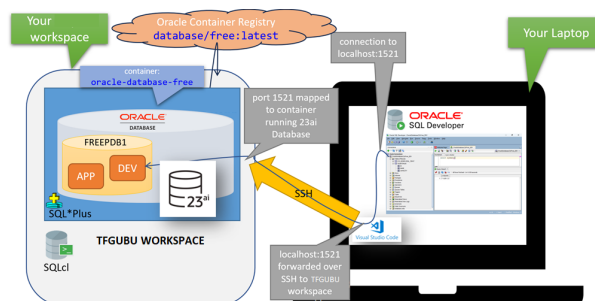


Figura 4.1: Esquema de Oracle23ai

Oracle23ai Free es una de las plataformas más usadas en el panorama comercial y que sigue ofreciendo su **IDE**, SQL Developer y varias funcionalidades de forma gratuita para desarrolladores. Entre las principales características se encuentran:

- **AI Vector Search:** (búsqueda vectorial de AI) es una colección de funciones que incluye un nuevo tipo de datos vectoriales, índices vectoriales y operadores SQL de búsqueda vectorial que permiten a **ORACLE 23ai (ODB)** almacenar el contenido semántico de documentos, imágenes y otros datos no estructurados como vectores y utilizarlos para ejecutar consultas de similitud rápidas.
- **JSON Relational Duality Views:** las vistas de dualidad relacional de JSON unifican los modelos de datos relacionales y documentales para ofrecer lo mejor de ambos mundos.
- **Gráficos de propiedades operativas:** Ofrece soporte nativo para estructuras de datos de gráficos de propiedades y consultas de gráficos.
- **SQL Firewall:** función de seguridad de base de datos integrada en el núcleo de **ORACLE 23ai (ODB)** que inspecciona todas las conexiones entrantes a la base de datos y sentencias SQL, y permite/registra/bloquea actividades no autorizadas de acuerdo con políticas específicas del usuario de base de datos.
- **True Cache:** esta solución simplifica el almacenamiento en caché en **ORACLE 23ai (ODB)**.

- **Mejoras de SQL:** incluye nuevas funciones, como dominios de uso de aplicaciones, que permiten a los desarrolladores definir las columnas que representan.
- **Mejoras en la escalabilidad y disponibilidad**

El proceso de desarrollo fue instalar localmente [10] para una vez finalizado el producto desplegarlo en la nube.

4.2. Apex 24.02

Para el desarrollo de **GeNomIn** se ha utilizado, como se ha comentado anteriormente **APEX** 2402, instalado inicialmente también de forma local [11]

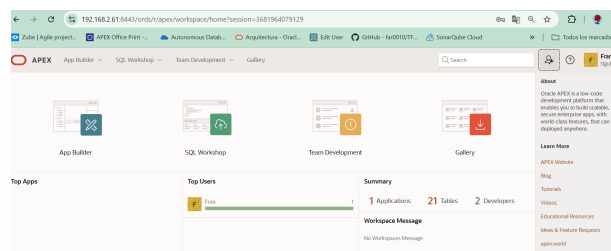


Figura 4.2: Home Apex TFGUBU-GeNomIn

Oracle Apex 24 es una plataforma **low-code** para desarrollos empresariales más utilizada que permite crear aplicaciones móviles y web escalables y seguras, pudiéndose implementar en la nube o localmente de forma gratuita.

Sus principales características son:

- **IA Generativa:** ofrece capacidades de IA generativa mejoradas, incluyendo un asistente de IA más potente y la posibilidad de configurar datos RAG (Retrieval-Augmented Generation)
- **Database Object Dependencies:** permite escanear y revisar los Objetos de Base de datos que están siendo referenciados en nuestra aplicación.
- **Fuentes de datos REST mejoradas:** amplía su capacidad para integrarse con APIs externas y servicios web de forma más flexible y potente.

- **Soporte para fuentes JSON:** permite trabajar con datos en formato JSON sin necesidad de almacenarlos en tablas tradicionales.

4.3. ORDS

Oracle **REST** Data Service Oracle REST Data Services sirve de puente entre **HTTPS** y tu Oracle Database. ORDS, una aplicación Java de nivel medio, proporciona una API REST de gestión de bases de datos, SQL Developer Web, una puerta de enlace PL/SQL, SODA para REST y la capacidad de publicar servicios web RESTful para interactuar con los datos y los procedimientos almacenados en su Oracle Database.

Puesto que era necesario el acceso **HTTPS**, fue necesaria la generación e instalación de **SSL**, para ello se generó un certificado con OpenSSL y su posterior con configuración, como se muestra en la imagen.

