|  |
| --- |
| Aplicación para la gestión del préstamo de libros |
| **1 de JULIO - 31 de AGOSTO - 2025**  IES. Arcipreste de Hita  Creado por: **Alejandro Trujillo Galán      https://github.com/Trujo666/proyecto\_alejandro\_trujillo** |





Contenido

[Estudio inicial de la aplicación 5](#_Toc199109029)

[Introducción 6](#_Toc199109030)

[Análisis de posibilidades 6](#_Toc199109031)

[Uso de una hoja Excel / manual 6](#_Toc199109032)

[Uso de una Base de datos y programación de la misma (Access) 6](#_Toc199109033)

[Creación de una aplicación propia 6](#_Toc199109034)

[Objetivo 6](#_Toc199109035)

[Requerimientos de la aplicación 6](#_Toc199109036)

[Requerimientos software y hardware 8](#_Toc199109037)

[Estudio de viabilidad 8](#_Toc199109038)

[Técnica 8](#_Toc199109039)

[Económica 8](#_Toc199109040)

[Operativa 8](#_Toc199109041)

[Legal 9](#_Toc199109042)

[Planificación del Proyecto 9](#_Toc199109043)

[Secuenciación de las fases del proyecto 9](#_Toc199109044)

[Planificación de recursos y tiempos 9](#_Toc199109045)

[Análisis 10](#_Toc199109046)

[Casos de uso 11](#_Toc199109047)

[Carga de datos (RF1 RF2) 11](#_Toc199109048)

[Gestión alumnos (RF3) 11](#_Toc199109049)

[Gestión prestamos (RF4 RF6) 11](#_Toc199109050)

[Gestión listados (RF7 RF8) 12](#_Toc199109051)

[Preservar los datos (RF9 RF10) 12](#_Toc199109052)

[Descripción de los casos de uso 13](#_Toc199109053)

[Carga de datos 13](#_Toc199109054)

[Gestión de alumnos 13](#_Toc199109055)

[Gestión de prestamos 14](#_Toc199109056)

[Gestión de listados 15](#_Toc199109057)

[Preservar los datos 15](#_Toc199109058)

[Diagrama Conceptual 16](#_Toc199109059)

[Diagrama de clases (diccionario de datos) 17](#_Toc199109060)

[Diagrama de actividades 18](#_Toc199109061)

[Generales 18](#_Toc199109062)

[Carga de datos 19](#_Toc199109063)

[Gestión de alumnos 20](#_Toc199109064)

[Gestión de préstamos 21](#_Toc199109065)

[Gestión de listados 22](#_Toc199109066)

[Preservar datos 23](#_Toc199109067)

[Esquema Entidad / Relación 24](#_Toc199109068)

[Modelo relacional y normalización 25](#_Toc199109069)

[Diseño del Proyecto 26](#_Toc199109070)

[Diseño físico de la Base de Datos 27](#_Toc199109071)

[Estructura de almacenamiento 27](#_Toc199109072)

[Diseño de procedimientos especiales 28](#_Toc199109073)

[Configuración de la aplicación. 28](#_Toc199109074)

[Estructura del fichero de datos 29](#_Toc199109075)

[Diseño de la estructura de clases y librerías (diagrama de clases) 29](#_Toc199109076)

[Diseño de la interfaz gráfica 30](#_Toc199109077)

[Diseño de movimiento de ventanas 30](#_Toc199109078)

[Diseño de las ventanas 31](#_Toc199109079)

[Implementar el proyecto 33](#_Toc199109080)

[Sql de creación (MySQL) 34](#_Toc199109081)

[Indicadores de calidad 38](#_Toc199109082)

[Elaboración de una batería de pruebas para detectar errores 39](#_Toc199109083)

[Evaluación y solución de incidencias 39](#_Toc199109084)

[Evaluación y seguimiento del proyecto 41](#_Toc199109085)

[Implantación del Proyecto 42](#_Toc199109086)

[Plan de implantación 43](#_Toc199109087)

[Manual de instalación 43](#_Toc199109088)

[Instalación de la aplicación 43](#_Toc199109089)

[Mejoras posibles 44](#_Toc199109090)

[Manual de usuario 44](#_Toc199109091)

[Anexos 46](#_Toc199109092)

[Formulario de contrato usado 47](#_Toc199109093)

[Fichero exportado de Delphos (alumnos) 48](#_Toc199109094)

[Ejemplo de formulario de satisfacción 49](#_Toc199109095)

[Datos de prueba 50](#_Toc199109096)

Tablas

**No se encuentran elementos de tabla de ilustraciones.**

Ilustraciones

**No se encuentran elementos de tabla de ilustraciones.**

# Estudio inicial de la aplicación

## Introducción

Dado el problema del aumento de préstamos en concesión de beca del instituto y la creciente gestión del mismo se detecta una necesidad de minimizar costes y tiempos en la gestión de este proceso. El mismo empieza al comienzo del curso en septiembre con la determinación del alumnado que ha sido becado y los libros existentes en depósito del año anterior. Del mismo modo, se deben configurar los datos de los cursos y libros en este periodo. Una vez realizada la configuración el proceso de gestión es tedioso y farragoso, en el que se incluye quedar con los alumnos y firmar el contrato de cesión. Una vez finalizado el curso hay que recuperar estos libros y finalizar la cesión, comprobando que estén bien.

“En este sentido, nos vemos en la obligación de buscar mecanismos más eficientes para la gestión del préstamo de libros”.

## Análisis de posibilidades

### Uso de una hoja Excel / manual

* Es el mecanismo actual y es el que nos ha llevado a determinar que es poco eficiente y debe ser mejorado.

### Uso de una Base de datos y programación de la misma (Access)

Esta opción se ha presentado como factible pero la falta de experiencia en la programación del alumnado hace que esta opción no sea deseable.

“Esta opción será la elegida si no se puede crear la aplicación”.

### Creación de una aplicación propia

Es evidente que esta opción sería la más deseable, pudiendo implementar todos los objetivos que sean necesarios y añadiendo el diseño que se desee. Además, se podría hacer extensible a más institutos en un futuro e incluso presentarse como un proyecto de innovación.

El problema principal que se plantea con esta solución es la seguridad.

## Objetivo

Crear una aplicación web que posibilite el registro del préstamo de libros de forma segura y eficiente a todas las familias del centro y a la vez mejore la gestión de las mismas por parte de la secretaría del centro.

## Requerimientos de la aplicación

La aplicación tendrá tres grandes áreas: **Carga de datos, Gestión de los libros y obtención de documentación**.



|  |  |
| --- | --- |
| **Requisitos funcionales** | |
| **RF 1** | **Carga de datos de materias, cursos y libros al comienzo del curso. En los libros establecer el número de ejemplares de cada uno.** |
| RF 2 | Carga de datos del alumnado |
| RF 3 | Modificación y creación de alumnado |
| RF 4 | Gestión del préstamo: libros al alumno, devolución. |
| RF 5 | Poder establecer el estado de cada libro prestado de forma individual (Sería deseable pero no sé si es imprescindible) |
| RF 6 | Creación e impresión del contrato de préstamos, registro de la firma |
| RF 7 | Búsquedas avanzadas, con filtros avanzados |
| RF 8 | Listados de libros, prestamos, alumnos |
| RF 9 | Copia de seguridad de todos los datos |
| **RF 10** | **Exportación de datos para el curso siguiente** |

## Requerimientos software y hardware

|  |  |
| --- | --- |
| **Software desarrollo** |  |
|  | IDE para desarrollo PHP (PhpStorm) |
|  | Apache 2.4.54 |
|  | PHP 8.2 |
|  | JavaScript |
|  | Bootstrap 5.3 jQuery 3.7  <link https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css  https://code.jquery.com/jquery-3.7.1.slim.min.js |
|  | Mysql 8.2 – MariaDB compatible |
|  | Software Dia de gráficos 0.97 - <https://wireframe.cc/>  o software draw.io <https://app.diagrams.net/> |
| **Software implantación** |  |
|  | Servidor web con cpanel |
|  | Servidor web con PHP 8.2 |
|  | Servidor web con acceso ftp |
|  | Servidor web con Mysql 8.2 |
|  | Un usuario de acceso a Mysql independiente y BBDD propia |
|  | Un usuario ftp al directorio de la aplicación independiente |
| **Hardware** |  |
|  | Hosting externo que cumpla los requisitos de implantación |
|  | 10GB libres en el servidor para subida de archivos |

## Estudio de viabilidad

### Técnica

Los recursos necesarios para este proyecto serán mínimos al ser una única persona la encargada de desarrollarlo. Se debería tener en cuenta los conocimientos necesarios para el desarrollo: Análisis de sistemas, análisis y desarrollo de bases de datos y desarrollo e implantación de aplicaciones web con PHP/JavaScript. Con los recursos actuales de la plantilla se llevarán a cabo.

