



XUNTA
DE GALICIA

CONSELLERÍA DE CULTURA,
EDUCACIÓN, FORMACIÓN
PROFESIONAL E UNIVERSIDADES



IES SAN MAMEDE

© Rúa do Castelo N° 3. 32700. Maceda (Ourense)
☎ Tlf: 988 788 561 ✉ ies.san.mamede@edu.xunta.gal
🌐 <http://www.iesanmamede.com>



GESTHOR

[GestThor Git](#)

Nome Alumno/a: Laureano De Sousa Dias

Curso: 2º DAM

Módulo: *Proxecto Final Ciclo*

Proposta Proxecto

1

Contido

Contido	2
1. Introducción	3
2. Obxectivos	3
3. Situación previa	4
4. Tecnoloxías empregadas	5
Backend (Servidor Java - Spring Boot)	5
Backend (Servidor Python - OCR)	6
Frontend (Angular)	7
Almacenamento	8
Control de versións, Tests e implementación	8
Herramientas de Visualización e IA:	8
5. Solución proposta	9
6. Presuposto Hardware e Software	10

1. Introducción

GesThor é unha aplicación de xestión de gastos persoais e empresariais que facilita o rexistro, análise e optimización financeira. Destaca polo seu uso de OCR para extraer datos de tickets e facturas automaticamente, ademais da integración dunha IA que proporciona recomendacións de aforro e comparación de prezos. Tamén inclúe funcionalidades para a xestión de subscricións e a xeración de informes en PDF.

2. Obxectivos

Obxectivo iniciais:

- Desenvolver unha plataforma que permita aos usuarios rexistrar os seus gastos de forma automática ou manual.
- Implementar un sistema de escaneo de tickets mediante OCR.
- Integrar unha IA que analice os hábitos de gasto e propoña recomendacións.
- Ofrecer unha interface cómoda e intuitiva, compatible con múltiples dispositivos.
- Garantir a seguridade dos datos mediante autenticación segura e cifrado.
- Permitir a descarga e compartición de informes en PDF ou CSV.

Obxetivos acadados:

- Rexistro e xestión de gastos mediante escaneo de tickets físicos, documentos(Tickets) dixitais ou entrada manual.
- Clasificación automática dos gastos por categorías.
- Chatbot IA para suxestións e asesoramento financeiro.
- Posibilidade de sincronización multi-dispositivo mediante API segura.
- Compatibilidade con sistemas de autenticación externa como Google.

3. Situación previa

GesThor nace para resolver las dificultades que enfrentan muchos usuarios al organizar sus gastos debido a la falta de tiempo. El proceso manual de registro de facturas y tickets es no solo tedioso, sino también propenso a errores. Aunque existen aplicaciones de contabilidad, muchas de ellas carecen de funcionalidades avanzadas como el OCR o asistentes basados en inteligencia artificial.

GesThor ofrece una plataforma intuitiva, segura y escalable que facilita el registro de gastos mediante el escaneo de tickets, la carga de tickets digitales o la entrada manual de datos. Además, proporciona gráficos detallados e informes interactivos para que los usuarios puedan comparar y analizar sus gastos, filtrarlos y realizar un seguimiento eficaz.

En el futuro, se incorporarán funcionalidades adicionales como el escaneo de facturas electrónicas, lo que permitirá una mayor automatización del proceso, mejorando aún más la experiencia del usuario y la eficiencia del sistema.