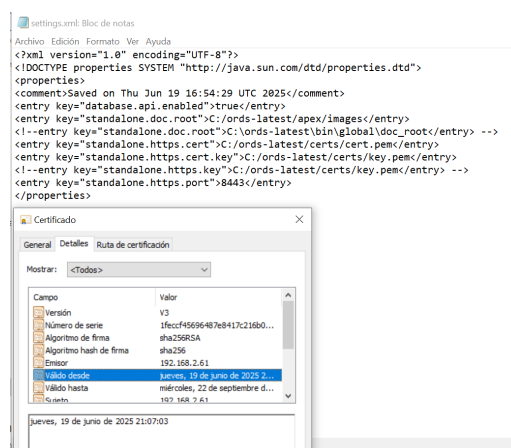


Figura 4.3: Configuración ORDS-HTTPS

4.4. TestCafe

TestCafe es un **IDE** multiplataforma para pruebas web integrales que no requiere WebDriver ni otras herramientas. TestCafé Studio funciona en Windows, macOS y Linux, y puede ejecutar pruebas en cualquier navegador de escritorio o móvil. [7] Principalmente se escogió este **IDE**, como en el caso de la versión mejorada de **APEX**, por ofrecer un desarrollo en entornos no manejados durante el Grado, tales como **Selenium**. Las principales características que ofrece son:

- **Grabación de pruebas y captura de imágenes:** TestCafe permite la grabación y toma de imágenes de las pruebas efectuadas, pero en la práctica, se limita a unos pocos frames iniciales.
- **Habilitado para distintos navegadores:** las pruebas pueden ser ejecutadas en diversos navegadores, chrome, edge, firefox, safari, etc..
- **Conjunto completo de Assertions:** en cualquier momento se puede comprobar el estado de un elemento, contenido, posición, valores de retorno y demás propiedades.
- **Espera automática:** está diseñado para la web asíncrona moderna. Identifica correctamente eventos como la carga de páginas, la representación de elementos y las solicitudes XHR, y espera a que se resuelvan antes de continuar con la prueba.
- **Selectores automáticos:** Las acciones y aserciones de prueba utilizan consultas de selector para identificar elementos por su clase, atributos, ID u otras propiedades. Cuando se registra una prueba, se genera consultas de selector automáticamente.
- **Informes completos:** genera informes de pruebas completos que incluyen resúmenes de las ejecuciones, así como información detallada sobre cada prueba. En nuestro desarrollo se han utilizado los informes **Allure**, ver imagen(3.7)

En la código siguiente vemos un ejemplo de test 25 que comprueba si se muestra el informe entre fechas y en la imagen su ejecución:

```
1  import { Selector } from 'testcafe';
2  import {NOMINAMENU, BUTTON, USERNAME, PASSW, TOGICON4,
   BT_CONNOM, VTOMENU } from './constanst.js';
3
4  fixture 'Test Suite-25'.page("https://192.168.2.61:8443/
   apex/f?p=100:LOGIN_DESKTOP:12651011480748::::")
5  .meta({TEST_RUN: 'Informe de contratos entre fechas',
   FEATURE: 'Informes', STORY: 'US25-Zube #23'});
6
7  test.meta({SEVERITY: 'critical', ISSUE_URL: 'https://github
   .com/far0010/TFGUBU-Fran_Arroyo/issues/16',
8  STORY: 'US25-Zube #23'})
9  ('US25-Zube #23: Informe de contratos entre fechas', async
   t => {
10     const INI = '01-ENE-2025';
11     const FIN = '31-DIC-2025';
12     // Espera a que los campos esten disponibles
```

```

13     await t.expect(USERNAME.exists).ok({ timeout: 5000 });
14     await t.expect(PASSW.exists).ok({ timeout: 5000 });
15
16     // Interaccion con los elementos
17     await t
18       .typeText(USERNAME, 'user01')
19       .typeText(PASSW, 'user01')
20
21     await BUTTON();;
22     await t
23       .click(TOGICON4) // Hace clic en el icono de despliegue
24       // Esperar a que aparezca Nomina mes
25       await t.expect(VTOMENU.exists).ok();
26
27     // Hace clic en el enlace Nomina mes
28     await t.click(VTOMENU);
29     // introducir fechas inicio - fin
30     const BT_VTOS = Selector('#B18949343265529115')
31     await t
32       .typeText(Selector('input[name="P17_FDESDE"]'), INI)
33       .typeText(Selector('input[name="P17_FHASTA"]'), FIN)
34       .click(BT_VTOS);
35
36     // comprobamos que ofrece las filas deseadas 3
37     const filasDatos = Selector('#
38       R18949894423529120_data_panel tbody tr')
39       .filter(node => node.querySelectorAll('td').length > 0);
40     await t
41       .expect(filasDatos.count)
42       .eql(3, 'La tabla no muestra exactamente 3 filas de
43         datos');
44   });