### Económica

* Sería necesaria una persona a tiempo completo durante todo el proyecto con una duración estimada de 3 meses o a tiempo parcial con una duración estimada de 6 meses.
* Sería imprescindible la compra del software de desarrollo PHPStorm con una licencia profesional.
* Sería imprescindible la compra del hosting de implantación.
* Se debería dedicar varias horas de formación al personal de la empresa para su correcto uso.

### Operativa

Hecho un análisis del funcionamiento actual de la gestión, es muy probable una alta aceptación por parte del personal de la aplicación. Al llevarse a cabo todo el desarrollo dentro del propio instituto permitirá al personal resolver dudas de funcionamiento que hará a la larga su completa implantación.

### Legal

La viabilidad legal está más cuestionada, al necesitar almacenar información de los usuarios en servidores públicos. Según la ley actual deberíamos solicitar permiso a la agencia de protección de datos y dar de alta el fichero. Deberíamos implementar las medidas de seguridad exigidas para un nivel básico y los mecanismos de logs necesarios.

En caso que decidamos proporcionar esta aplicación a otros institutos, se deberá de tener en cuenta este punto y realizar un estudio más detallado.

## Planificación del Proyecto

### Secuenciación de las fases del proyecto



### Planificación de recursos y tiempos



# Análisis

## Casos de uso

### Carga de datos (RF1 RF2)



### Gestión alumnos (RF3)



### Gestión prestamos (RF4 RF6)

****

### Gestión listados (RF7 RF8)



### Preservar los datos (RF9 RF10)



## Descripción de los casos de uso

### Carga de datos

|  |  |
| --- | --- |
| Nombre | **Login** |
| Actores: | Administrador |
| Función: | Da acceso al sistema para poder realizar el resto de acciones. |
| Referencias: | RF - Todos |
|  | |
| Nombre | **Vaciar la BBDD** |
| Actores: | Administrador |
| Función: | Para un correcto funcionamiento la BBDD debe estar vacía de datos, por lo que una vez validado, en este caso se podrá vaciar. |
| Referencias: | RF1 – RF2 |
|  | |
| Nombre | **Cargar la BBDD** |
| Actores: | Administrador |
| Función: | Se cargarán todos los datos necesarios, además del número de ejemplares de cada libro. |
| Referencias: | RF1 – RF2 |
|  | |
| Nombre | **Cargar la BBDD subcaso: Gestion de errores** |
| Actores: | Administrador |
| Función: | Si la carga de datos genera errores, se mostrará un informe y se darán posibles soluciones a los mismos. |
| Referencias: | RF1 – RF2 |
|  | |
| Nombre | **logout** |
| Actores: | Administrador |
| Función: | Salida del sistema. |
| Referencias: | RF Todos |

### Gestión de alumnos

|  |  |
| --- | --- |
| Nombre | **Login** |
| Actores: | Administrador |
| Función: | Da acceso al sistema para poder realizar el resto de acciones. |
| Referencias: | RF - Todos |
|  | |
| Nombre | **Filtrar** |
| Actores: | Administrador |
| Función: | Seleccionar el conjunto de alumnos o alumno sobre el que realizar la modificación, visualización de los datos |
| Referencias: | RF3 |
|  | |
| Nombre | **Filtrar subcaso: Ver datos de alumnos** |
| Actores: | Administrador |
| Función: | Se mostrarán todos los datos del alumno seleccionado incluso los libros asignados. |
| Referencias: | RF3 |
|  | |
| Nombre | **Filtrar subcaso: Modificar datos** |
| Actores: | Administrador |
| Función: | Se podrán cambiar los datos personales del alumno, así como el estado de los libros. NO se podrá cambiar de curso ya que implicaría cambios en la BBDD muy drásticos. |
| Referencias: | RF3 |
|  | |
| Nombre | **logout** |
| Actores: | Administrador |
| Función: | Salida del sistema. |
| Referencias: | RF Todos |

### Gestión de prestamos

|  |  |
| --- | --- |
| Nombre | **Login** |
| Actores: | Administrador |
| Función: | Da acceso al sistema para poder realizar el resto de acciones. |
| Referencias: | RF - Todos |
|  | |
| Nombre | **Buscar alumno** |
| Actores: | Administrador |
| Función: | Seleccionar el conjunto de alumnos o alumno sobre el que realizar la modificación, visualización de los datos |
| Referencias: | RF4 – RF6 |
|  | |
| Nombre | **Buscar alumno subcaso: asignar libros** |
| Actores: | Administrador |
| Función: | Se mostrarán todos los libros del curso y se podrán determinar cuáles se han prestado. |
| Referencias: | RF7 – RF8 |
|  | |
| Nombre | **Buscar alumno subcaso: cerrar préstamos** |
| Actores: | Administrador |
| Función: | Una vez devueltos todos los libros, se cerrará el préstamo para establecer que se tiene derecho al curso siguiente a préstamo |
| Referencias: | RF7 – RF8 |
|  | |
| Nombre | **Buscar alumno subcaso: firmar contrato** |
| Actores: | Administrador |
| Función: | Se podrá imprimir el contrato con el listado de libros y establecer como firmado |
| Referencias: | RF7 – RF8 |
|  | |
| Nombre | **logout** |
| Actores: | Administrador |
| Función: | Salida del sistema. |
| Referencias: | RF Todos |

### Gestión de listados

|  |  |
| --- | --- |
| Nombre | **Login** |
| Actores: | Administrador |
| Función: | Da acceso al sistema para poder realizar el resto de acciones. |
| Referencias: | RF - Todos |
|  | |
| Nombre | **Buscar alumno** |
| Actores: | Administrador |
| Función: | Seleccionar el conjunto de alumnos o alumno sobre el que realizar la modificación, visualización de los datos |
| Referencias: | RF7 – RF8 |
|  | |
| Nombre | **Listados** |
| Actores: | Administrador |
| Función: | Se generarán listados de todo tipo: libros, prestamos, por curso, materias, estado. |
| Referencias: | RF7 – RF8 |
|  | |
| Nombre | **logout** |
| Actores: | Administrador |
| Función: | Salida del sistema. |
| Referencias: | RF Todos |

### Preservar los datos

|  |  |
| --- | --- |
| Nombre | **Login** |
| Actores: | Administrador |
| Función: | Da acceso al sistema para poder realizar el resto de acciones. |
| Referencias: | RF - Todos |
|  | |
| Nombre | **Exportar todos los datos** |
| Actores: | Administrador |
| Función: | Se generará un fichero SQL-92 para preservar los datos. Se dará la posibilidad de exportarlos a csv también. |
| Referencias: | RF9 – RF10 |
|  | |
| Nombre | **Selección de datos** |
| Actores: | Administrador |
| Función: | Se mostrarán todas las tablas para determinar cuáles se van a preservar |
| Referencias: | RF9 – RF10 |
|  | |
| Nombre | **Selección de datos subcaso: Selección de formato** |
| Actores: | Administrador |
| Función: | Se pedirá el formato y sus características para exportar: json, csv, sql-92 |
| Referencias: | RF9 – RF10 |
|  | |
| Nombre | **Selección de datos subcaso: Guardar datos** |
| Actores: | Administrador |
| Función: | Se generará el fichero correspondiente para poder descargar y guardar |
| Referencias: | RF9 – RF10 |
|  | |
| Nombre | **logout** |
| Actores: | Administrador |
| Función: | Salida del sistema. |
| Referencias: | RF Todos |

## Diagrama Conceptual



## Diagrama de clases (diccionario de datos)



## Diagrama de actividades

### Generales





### Carga de datos





### Gestión de alumnos

### Gestión de préstamos



### Gestión de listados



### Preservar datos



## Esquema Entidad / Relación

Se ha determinado que no es necesaria una tabla de configuraciones, permitiendo que en diseño se tomen las alternativas más eficientes para este caso (es recomendable que se cree un fichero con los datos de configuración).