4. Tecnoloxías empregadas

Backend (Servidor Java - Spring Boot)






-  **Spring Boot:**
Crea a API REST, implementa a lóxica de negocio e xestiona as dependencias do proxecto.
-  **Spring Security:**
Proporciona autenticación e autorización para protexer a aplicación.
-  **JWT (JSON Web Token):**
Xestiona a autenticación dos usuarios mediante tokens seguros e sen estado.
-  **OAuth2 (Google):**
Permite o inicio de sesión usando contas de Google.
-  **MySQL / PostgreSQL:**
Bases de datos relacionais para almacenar a información da aplicación.
-  **OpenAPI (Swagger):**
Documenta a API REST e facilita probas e integración con outras ferramentas.
-  **AWS SDK for Java / Spring Cloud AWS:**
Integra almacenamento na nube (ex. Amazon S3) para gardar arquivos.
-  **Ciente de conexión Java-Python:**
Permite a comunicación entre o backend Java e servizos Python.
-  **BCrypt:**
Encripta datos sensibles (como contrasinais) para almacenamento/transmisión segura.
-  **Lombok:**
Evita escribir código repetitivo como constructores, getters e setters.





Backend (Servidor Python - OCR)

- ⚡ **FastAPI (futura migración a Flask por seguridad):**
Framework para crear APIs REST de alto rendimiento, especialmente útil para procesos como o OCR.
- 🔍 **Tesseract OCR:**
Extrae texto de imaxes de tickets e facturas, permitindo a extracción automática de datos.
- 📄 **pdf2image:**
Converte arquivos PDF a imaxes para facilitar o seu procesamento mediante OCR.
- 📷 **OpenCV:**
Mellora e procesa imaxes para optimizar a extracción de datos usando técnicas de procesamento dixital.
- 📊 **NumPy:**
Facilita o procesado de datos numéricos e imaxes de forma eficiente, fundamental para operacións con arrays e cálculos matemáticos.
- 🌐 **Requests:**
Realiza chamadas HTTP para comunicarse con outras APIs (como a de Spring Boot), integrando servizos de forma fluída.
- 🧩 **re (expresións regulares):**
Identifica e valida patróns de texto extraídos, permitindo a extracción de información específica como datas, importes, etc.
- 📖 **Sphinx:**
Ferramenta para documentar a API REST, xerando documentación automaticamente a partir dos *docstrings*, o que facilita as probas e a integración.
- 🚀 **Uvicorn:**
Servidor ASGI de alto rendimiento que executa aplicacións FastAPI e permite peticións asíncronas.




Frontend (Angular)

-  **Angular:**
Desenvolve a interface web dinámica e interactiva para os usuarios.
-  **Bootstrap:**
Proporciona un deseño responsivo e atractivo para a UI, mellorando a experiencia visual.
-  **TypeScript:**
Versión tipada de JavaScript que mellora a robustez, mantenibilidade e escalabilidade do código en Angular.
-  **Chart.js / ng2-charts:**
Empregado para a xeración de gráficos interactivos e visualizacións de datos.
-  **Angular CLI:**
Facilita a creación de proxectos, desenvolvemento, xeración de compoñentes e tarefas comúns en Angular.



Almacenamento

-  **Almacenamento interno:**
A base de datos principal será **MySQL**, garantindo estabilidade e compatibilidade coa aplicación. Existe a opción de migración futura a **PostgreSQL** segundo necesidades ou melloras no rendemento.
-  **Almacenamento na nube (opcional):**
Segundo o tempo dispoñible no desenvolvemento, poderá integrarse con servizos na nube (como **Amazon S3**) para almacenar arquivos e documentos de forma externa e segura.

Control de versións, Tests e implementación

-  **GitHub:**
Xestión do código fonte, control de versións e colaboración entre desenvolvedores.
-  **Docker:**
Uso de contedores para despregar e manter a infraestrutura do proxecto de forma flexible e portátil.
-  **Postman:**
Proba e documenta os endpoints da API de forma interactiva, facilitando a integración e o desenvolvemento backend-frontend.