```

Código 4.2: Codigo en js para prueba en TestCafe

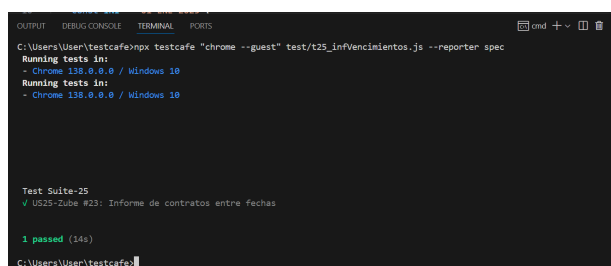


Figura 4.4: Resultado del Test 25:comprueba el informe entre fechas

Una vez realizado el desarrollo y completadas las pruebas establecidas en los **Caso de Uso**, se realiza el despliegue. Así, siguiendo la misma operativa

en el desarrollo del **TFG**, se opta por el despliegue en herramientas gratuitas, en este caso Oracle sigue ofreciendo su versión **cloud free tier**, para desarrolladores **Oracle Cloud Infrastructure (OCI)**.

4.5. Oracle Cloud Free Tier

Como se ha indicado anteriormente **OCI**, ofrece una serie de servicios "*gratuitos*" para desarrolladores en su nube. Tras el proceso de registro, que no es nada sencillo, ya que es preciso aportar una tarjeta física en vigor (con un cobro de 0,73€) y los datos son revisados. Para este trabajo especialmente se accede a:

- **Autonomous Data Base**: Base de datos autónoma sin costo que se gestiona automáticamente (aprovisionamiento, seguridad, disponibilidad, rendimiento, etc.)
- **AMD Compute Instance**: 2 **VM** basadas en AMD, con 1/8 de OCPU y 1 GB de memoria cada una.
- **APEX**: plataforma para el desarrollo de aplicaciones low-code

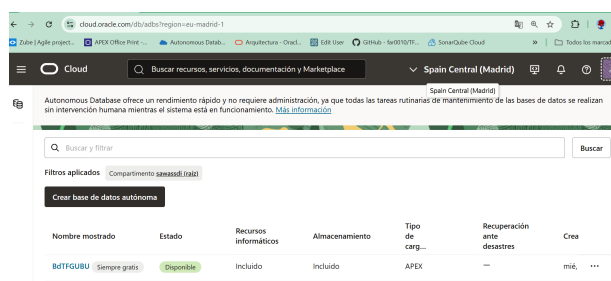



Figura 4.5: Base de Datos Autónoma en Oracle Cloud

Una vez creada la instancia de la base de datos autónoma, es preciso crear la instancia de **APEX** e iniciar el servicio y así poder proceder a la importación tanto de la BD como la aplicación generada localmente. Para el proceso de exportación se utilizó el asistente de SqlDeveloper e importando los datos en Cloud a través de las acciones de la BD, sql en su hoja de trabajo. Se intentó realizar el proceso a través de pump de datos (que realiza una copia especular), pero principalmente por compatibilidades de configuración (zona horaria) resultó imposible.

4.6. Apex Office Print

Uno de los objetivos principales de la aplicación, además de la gestión de los contratos, es ofrecer informe detallado de las nóminas que se pagan en un mes determinado. Queriendo ofrecer una personalización de los mismo se utiliza el servicio **AOP**, que permite generar informes Office y PDF, pudiendo personalizar éstos [5]. Para ello es necesario darse de alta en el servicio y generar una plantilla, en la que se detallen los campos del informe a imprimir:




UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

Genomín: 02/08/2025 9:08:12

INFORME NÓMINA: {{InformeNomPDF2[0].MES_PEDIDO}} / {{InformeNomPDF2[0].ANNO_PEDIDO}}

ORGANICA	NOMBRE	MENSUALIDAD	SEC-SOCIAL	TOTAL	OBSERVACIONES
{{InformeNomPDF2[0].ORGANICA}}	{{NOMBRE}}	{{MENSUALIDAD}}	{{SEC_SOC}}	{{TOTAL}}	{{OBSERVACIONES}}
TOTAL GENERAL				{{InformeNomPDF2[0].TOTAL_GENERAL}}	



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

Genomín: 02/08/2025 9:08:12

INFORME NÓMINA: (12 / 2025)