## Modelo relacional y normalización

Una vez llevada a cabo la normalización, teniendo en cuenta hasta la forma de Boyce-Cood, el resultado es el siguiente:

**Materias**: id, nombre, departamento

**Cursos**: curso, nivel

**Libros**: isbn, titulo, autor, numero\_ejemplares, id\_materia, id\_curso

**Alumnos**: id, nombre, apellidos, tramo (I, II, Ninguno), bilingüe (S/N)

**AlumnosCursosLibros**: isbn, id\_alumno, id\_curso, fecha\_entrega, fecha\_devolucion, estado (entregado, por\_devolver)

# Diseño del Proyecto

## Diseño físico de la Base de Datos

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **materias** | | | | | | | |
| **Nombre campo** | **Tipo** | **Tamaño** | **PK** | **FK** | **NONULL** | **Valor defecto** | **Observaciones** |
| **id** | **INT** | **11** | **X** |  | **X** |  | **AUTONUMÉRICO** |
| **nombre** | **VARCHAR** | **255** |  |  | **X** |  |  |
| **departamento** | **VARCHAR** | **255** |  |  | **X** |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **cursos** | | | | | | | |
| **Nombre campo** | **Tipo** | **Tamaño** | **PK** | **FK** | **NONULL** | **Valor defecto** | **Observaciones** |
| **curso** | **VARCHAR** | **20** | **X** |  | **X** |  | **Letra del niev** |
| **nivel** | **VARCHAR** | **20** |  |  | **X** |  | **1ºeso, 2eso….** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **libros** | | | | | | | |
| **Nombre campo** | **Tipo** | **Tamaño** | **PK** | **FK** | **NONULL** | **Valor defecto** | **Observaciones** |
| **isbn** | **VARCHAR** | **20** | **X** |  | **X** |  | **Validar formato isbn** |
| **titulo** | **VARCHAR** | **255** |  |  | **X** |  |  |
| **autor** | **VARCHAR** | **255** |  |  | **X** |  |  |
| **numero\_ejemplares** | **INT** | **5** |  |  | **X** |  |  |
| **id\_materia** | **INT** | **11** |  | **X** | **X** |  | **Tabla materias** |
| **id\_curso** | **VARCHAR** | **20** |  | **X** | **X** |  | **Tabla cursos** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **alumnos** | | | | | | | |
| **Nombre campo** | **Tipo** | **Tamaño** | **PK** | **FK** | **NONULL** | **Valor defecto** | **Observaciones** |
| **nie** | **VARCHAR** | **10** | **X** |  | **X** |  | **Número Identificación Escolar** |
| **nombre** | **VARCHAR** | **255** |  |  | **X** |  |  |
| **apellidos** | **VARCHAR** | **255** |  |  | **X** |  |  |
| **tramo** | **VARCHAR** | **1** |  |  | **X** |  | **0, I, II (0 para no beca, tramo I y II)** |
| **bilingue** | **BOOL** |  |  |  | **X** | **False** |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **alumnoscursoslibros** | | | | | | | |
| **Nombre campo** | **Tipo** | **Tamaño** | **PK** | **FK** | **NONULL** | **Valor defecto** | **Observaciones** |
| **nie** | **VARCHAR** | **10** | **X** | **X** | **X** |  | **Tabla alumnos** |
| **curso** | **VARCHAR** | **20** | **X** | **X** | **X** |  | **Tabla cursos** |
| **isbn** | **VARCHAR** | **20** | **X** | **X** | **X** |  | **Tabla libros** |
| **fecha\_entrega** | **DATE** |  |  |  | **X** | **Hora y día actual** |  |
| **fecha\_devolución** | **DATE** |  |  |  | **X** |  |  |
| **estado** | **VARCHAR** |  |  |  |  |  | **P: Prestado, D:Devuelto** |

## Estructura de almacenamiento

El programa está estructurado en **múltiples archivos .py**, cada uno de ellos contiene una clase responsable de una parte específica de la lógica del sistema. Esta modularización permite mantener un código más limpio, organizado y fácil de mantener.

El almacenamiento de los datos se realiza mediante **archivos de texto plano (.txt)**, que funcionan como una **base de datos simple** adaptada a las necesidades del programa. Este sistema es suficiente para gestionar la información de un único centro educativo sin requerir bases de datos complejas.

/Programa/

|

|----/Clases/

| |---- alumnos.py

| |---- libros.py

| |---- prestamos.py

| |---- cursos.py

| |---- materias.py

| └----main.py

|

|----/Datos/

| |---- alumnos.txt

| |---- libros.txt

| |---- cursos.txt

| |---- materias.py

| └---- prestamos.txt

|

|----/Contratos/

| └---- Contrato\_<NIE>.txt

|

|----/Usuarios/

└---- usuariostxt

## Diseño de procedimientos especiales

### Configuración de la aplicación.

La base de datos del programa está organizada y distribuida en ficheros de texto plano (.txt). Dado que se trata de una aplicación destinada a la gestión de préstamos de libros en un centro educativo concreto, no se requiere una base de datos compleja ni de gran escala.

Los datos se dividen en dos grupos:

**Datos editables desde la aplicación:**

* alumnos.txt
* prestamos.txt

Estos ficheros permiten **crear, modificar y visualizar información** directamente desde el programa, ya que son elementos que cambian con frecuencia durante el curso escolar.

**Datos editables de forma manual:**

* libros.txt
* cursos.txt
* materias.txt
* usuarios.txt

Estos archivos contienen información más **estructural y estable** (como el catálogo de libros o los cursos y materias del centro). Debido a que rara vez cambian —normalmente solo se modifican entre cursos o tras cambios normativos—, su mantenimiento se realiza **manualmente fuera del programa**. Sin embargo, **sí pueden ser consultados** desde la aplicación en modo solo lectura.

**Los directorios /Contratos y /Usuarios contienen ficheros especiales:**

|----/Contratos/

| └---- Contrato\_(Nº de documento).txt

|

|----/Usuarios/

└---- usuarios.txt

* **/Contratos/:** Guarda archivos contrato\_<NIE>.txt generados al registrar un préstamo. Su función es dejar constancia por escrito de que el alumno ha recibido el libro y acepta devolverlo, mediante firma física.
* **/Usuarios/:** Contiene usuarios.txt, con usuarios y contraseñas autorizados. El programa lo consulta al iniciar para validar el acceso.

### Estructura del fichero de datos

Cada fichero .txt almacena los distintos **atributos asociados a una clase** del programa, y por tanto contiene los **datos que forman parte de cada objeto creado**.

/Datos/

│

├── alumnos.txt 🡪 Información de cada alumno (DNI, nombre, apellidos, etc.)