Herramientas de Visualización e IA:

-  **ChatGPT:**
Empregado para a xeración de documentación, explicacións técnicas e apoio ao desenvolvemento.
-  **DeepSeek:**
Utilizado para xerar datos de proba e comentar código de forma automática.
-  **Pika:**
Ferramenta de edición de vídeos con efectos visuais avanzados e capacidades de intelixencia artificial.
-  **ContentCore:**
Aplicación para creación de maquillaxes ou modificacións visuais en imaxes mediante IA.
-  **Canvas:**
Ferramenta para a creación de gráficos e visualización de datos de forma sinxela.
-  **Figma:**
Plataforma colaborativa para deseño de interfaces e creación de prototipos interactivos.
-  **Responsively.app:**
Aplicación para deseño e previsualización de interfaces responsivas en múltiples dispositivos.
-  **ASCII Art Generator:**
ascii-art-generator.org
Xera arte en texto a partir de imaxes ou palabras, útil para elementos creativos en documentación ou presentacións.

5. Solución proposta

Solución proposta GesThor proporciona unha plataforma de xestión de gastos con tecnoloxías avanzadas de recoñecemento de texto e intelixencia artificial. As funcionalidades principais inclúen:

- Escaneo de tickets dixitais e físicos con OCR para a extracción automática de datos.
- Entrada manual de gastos para correccións ou rexistros personalizados.
- Chatbot integrado que ofrece recomendacións financeiras.
- Xestión de subscricións e gastos recorrentes.
- Xeración de informes en PDF con detalles financeiros.
- Autenticación segura mediante JWT e compatibilidade con Google.
- Escalabilidade grazas a Spring Boot no backend e Python para o análise avanzado de datos.

Fluxo de funcionamento O cliente inicia sesión na páxina ou aplicación, obtendo un token de acceso. Con este token pode realizar peticións á API en Spring Boot, que se encarga de servir e gardar os datos. Ademais, haberá un servidor Flask/FastAPI que utilizará unha librería OCR optimizada en Python.

O proceso de carga de tickets funciona da seguinte maneira:

1. O cliente carga un ticket, que se envía ao servidor Spring Boot.
2. Spring Boot almacena temporalmente o ticket e o remite ao servidor Python.
3. O servidor Python realiza o proceso de OCR, extrae os datos e os devolve a Spring Boot.
4. Spring Boot preenche o ticket con estes datos e envía unha vista previa ao cliente.
5. O cliente pode aceptar, modificar ou descartar os datos extraídos.
6. Se os acepta, gárdanse a imaxe precargada e os datos definitivos; en caso contrario, descártase todo.

A comunicación entre Spring Boot e o servidor Python será mediante endpoints internos para garantir eficiencia e seguridade con claves e cifrados (Si funciona e da tempo).

6. Presupuesto Hardware e Software

Elemento Hardware	Descrición	Prezo
Procesador	Intel Core i5-9400F	140€
RAM	16GB DDR4	60€
Placa base	MPG Z390 GAMING PLUS	130€
Tarxeta gráfica	NVIDIA GeForce GTX 1660 Ti	200€
SSD	Kingston SA400M8120G (120GB)	25€
HDD	Seagate ST1000DM010-2EP102 (1TB)	40€
Caixa	NOX Hammer	30€
WiFi USB	Adaptador	2€
Periféricos	Teclado e rato HOCO	Gratis
Auriculares	Hi-Fi	Gratis
Tablet	10" para visualización	Gratis
Disco externo	WD 1TB	50€
Monitor 1	Nilox 24" reacondicionado	45€
Monitor 2	Philips 24" reacondicionado	65€



Elemento Software	Descrición	Prezo
SO	Windows 11 Home	Gratis
Backend IDE	IntelliJ IDEA	-
Frontend IDE	Visual Studio Code	-
Contedores	Docker	-
Control de versións	GitHub	-
Base de datos	MySQL Server (con opción de PostgreSQL)	-
API Testing	Postman	-
Frameworks	Spring Boot, Angular CLI, FastAPI/FaskAPI	-
OCR/IA	Librerías Python/IA ChatGPT	5€
Revisión de Código y Documentación, generación de Imágenes y animaciones,etc..	ChatGPT/DeepSeek/Pika/ContentCore/Canvas/Figma/responsively.app/	Gratis