ORGANICA	NOMBRE	MENSUALIDAD	SEC-SOCIAL	TOTAL	OBSERVACIONES
ORG33	María Martínez Sánchez	1000	300	1300	
ORG55	Pedro Picapiedra Dura	1450	120	1570	Último mes con indemnización
ORG55	Juncal Álvarez Leal	1250	360	1610	
TOTAL GENERAL				4480	

Figura 4.6: Plantilla de informes-Informe AOP

4.7. GitHub

Es una plataforma para alojamiento y gestión de código fuente que se basa en el sistema de control de versiones Git, siendo una herramienta fundamental en el desarrollo de proyectos [9]. Para este proyecto se ha utilizado junto con la versión de escritorio **GitHub Desktop**, haciendo commits tanto del nuevo software como de las actualizaciones y subiéndolas al repositorio (push).

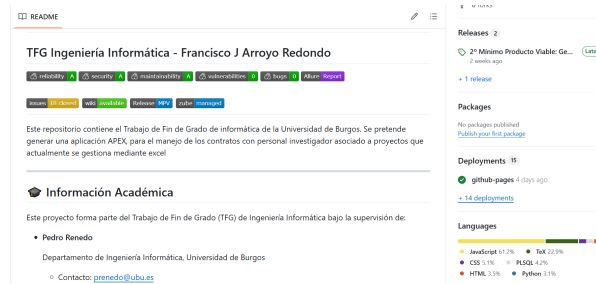


Figura 4.7: GitHub del Proyecto GeNomIn

Control de Versiones

: las diferentes versiones del desarrollo realizado, están disponibles a través de los commits. Principalmente se han actualizado código **SQL**, **PL/SQL** y **JavaScript**, además de la diversa documentación generada. Es posible la solicitud de cambios de código al resto de colaboradores (pull-requests), aunque en el ámbito de este desarrollo individual no tiene mucho sentido.

Integración de Herramientas

Como se puede observar en la imagen anterior, **github**, permite la integración de diversas herramientas que permiten un mayor control del proyecto, tales como **SonarCloud**, para verificar el código a través de *Actions*, **Zube** e informes de **Allure**

Documentación

El repositorio permite la organización de la documentación de forma estructurada, así como la creación de *Wikis*, para facilitar la comprensión del proyecto.

Release

Una de las partes importantes de un proyecto, es la generación de versiones que indican el progreso del mismo. GitHub facilita esta tarea con su funcionalidad de *releases* que permiten empaquetar y distribuir las distintas versiones del producto de forma organizada y documentada.

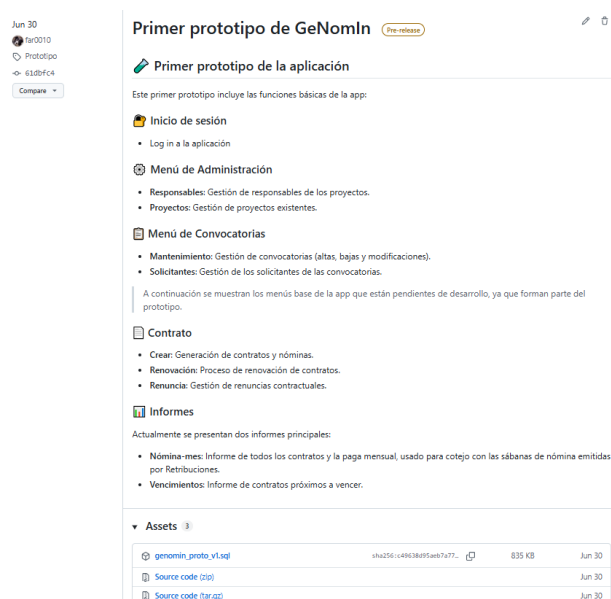


Figura 4.8: Vista de un release

Proyecto GeNomIn en GitHub

4.8. Zube

Zube es una plataforma para gestión de proyectos colaborativos. Se basa en tableros **Kanban** que facilitan la planificación, la gestión de **sprints**, **Milestone** y demás herramientas gráficas que permiten un análisis del trabajo a realizar y realizado. Ver imagen(3.7)

Zube está perfectamente integrado con **GitHub** permitiendo una sincronización de tareas y código fundamentales para la planificación del proyecto.

4.9. TextStudio

TextStudio es una herramienta especializada en la edición de textos **LATEX** que facilita la redacción de trabajos académicos y científicos mediante características como el resaltado de sintaxis, corrección ortográfica y semántica en tiempo real y fundamentalmente, la *compilación automática*, la cual permite previsualizar el documento mientras se trabaja. Es cierto que al estar acostumbrados a los productos office, inicialmente resulta inquietante", pero los resultados son más profesionales [1].