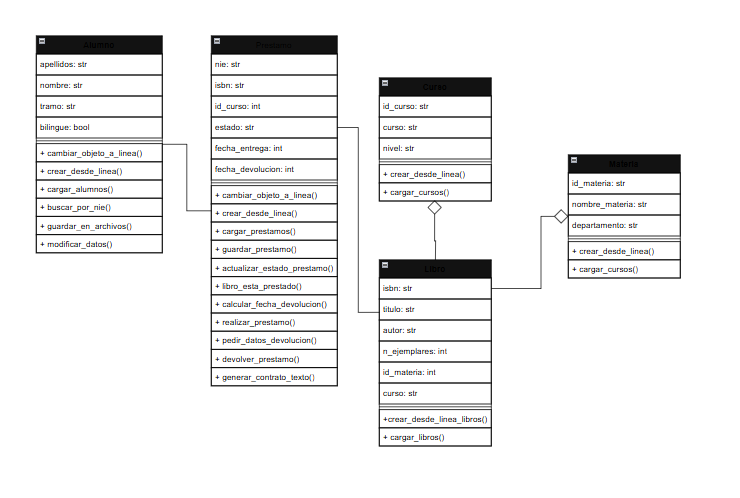
├── prestamos.txt 🡪 Registros de los préstamos realizados.

├── libros.txt 🡪 Catálogo de libros disponibles.

├── cursos.txt 🡪 Lista de cursos del centro educativo.

└── materias.txt 🡪 Materias asociadas a los cursos.

## Diseño de la estructura de clases y librerías (diagrama de clases)



## Diseño de la interfaz gráfica

### Diseño de movimiento de ventanas

El diseño se ha pensado en facilitar el trabajo de los dos tipos de usuarios teniendo en cuenta un nivel bajo de conocimientos de informática. En este caso se recomiendan recorridos lineales y guiados para evitar los errores. Toda ventana permitirá la vuelta atrás del proceso para ayudar al usuario.

1. Gestionar alumnos.

1. Crear alumno.

2. Modificar alumno.

1. Modificar NIE del alumno.

2. Modificar nombre del alumno.

3. Modificar apellidos del alumno.

4. Modificar bilingüe.

5. Modificar tramo becado.

6. Salir de menú 'Modificar alumno'.

3. Mostrar información de alumno.

4. Mostrar información de alumnos.

5. Salir al menú principal.

2. Gestionar préstamos.

1. Realizar préstamo.

2. Devolución préstamo.

3. Consultar estado libro.

4. Salir al menú principal.

3. Mostrar datos.

1. Búsqueda avanzada.

1. Buscar libro.

2. Buscar curso.

3. Buscar materia.

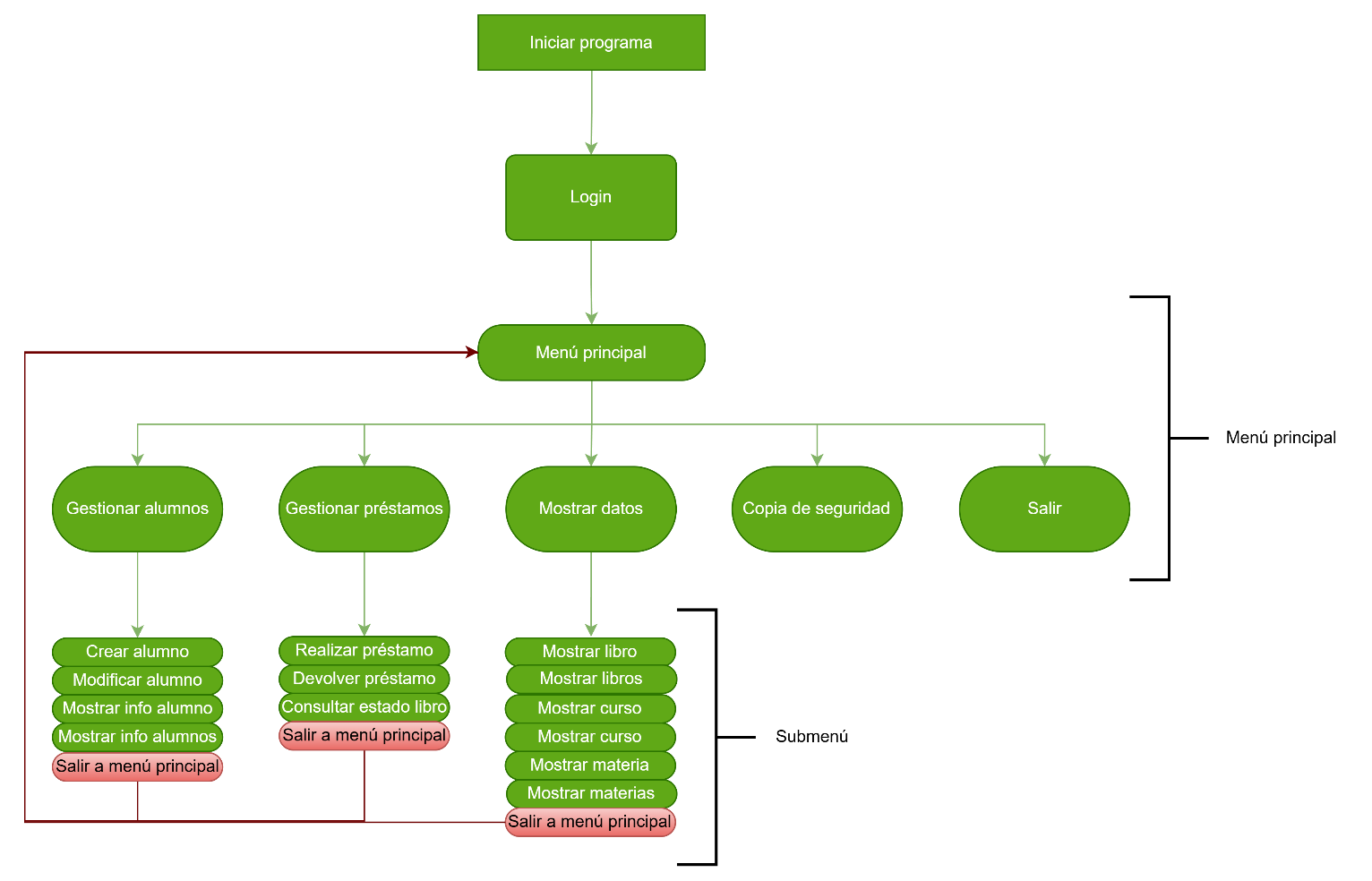
1. Mostrar libros.

2. Mostrar cursos.

3. Mostrar materias.

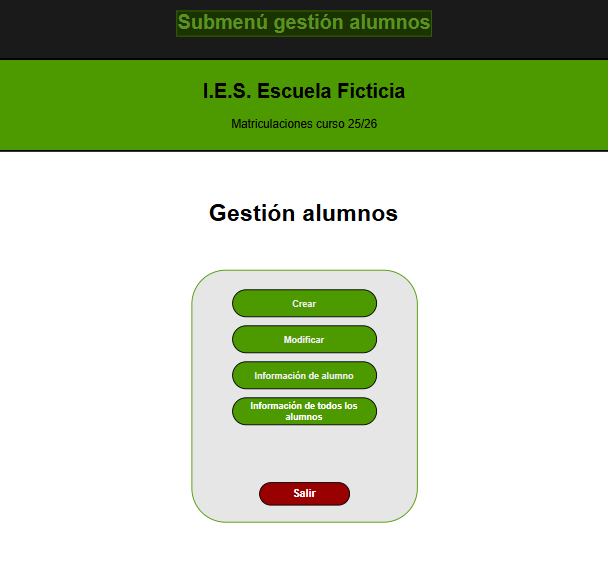
4. Salir al menú principal.