5. Aspectos relevantes del desarrollo del proyecto

En este capítulo se detallan los aspectos que han influido en el desarrollo de la aplicación **GeNomIn**, tanto técnicas, económicas y de oportunidad.

5.1. Oportunidad de negocio

Cuando se aborda un **TFG**, según mi opinión, debe poder trasladarse al ámbito profesional y así desarrollar las competencias adquiridas durante estos años de aprendizaje en el Grado. En el entorno laboral en el que desarrollo mi trabajo, existía la necesidad de un control más férreo y menos tedioso de los pagos de nómina a través de proyectos de Investigación que se estaban realizando a través de hojas de cálculo. En este sentido, cabe explicar que no es que se pague mal, si no en el proceso de comprobación utilizado, en lo relativo a los contratos realizados con proyectos de investigación. Mensualmente se emite un listado de los pagos que corresponden a estos proyectos y se cotejan contra una hoja de Excel, y con más de 400 contratos, es evidente que resulta inviable. (ver imagen **3.3**) Así pues, se planteó la propuesta de crear una aplicación, que no sólo ofreciera un tipo de listado más uniforme, si no que permitiera un control más adecuado de las personas contratadas, las convocatorias y sus responsables. En este sentido, hay que indicar, que la plataforma que actualmente utiliza la Universidad, ofrece la gestión de convocatorias y contratos, pero no es posible el control de pagos de estos contratos con los módulos instalados.

5.2. ¿Por qué APEX 2402?

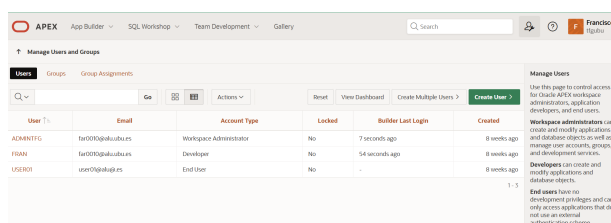
Uno de los aspectos más importantes y que influyen en el desarrollo de una aplicación es la herramienta que se utiliza. En este sentido y puesto que el entorno de trabajo se ha desarrollado con **APEX**, parecía bastante adecuado, tratando así de que la transición de la hoja de cálculo a la aplicación sea lo menos costosa para los usuarios.

Como vimos en la sección dedicada a esta herramienta 4.2, **APEX** ofrece muchas características técnicas que ya de por sí solas la hacen elegible, pero veamos qué facilita para el desarrollo y la gestión.

Integración con **ORACLE 23ai** (ODB)

APEX aprovecha la potencia y eficiencia de **ODB** así como sus características avanzadas así como integración nativa con **SQL** y **PL/SQL** desarrollado por Oracle [4]

Gestión de Usuarios



User	Email	Account Type	Locked	Builder Last Login	Created
ADMINITG	far000@okulab.com	Workspace Administrator	No	7 seconds ago	8 weeks ago
FRAN	far000@okulab.com	Developer	No	54 seconds ago	8 weeks ago
USER01	user01@okulab.com	End User	No	-	8 weeks ago

Figura 5.1: Gestión de usuarios de APEX

Como vemos en la imagen anterior **APEX**, permite un control total de los usuarios a través de:

- **Listas de Control de Acceso (ACL)**: permiten asignar roles a usuarios específicos.
- **Esquemas de autorización**: que se aplican a páginas o componentes individuales para restringir acceso.
- **Generación automática de componentes**: al activar este **ACL**, se generan automáticamente distintos componentes; administración de usuarios, región de control de acceso, roles, compilaciones condicionales, etc. [3], [2]

5.3. Creación de las Páginas

APEX combina el desarrollo visual con la lógica declarativa y programación libre, para el desarrollo de aplicaciones web a través del **diseñador de páginas** (entorno visual para definir la estructura y componentes) y del **asistente de creación** (para generar formularios, informes interactivos, dashboards, etc.)

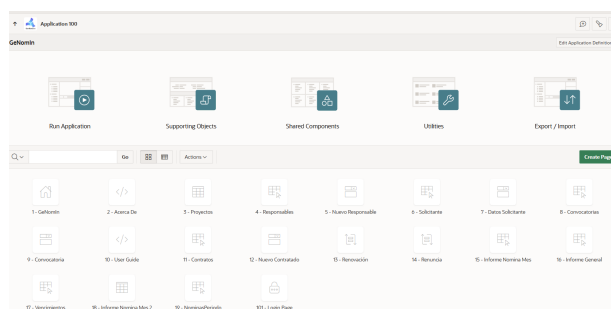


Figura 5.2: Vista de las páginas APEX

5.4. Programación

APEX permite la programación de procesos y acciones dinámicas en **PL/SQL**, **JS** sin complejidad de código. Además se pueden establecer condiciones y validaciones para mostrar u ocultar elementos según reglas establecidas.

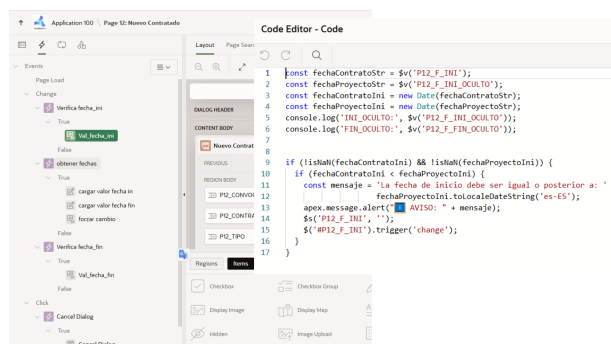


Figura 5.3: Acciones dinámicas APEX y su código js

En la imagen superior podemos ver un grupo de acciones dinámicas asociadas a la pag 12 del proyecto: **Nuevo contratado**, y se muestra el

código de la que verifica, si la fecha de inicio del contrato, es posterior a la fecha de inicio del proyecto asociado.

5.5. Despliegue en Oracle Cloud Free Tier

Una de las características principales por las que se eligió este modelo de desarrollo en el binomio **APEX ODB**, fue la gratuidad y facilidad de los servicios. Aunque como ya se explicó en el capítulo 4 (ver 4.5), el alta en Cloud, no es sencillo, se prefirió a otras opciones como **kubernetes**, con una curva de aprendizaje y configuración mayores o **virtualbox** que hubiera valido para un entorno local, exportando la imagen y confiando en la configuración del usuario de destino.

Así, para este tipo de desarrollos más "*experimentales*" se recomienda su uso, ya que además permite interactuar con entorno profesional, de bajo coste, alta disponibilidad, escalable (previo pago) y seguridad integrada, todo ello ofrecido por Oracle Cloud. Ver imagen(4.5)

6. Trabajos relacionados

Para la realización de este trabajo se hicieron diversas consultas a través de buscadores, de aplicaciones similares, que hicieran esta función de aunar los proyectos de investigación, sus convocatorias y sus contratos emitiendo informes personalizados. Así existen en el mercado aplicaciones como **PandaDoc** o **Zoho Contracts** que gestionan contratos, pagos, etc..., pero no están vinculados a convocatorias de proyectos de investigación.

En este sentido y ya conocido, **Universitas XXI**, sí ofrece varios módulos dedicados a la gestión de RRHH, proyectos de Investigación y gestión económica, con emisión de varios informes independientes, pero que no se adaptan por completo a los requerimientos de los usuarios. Veamos en esta tabla comparativa de servicios, entre la aplicación **GeNomIn** y los módulos de **UXXI**:

Funcionalidad	GeNomIn	UXXI-Inv	UXXI-Eco	UXXI-RRHH
Contratos Inv.	Parcial	Sí	No	No
Convocatorias Inv.	Parcial	Sí	No	No
Ges. Investigadores	Parcial	Sí	No	No
Informes nómina	Sí	Sí	General	Contable
Contratos con pagos	Sí	NO	General	Contable
Informes Person.	Sí	Propio	Propio	Propio
Infraestructura	Oracle Cloud	Propio	Propio	Propio
Flexibilidad	Alta	Versiones	Versiones	Versiones
Costo	Free Tier	Licencia	Licencia	Licencia

Tabla 6.1: Comparativa GeNomIn vs Universitas XXI

Como podemos comprobar la opción comercial tiene muchísimo más desarrollo pero no se personaliza a los objetivos de cada usuario, teniendo que esperar a la liberación de versiones, si es que está incluida esa actualización o solicitar una personalización, que generalmente no es barata.

7. Conclusiones, valoración personal y trabajo futuro

En este capítulo se detalla si se han cumplido los Objetivos del proyecto, tanto funcionales, no funcionales y personales, y si se han desarrollado y consolidado las competencias adquiridas durante el Grado.

7.1. Cumplimiento de objetivos funcionales

El principal objetivo de este **TFG**, era la transformación de una hoja de cálculo Excel, en la que se contabilizan las nóminas del personal contratado (ver imagen 3.3), en una **ODB** y la generación de una aplicación en **APEX**, que además de llevar la gestión de las convocatorias y contratos, presentara un informe para el cotejo con la retribución emitida por RRHH (ver imagen 4.6).