4. Copia de seguridad.

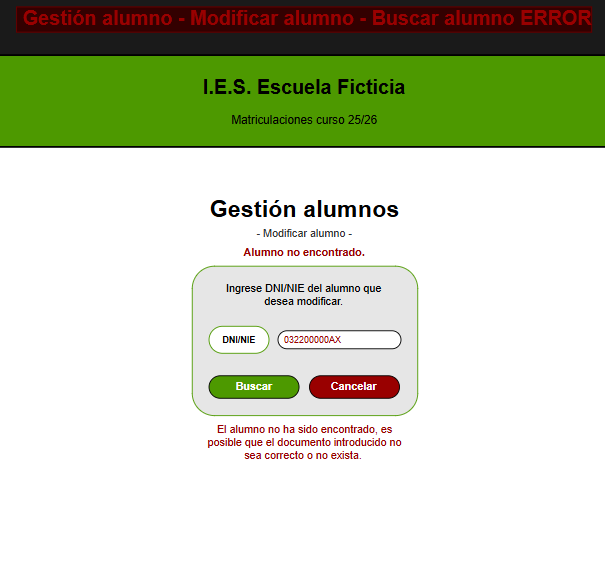
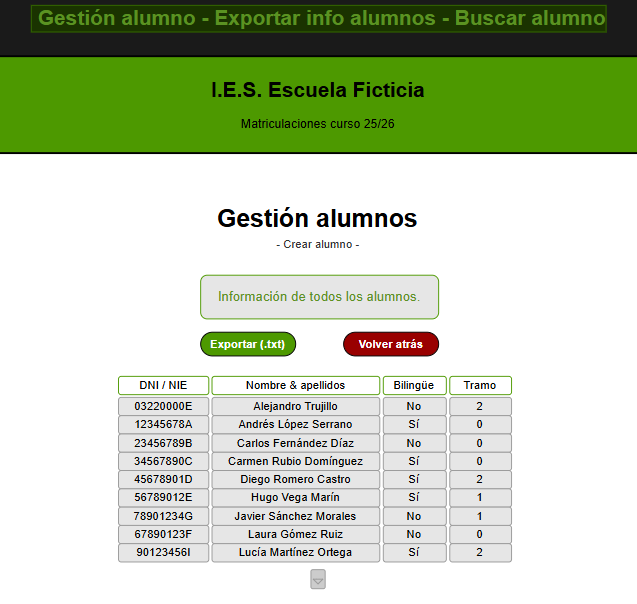
5. Salir.

### Diseño de las ventanas

#### Ventana : Iniciar sesión / login Ventana : Menú principal

*Ventana : Gestión alumnos* *Ventana : Gestión alumnos – Crear alumnos*

*Ventana : Modificar alumno - ERROR Ventana : Información alumnos*



Se han añadido a la documentación seis diseños de ventana que permiten visualizar cómo podría ser la interfaz final del programa. Estos esquemas sirven para representar el flujo de trabajo de la aplicación y ayudan a imaginar de forma clara y ordenada la interacción del usuario con el sistema.

# Implementar el proyecto

### Sql de creación (MySQL)

-- phpMyAdmin SQL Dump

-- version 5.2.0

-- https://www.phpmyadmin.net/

--

-- Servidor: 127.0.0.1

-- Tiempo de generación: 24-06-2024 a las 18:56:50

-- Versión del servidor: 10.4.27-MariaDB

-- Versión de PHP: 8.2.0

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Base de datos: `bancolibros`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `alumnos`

--

CREATE TABLE `alumnos` (

`nie` varchar(10) NOT NULL,

`nombre` varchar(255) NOT NULL,

`apellidos` varchar(255) NOT NULL,

`tramo` varchar(2) NOT NULL DEFAULT '0' COMMENT '0 nada, I tramo I y II Tramo 2',

`bilingue` tinyint(1) NOT NULL DEFAULT 1 COMMENT '0 True, 1 false'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish2\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `alumnoscrusoslibros`

--

CREATE TABLE `alumnoscrusoslibros` (

`nie` varchar(10) NOT NULL,

`curso` varchar(255) NOT NULL,

`isbn` varchar(20) NOT NULL,

`fecha\_entrega` date NOT NULL,

`fecha\_devolucion` date NOT NULL,

`estado` varchar(1) NOT NULL DEFAULT 'P' COMMENT 'P: Prestado, D:Devuelto'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish2\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `cursos`

--

CREATE TABLE `cursos` (

`curso` varchar(20) NOT NULL,

`nivel` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish2\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `libros`

--

CREATE TABLE `libros` (

`isbn` varchar(20) NOT NULL,

`titulo` varchar(255) NOT NULL,

`autor` varchar(255) NOT NULL,

`numero\_ejemplares` int(5) NOT NULL,

`id\_materia` int(11) NOT NULL,

`id\_curso` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish2\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `materias`

--

CREATE TABLE `materias` (

`id` int(11) NOT NULL,

`nombre` varchar(255) NOT NULL,

`departamento` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish2\_ci;

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `alumnos`

--

ALTER TABLE `alumnos`

ADD PRIMARY KEY (`nie`);

--

-- Indices de la tabla `alumnoscrusoslibros`

--

ALTER TABLE `alumnoscrusoslibros`

ADD PRIMARY KEY (`nie`,`curso`,`isbn`),

ADD KEY `isbn` (`isbn`),

ADD KEY `curso` (`curso`);

--

-- Indices de la tabla `cursos`

--

ALTER TABLE `cursos`

ADD PRIMARY KEY (`curso`);

--

-- Indices de la tabla `libros`

--

ALTER TABLE `libros`

ADD PRIMARY KEY (`isbn`),

ADD KEY `FK\_Materias` (`id\_materia`),

ADD KEY `FK\_Cursos` (`id\_curso`) USING BTREE;

--

-- Indices de la tabla `materias`

--

ALTER TABLE `materias`

ADD PRIMARY KEY (`id`);

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `alumnoscrusoslibros`

--

ALTER TABLE `alumnoscrusoslibros`

ADD CONSTRAINT `alumnoscrusoslibros\_ibfk\_1` FOREIGN KEY (`isbn`) REFERENCES `libros` (`isbn`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `alumnoscrusoslibros\_ibfk\_2` FOREIGN KEY (`nie`) REFERENCES `alumnos` (`nie`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `alumnoscrusoslibros\_ibfk\_3` FOREIGN KEY (`curso`) REFERENCES `cursos` (`curso`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Filtros para la tabla `libros`

--

ALTER TABLE `libros`

ADD CONSTRAINT `fk\_libros\_cursos` FOREIGN KEY (`id\_curso`) REFERENCES `cursos` (`curso`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `fk\_libros\_materias` FOREIGN KEY (`id\_materia`) REFERENCES `materias` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

## Indicadores de calidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Usabilidad** | | | | |
| **Categoría** | **Parámetro** | **Indicador** | **Fuente** | **Resultado** |
| Navegabilidad | ¿Navegación intuitiva? | Se organiza en secciones | Encuesta | Si No |
|  | Se accede de forma rápida | Encuesta | Si No |
| ¿Dispone de menú? | Tiene un menú navegable | Encuesta | Si No |
|  | Dispone de mapa | Encuesta | Si No |
| ¿Compatible edge? | Las imágenes se cargan | Encuesta | Si No |
|  | Se visualiza de forma correcta | Encuesta | Si No |
| ¿Compatible chrome? | Las imágenes se cargan | Encuesta | Si No |
|  | Se visualiza de forma correcta | Encuesta | Si No |
| ¿Compatible Firefox? | Las imágenes se cargan | Encuesta | Si No |
|  | Se visualiza de forma correcta | Encuesta | Si No |
| ¿Compatible Safari? | Las imágenes se cargan | Encuesta | Si No |
|  | Se visualiza de forma correcta | Encuesta | Si No |
| Velocidad | ¿Velocidad de carga? | Tamaño medio de la página | Aplicación | \_\_Kb |
|  | Tiempo hasta la carga completa | Aplicación | \_\_milisegundos |
| Diseño general | ¿Diseño? | Simplicidad | Encuesta | 0 a 10 |
|  |  | Claridad | Encuesta | 0 a 10 |
|  |  | Coherencia | Encuesta | 0 a 10 |
|  |  | Facilidad de uso | Encuesta | 0 a 10 |
| **Disponibilidad** | | | | |
| **Categoría** | **Parámetro** | **Indicador** | **Fuente** | **Resultado** |
| Visualización en diversos dispositivos | ¿Funciona con móviles? | Funciona el texto | Encuesta | Si No |
|  | Funciona las búsquedas | Encuesta | Si No |
|  | Funcionan las acciones de los botones | Encuesta | Si No |
| ¿Funciona con táblets? | Funciona el texto | Encuesta | Si No |
|  | Funciona las búsquedas | Encuesta | Si No |
|  | Funcionan las acciones de los botones | Encuesta | Si No |
| Disponibilidad | ¿Tiempo disponible? | Media de disponibilidad | Encuesta | \_ porcentaje |
| **Analítica** | | | | |
| **Categoría** | **Parámetro** | **Indicador** | **Fuente** | **Resultado** |
| Audiencia | ¿Número de usuarios? | Conexiones por día | Aplicación | \_ Conexiones |
|  | Duración media de la sesión | Aplicación | \_segundos |
| **Funcionamiento** | | | | |
| **Categoría** | **Parámetro** | **Indicador** | **Fuente** | **Resultado** |
| Satisfacción | ¿Satisfacción general? | Valoración | Encuesta | 0 a 10 |
|  | Funcionamiento general | Encuesta | 0 a 10 |
|  | Disponibilidad | Encuesta | 0 a 10 |

## Elaboración de una batería de pruebas para detectar errores

La batería de pruebas que se muestran a continuación se deberá realizar en al menos los navegadores Chrome y Firefox, siendo recomendable que se hicieran también en Edge y Safari tanto en escritorio Windows, Mac y Linux como en Android y Apple.

Para comprobar el correcto funcionamiento de la aplicación, se han realizado pruebas unitarias sobre las distintas clases que la componen, utilizando el módulo *unittest* de Python.

Se han aplicado los siguientes tipos de pruebas:

* **Pruebas unitarias**: sobre métodos individuales.
* **Pruebas de caja blanca**: comprobación directa del comportamiento del código.
* **Caja negra**: verificación de entrada/salida esperada.
* **Pruebas automáticas**: sin intervención manual.

**Los resultados los dejamos reflejados en el apartado *Datos de prueba*.**

## Evaluación y solución de incidencias

**Validación de datos**

Durante el desarrollo del proyecto se han implementado varias validaciones básicas y estructurales para garantizar que los datos manejados por la aplicación sean coherentes y seguros. A continuación, se resumen las principales validaciones aplicadas:

* **Verificación de existencia de archivos**: al cargar datos desde archivos .txt, se comprueba si existen usando try/except. Si no se encuentra un archivo, se muestra un mensaje informativo y se prevé la creación del archivo en blanco.
* **Conversión segura de datos**: al leer información desde texto, se aplican conversiones seguras:
  + El campo bilingüe se transforma correctamente a booleano (True o False) desde un número (0 o 1).
  + Los datos leídos de archivo se dividen con split("|") y se limpian con .strip() para evitar errores por espacios vacíos.
* **Control de búsqueda**: cuando se busca un alumno por su NIE, si no se encuentra, se devuelve None, lo que permite manejar el flujo del programa de forma segura (por ejemplo, mostrar un mensaje o no continuar la operación).

**Validaciones recomendadas (mejoras futuras)**

* Comprobar que los campos no estén vacíos antes de guardar (nombre, NIE, etc.)
* Validar que el NIE tenga 9 caracteres y acabe en una letra.
* Verificar que el ISBN del libro tenga 10 o 13 dígitos y sea numérico.
* Usar formato de fecha válido (YYYY-MM-DD) al introducir préstamos.

Estas validaciones pueden ampliarse fácilmente añadiendo funciones específicas o controles antes de guardar los datos.

**Ejemplo de validaciones en menú:**

def gestionar\_alumnos(self):

while True:

print(self.submenu\_gestionar\_alumnos())

opcion = self.pedir\_opcion("Escoja una opción: ")

match opcion:

case 1:

self.crear\_alumno()

case 2:

nie = input("Ingrese NIE del alumno a modificar: ").strip().upper()

alumno = Alumno.buscar\_por\_nie(nie)

if alumno:

self.modificar\_alumno(alumno)

else:

print("Alumno no encontrado.")

case 3:

nie = input("Ingrese NIE: ")

alumno = Alumno.buscar\_por\_nie(nie)

if alumno:

print(alumno)

else:

print("Alumno no encontrado.")

case 4:

alumnos = Alumno.cargar\_alumnos()

if not alumnos:

print("No hay alumnos registrados.")

else:

for alumno in alumnos:

print(alumno)

case 5:

break

case \_:

print("Opción inválida.")

## Evaluación y seguimiento del proyecto

Durante el desarrollo del proyecto se ha llevado a cabo un proceso progresivo y estructurado que ha permitido cumplir con los objetivos establecidos. La planificación se ha dividido en varias fases: análisis del problema, diseño de clases, desarrollo del código, pruebas unitarias y documentación.

A lo largo del proceso se han seguido buenas prácticas de programación, como la modularización del código, la reutilización de funciones, la separación de responsabilidades y la lectura/escritura de datos a través de archivos. Esto ha facilitado tanto el mantenimiento del programa como la implementación de nuevas funcionalidades.

Se ha llevado un seguimiento continuo del proyecto a través de pruebas frecuentes, lo que ha permitido detectar errores rápidamente y solucionarlos antes de que afectaran a otras partes del sistema. También se ha mejorado la robustez del programa añadiendo validaciones para controlar posibles errores en la entrada de datos y prevenir caídas del sistema.

El uso de pruebas automáticas con *unittest* ha sido fundamental para verificar el comportamiento correcto de las clases de forma sistemática y sin depender de pruebas manuales.

En general, el proyecto ha sido una oportunidad práctica para aplicar los conocimientos adquiridos en programación, entender el trabajo estructurado y simular un entorno de desarrollo real.

# Implantación del Proyecto

## Plan de implantación

Dado que se trata de una aplicación sencilla de escritorio orientada a un único centro educativo, la implantación no requiere procesos complejos ni paradas técnicas.

La instalación se realizará directamente en el equipo o equipos del personal administrativo encargado de gestionar los préstamos, con una puesta en marcha inmediata tras una configuración mínima.

**Tabla: Plan de implantación**

|  |  |  |  |
| --- | --- | --- | --- |
| Etapa | Actividades | Duración estimada | Horario recomendado |
| Instalación | Copia del programa y los ficheros .txt en el sistema. | 1 día | Mañana |
| Configuración | Revisión de rutas, permisos de lectura/escritura. Entrega de usuarios. | 1 día | Mañana |
| Formación | Explicación del uso al personal responsable y entrega de manual de usuario. | 1 día | Mañana o en horario libre |
| Pruebas | Alta de alumnos y pruebas de préstamo simuladas. | 1 día | Mañana |

## Manual de instalación

### Instalación de la aplicación

1. Descargar el proyecto completo en formato .zip o clonado desde repositorio (si aplica).
2. Extraer o ubicar la carpeta en una ubicación accesible, por ejemplo:  
   C:\Centro\GestorLibros\Programa\
3. Verificar que se tenga instalada una versión de **Python 3.11 o superior**.
4. Ejecutar el archivo main.py dentro de la carpeta Clases desde terminal:

*python Clases/main.py*

1. Asegurarse de que la carpeta Datos/ existe al mismo nivel que Clases/, y contiene los archivos:
   * alumnos.txt
   * prestamos.txt
   * libros.txt
   * cursos.txt
   * materias.txt

⚠️ En caso de que el archivo alumnos.txt o prestamos.txt **no existan**, el programa los **creará automáticamente** al guardar el primer dato correspondiente, iniciando con cero registros.