Una vez finalizado el desarrollo, el proyecto ha logrado cumplir estos objetivos, desarrollando una aplicación **Genomin**, que cumple a la perfección todos los retos planteados.



Figura 7.1: Logotipo de la aplicación: GeNomIn

7.2. Cumplimiento de objetivos no funcionales

Dentro de los objetivos no funcionales se pretendía reforzar las competencias adquiridas durante el Grado, así como explorar nuevas herramientas de desarrollo y ampliar algunas de las conocidas. Todos éstos retos han sido superados desde la gestión de proyectos con la **Metodología ágil**, generación de **ODB**, **SSL**, creación de la aplicación en **APEX** 24, realización de test de **VYP**, hasta su despliegue en **Oracle Cloud** y todo ello documentado y apoyado con el repositorio documental **GitHub**

7.3. Valoración Personal

Antes de empezar este **TFG**, tenía la sensación de que era un mero trámite más para la consecución del Grado en Informática, pero lo cierto es que es una de las asignaturas más enriquecedoras. Ya no solo supone un reto en cuanto al desarrollo de código, que hoy en día no lo es tanto con la irrupción de la **IA**, si no que supone un esfuerzo organizativo, que requiere de muchas de las competencias adquiridas durante la carrera. Es aquí donde se comprenden muchos conceptos, como la **Normalización**, **Modelo Relacional**, la gestión de proyectos, el uso de **Metodología ágil**, la **Validación y Pruebas (VYP)** etc., en el entorno de desarrollo de un proyecto. En mi caso la realización de este proyecto una vez finalizado el curso y completadas el resto de asignaturas, me ha permitido dedicar mucho más tiempo, organización e investigación que de otra manera hubiera resultado muchísimo más complejo.

Con todo esto, la aplicación web **GeNomIn** desarrollada, desde mi punto de vista, cumple con los objetivos inicialmente planteados y podría ser desplegada perfectamente para su uso en el Servicio para el que ha sido diseñada, puesto que corre sobre **ORACLE 23ai (ODB)**, la BD utilizada por la Universidad y teniendo como plataforma **APEX**, utilizado por **UXXI-Investigación** (actualmente proveedor de la Universidad de Burgos), lo cual no supondría ningún problema de adaptabilidad por parte de los usuarios.

7.4. Trabajo futuro

El trabajo abordado en este **TFG**, nace de la necesidad observada en la gestión y cotejo de pagos al personal investigador contratado, con cargo a proyectos de investigación. En este sentido se ha abordado la generación

de este informe (**Pago nómina-mes**), otros como la renovación y la renuncia de contratos, e informes adicionales que muestran información sobre vencimientos de contratos en un periodo determinado y de carácter general.

Quedaría por abordar, las **subidas salariales** y **bajas médicas** que también tienen influencia en los pagos mensuales, que tal y como está ahora mismo la aplicación, se podrían realizar modificando directamente los pagos en la tabla correspondiente (**NOMINAS**), pero sin tener una gestión directa. Seguramente, si se llegase a poner en explotación, sería imprescindible añadir otro tipo de **informes personalizados**, que surgieran del propio uso de la aplicación y la gestión diaria.

Evidentemente este proyecto simplemente es para uso de forma local, por el propio personal del Servicio de Investigación que gestiona contratos, ya que la plataforma **UXXI** desarrolla sus propios módulos bajo licencia, pero puesto que se ha considerado su necesidad podría ser planteado para desarrollos futuros a la propia empresa.

7.5. Reflexión sobre el uso de IA

Durante la realización de este trabajo se han realizado diversas consultas a través de **Copilot** sobre todo en "**momentos de pánico**". En este sentido cabe destacar, que en las versiones libres (la utilizada en este proyecto), son un mero apoyo y consulta, ya que normalmente las respuestas obtenidas ofrecen más una orientación, que una respuesta efectiva y que generalmente pueden llevar al error, sobre todo cuando se realizan, por ejemplo, consultas complejas. Evidentemente no se pueden poner puertas al campo y el uso de la **IA** está cada vez más extendido, sería deseable su inclusión en las asignaturas del Grado, como apoyo al desarrollo y al conocimiento, como en su día fueron, los libros, Internet y ahora esta nueva tecnología. Tendremos que adaptarnos todos.