## Mejoras posibles

Aunque la aplicación cumple con los requisitos básicos de funcionamiento, se podrían implementar en el futuro las siguientes mejoras para aumentar su funcionalidad y facilidad de uso:

* Crear una **interfaz gráfica** para facilitar el uso al personal no técnico. (Tkinter o PyQt)
* Incorporar **distintos niveles de usuario** (administrador, solo lectura, etc.).
* Mejorar el sistema de login con:
* Encriptación de contraseñas.
* Registro de accesos.
* Permitir **cambio de contraseña** desde el programa.
* Base de datos más robusta (SQLite, MySQL).
* **Informes en PDF o Excel**.
* Gestión de historial de préstamos.
* Alertas por fecha de devolución.

## Manual de usuario

Este manual describe cómo utilizar la aplicación desde el punto de vista del usuario responsable de gestionar préstamos de libros en el centro educativo.

#### Acceso al sistema (login)

Al iniciar el programa, el usuario deberá **identificarse** introduciendo su **nombre de usuario y contraseña**.

Estos datos están registrados en el archivo usuarios.txt, el cual contiene las credenciales necesarias para acceder al sistema.

* Si las credenciales son correctas, el usuario accede al menú principal.
* Si son incorrectas, se mostrará un mensaje de error y se volverá a solicitar el login.

**Formato del archivo usuarios.txt:**

*admin;1234*

*usuario;clave123*

⚠️ Por seguridad, no se permiten espacios ni campos vacíos. La contraseña no se muestra en pantalla mientras se escribe.

#### Iniciar el programa

1. Abrir una terminal o ejecutar el archivo main.py.
2. Introducir usuario y contraseña cuando se solicite.
3. Si el acceso es correcto, se muestra el menú principal.

**Menú principal**

* **Gestionar alumnos**: alta, modificación y consulta de alumnos.
* **Gestionar préstamos**: asignar o devolver libros.
* **Mostrar datos**: consultar libros, cursos y materias.
* **Copia de seguridad:** duplicado de todos los ficheros .txt.
* **Salir**: cierra el programa.

**Submenú - Gestión de alumnos**

* **Crear:** añade un nuevo alumno con todos sus datos.
* **Modificar:** permite cambiar los datos de un alumno existente.
* **Mostrar información de alumno (búsqueda avanzada):** visualizar un alumno.
* **Mostrar información de alumnos:** visualizar un listado completo de todos los alumnos.

**Submenú - Mostrar datos**

* **Libros**: disponibles solo en modo consulta (no editables desde el programa). Se puede buscar por ISBN o título, o ver el listado completo.
* **Cursos:** disponibles solo en modo consulta (no editables desde el programa).
* **Materias**: disponibles solo en modo consulta (no editables desde el programa).

# Anexos

## Formulario de contrato usado



## Fichero exportado de Delphos (alumnos)



CSV separado por comas (,) cualificado el nombre por comillas dobles (“). Los campos son:

Alumno (Apellidos, Nombre), NumeroEscolar, NumeroSolicitud, FechaSolicitud, CampoNoNecesario, CursoAcadémico, CampoNoNecesario, CampoNoNecesario, CampoNoNecesario, CampoNoNecesario, CampoNoNecesario, CampoNoNecesario, CampoNoNecesario, CampoNoNecesario, ResultadoLibros, ResultadoComedor, Matriculado, TipoBecaLibros, TipoBecaComedor

## Ejemplo de formulario de satisfacción

Fecha:

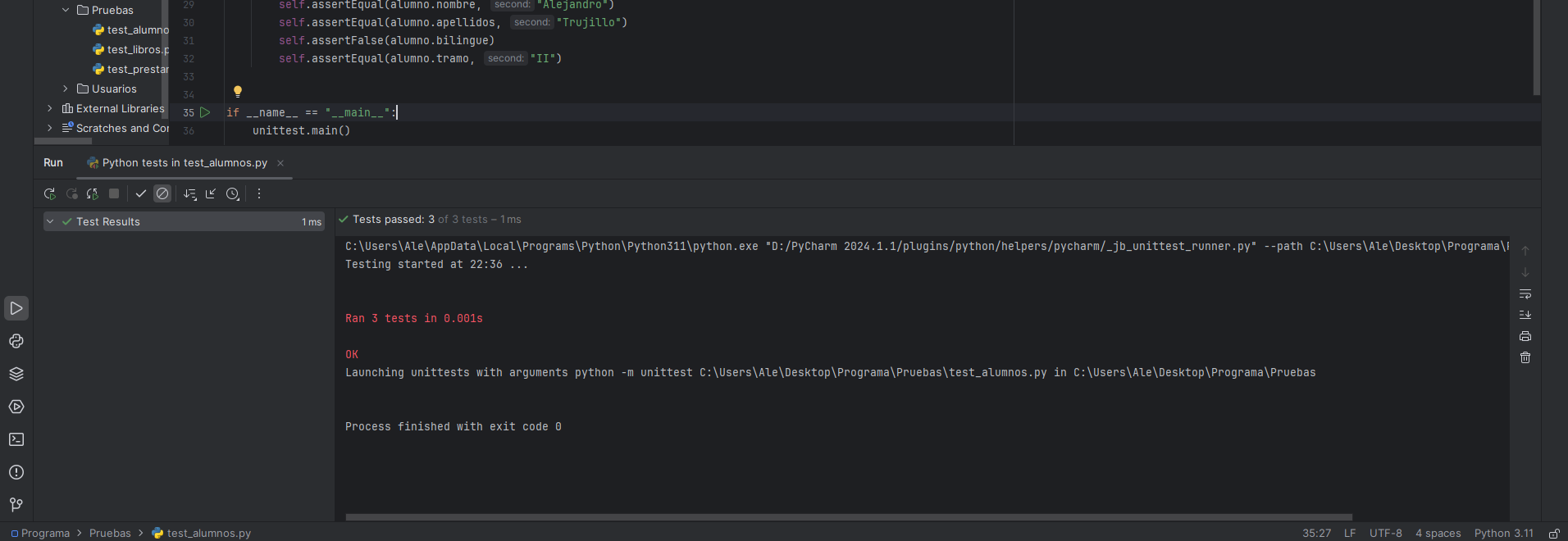
|  |  |  |
| --- | --- | --- |
| **Indicador** |  | |
| Se organiza en secciones | Si | No |
| Se accede de forma rápida | Si | No |
| Tiene un menú navegable | Si | No |
| Dispone de mapa | Si | No |
| Las imágenes se cargan | Si | No |
| Se visualiza de forma correcta | Si | No |
| Las imágenes se cargan | Si | No |
| Se visualiza de forma correcta | Si | No |
| Las imágenes se cargan | Si | No |
| Se visualiza de forma correcta | Si | No |
| Las imágenes se cargan | Si | No |
| Se visualiza de forma correcta | Si | No |
| Funciona el texto | Si | No |
| Funciona las búsquedas | Si | No |
| Funcionan las acciones de los botones | Si | No |
| Funciona el texto | Si | No |
| Funciona las búsquedas | Si | No |
| Funcionan las acciones de los botones | Si | No |
|  | | | | | | | | | | | |
| Valoración | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Funcionamiento general |  |  |  |  |  |  |  |  |  |  |  |
| Disponibilidad |  |  |  |  |  |  |  |  |  |  |  |
| Simplicidad |  |  |  |  |  |  |  |  |  |  |  |
| Claridad |  |  |  |  |  |  |  |  |  |  |  |
| Coherencia |  |  |  |  |  |  |  |  |  |  |  |
| Facilidad de uso |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | |
| Comentarios: | | | | | | | | | | | |
|  | | | | | | | | | | | |

## Datos de prueba

**Clase Alumno**

A continuación, se presenta una tabla con algunos de los casos de prueba aplicados a la clase Alumno.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre prueba | Archivo | Método probado | Parámetros | Valor esperado | Valor obtenido | Resultado |
| test\_crear\_objeto | test\_alumnos.py | \_\_init\_\_ | "03220367E", "Alejandro", "Trujillo", False, "II" | Atributos correctamente asignados | OK | Correcto |
| test\_cambiar\_objeto\_a\_linea | test\_alumnos.py | cambiar\_objeto\_a\_linea | Sin parámetros (usa self) | "03220367E|Alejandro|Trujillo|0|II" | OK | Correcto |
| test\_crear\_desde\_linea | test\_alumnos.py | crear\_desde\_linea | "03220367E|Alejandro|Trujillo|0|II" | Objeto Prestamo con mismos datos | OK | Correcto |