Siglas

ACL Listas de Control de Acceso. 30

AOP APEX Office Print. 26

APEX *Application Express*. 1, 1, 3, 4, 5, 12, 14, 16, 19, 21, 22, 25, 30, 31, 32, 35, 36

BDR *Base de Datos Relacional*. 4, 7, 9, 10

DDL *Lenguaje de Definición de Datos*. 11

DML *Lenguaje de Manipulación de Datos*. 11

FK *Foreign Key-Clave Externa*. 9, 10

FN *Forma Normal*. 3, 9, 10

HTTPS *Secure Hypertext Transfer Protocol*. 4, 19, 22

IA Inteligencia Artificial. iv, 36, 37

IDE *Entorno de Desarrollo Integrado*. 20, 22

JS JavaScript. 31

MPV *Mínimo Producto Viable*. 15

OCI Oracle Cloud Infrastructure. 4, 5, 25

ODB *ORACLE 23ai*. 3, 20, 30, 32, 35, 36

ODS *Objetivos de Desarrollo Sostenible*. [III](#), [VI](#), [16](#), [17](#)

ORDS Oracle REST Data Services. [IV](#), [19](#), [22](#)

PK *Primary Key-Clave Primaria*. [9](#), [10](#)

PL/SQL Procedural Language/Structured Query Language. [30](#), [31](#)

REST REpresentational State Transfer. [22](#)

SGBD *Sistemas de Gestión de BD*. [III](#), [4](#), [11](#), [12](#), [19](#)

SQL *Structured Query Language*. [11](#), [30](#)

SSL Secure Sockets Layer. [22](#), [36](#)

TFG *Trabajo de Fin de Grado*. [5](#), [13](#), [15](#), [19](#), [25](#), [29](#), [35](#), [36](#)

UXXI Universitas XXI. [33](#), [36](#), [37](#)

VM Máquina Virtual. [25](#)

VYP *Validación y Pruebas*. [4](#), [36](#)

Glosario

Atomicidad Propiedad de un campo, por el cual no puede ser descompuesto en pedazos más pequeños por el DBMS. [9](#)

Cardinalidad Relación entre el número de instancias de una entidad con respecto a otra entidad con la que se relaciona. [10](#)

Caso de Uso Descripción de cómo los actores(usuarios), interactúan con el sistema para lograr un objetivo específico. [7](#), [24](#)

Copilot Asistente de inteligencia artificial desarrollado por Microsoft, diseñado para ayudar a los usuarios en diversas tareas a través de diferentes aplicaciones. [37](#)

Datos MOCK Datos ficticios para probar aplicaciones, bases de datos, interfaces, sin necesidad de usar datos reales o confidenciales. [12](#)

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.. [28](#)

LATEX Sistema de composición de textos orientado a la creación de documentos escritos, genralmente científicos que presenten una alta calidad tipográfica. [5](#), [28](#)

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.. 36

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. 15, 28

Modelo Relacional Herramienta que permite organizar los datos en tablas y las relaciones entre las mismas. 3, 36

Normalización Proceso de estructuración los datos en una BD relacional, eliminando redundancia de datos y manteniendo la integridad a través de Formas Normales. 3, 9, 36

Release Versión específica de un software que ha sido completada, probada y puesta a disposición para su uso por los usuarios finales.. 15, 27

Secure Sockets Layer protocolo de seguridad que cifra la comunicación entre un navegador web y un servidor, protegiendo la información transmitida. 4

Bibliografía

- [1] LaTeX tú TFG TFM Tesis desde cero y con estilo (4^a edición).
- [2] Oracle APEX Administration Guide.
- [3] Oracle APEX App Builder User's Guide.
- [4] Oracle APEX SQL Workshop Guide.
- [5] Overview | APEX Office Print (AOP).
- [6] Parte 1/2 - Configuración de APEX Office Print en una instancia de Oracle APEX.
- [7] Test Code Guide: TestCafe | DevExpress.
- [8] Henry F. Korth Abraham Silberschatz. *Fundamentos de Bases de Datos*. McGraw-Hill, 5^a edition.
- [9] Scott Chacon and Ben Straub. *Pro Git, Todo Lo Que Necesitas Saber Acerca de Git*. Versión 2.1.22-4-G2264d13. APRESS, 2^a edition.
- [10] 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.
- [11] Terri Jennings, David Bliss, Christina Cho, Hilary Farrell, Salim Hlayel, Christian Neumueller, Marc Sewtz, Jason Straub, Vlad Uvarov, Patrick Wolf, and John Godfrey. Installing and Configuring APEX and Oracle REST Data Services.

- [12] Urtzi Markiegi, Inigo Aldalur, and Alain Pérez. Integrando los ODS en el grado de Ingeniería Informática.
- [13] Mercedes Marqués. *Bases de Datos*. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions, 1^a edition.
- [14] Ken Schwaber and Jeff Sutherland. La Guía Definitiva de Scrum: Las Reglas del Juego.