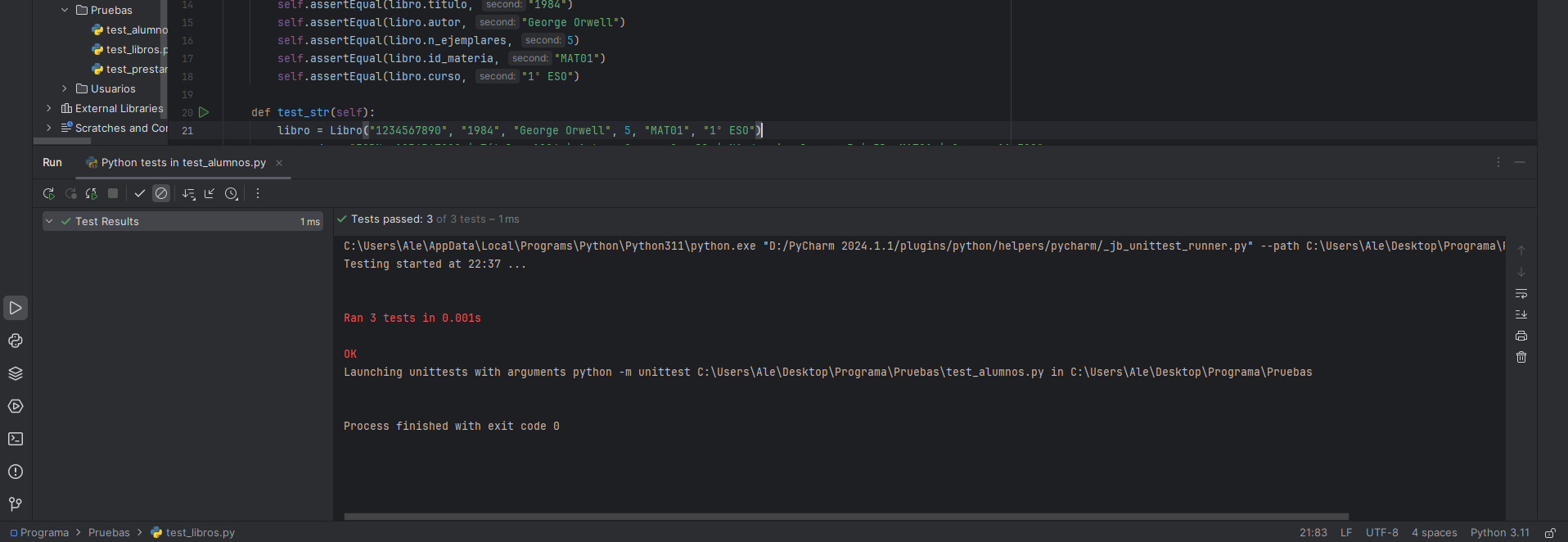
Código de prueba unitaria con *unittest*.

Resultado correcto de pruebas automáticas.

**Clase Libro**

A continuación, se presenta una tabla con algunos de los casos de prueba aplicados a la clase Libro.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre prueba | Archivo | Método probado | Parámetros | Valor esperado | Valor obtenido | Resultado |
| test\_crear\_libro | test\_libros.py | \_\_init\_\_ | "1234567890", "1984", "George Orwell", 5, "MAT01", "1º ESO" | Atributos correctamente asignados | OK | Correcto |
| test\_str | test\_libros.py | \_\_str\_\_ | Sin parámetros (usa self) | Texto con toda la información del libro | OK | Correcto |
| test\_crear\_desde\_linea\_libros | test\_libros.py | crear\_desde\_linea\_libros | "1234567890|1984|George Orwell|5|MAT01|1º ESO" | Objeto Libro con mismos datos | OK | Correcto |



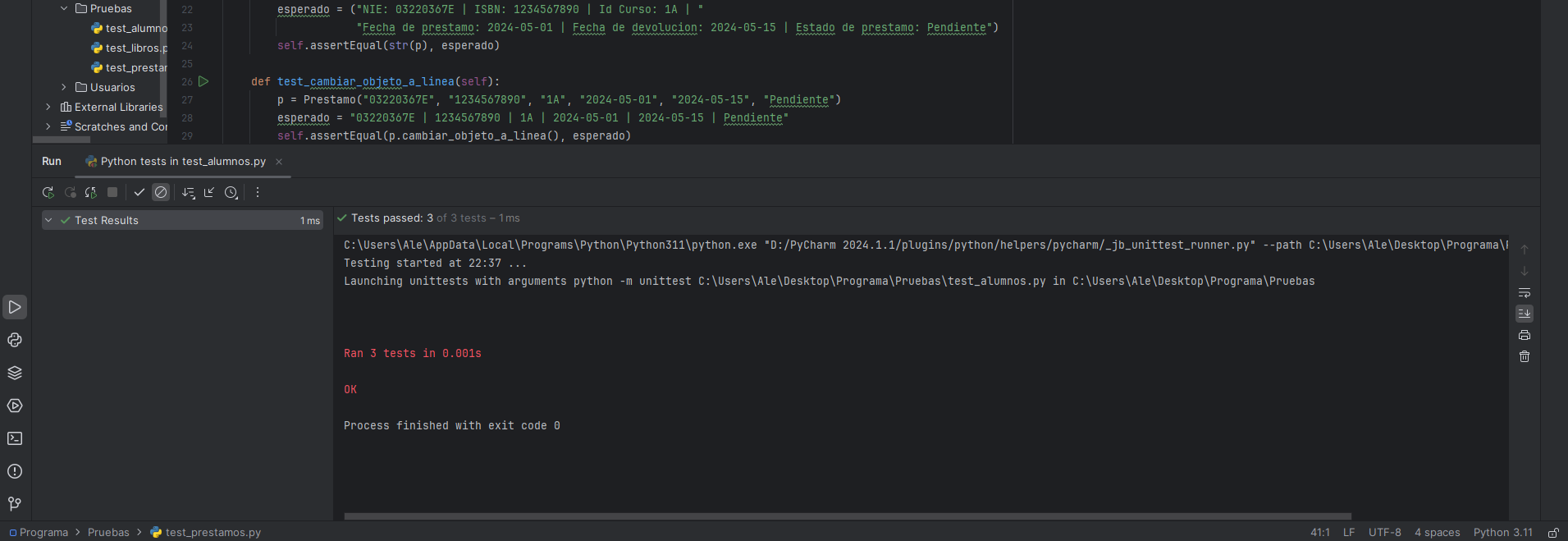
Código de prueba unitaria con *unittest*.

Resultado correcto de pruebas automáticas.

**Clase Prestamo**

A continuación, se presenta una tabla con algunos de los casos de prueba aplicados a la clase Alumno.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre prueba | Archivo | Método probado | Parámetros | Valor esperado | Valor obtenido | Resultado |
| test\_crear\_prestamo | test\_prestamos.py | \_\_init\_\_ | "03220367E", "1234567890", "1A", "2024-05-01", "2024-05-15", "Pendiente" | Atributos correctamente asignados | OK | Correcto |
| test\_str | test\_prestamos.py | \_\_str\_\_ | Sin parámetros (usa self) | Texto con toda la información del préstamo | OK | Correcto |
| test\_cambiar\_objeto\_a\_linea | test\_prestamos.py | cambiar\_objeto\_a\_linea | Sin parámetros (usa self) | "03220367E | 1234567890 | 1A | 2024-05-01 | 2024-05-15 | Pendiente" | OK | Correcto |
| test\_crear\_desde\_linea | test\_prestamos.py | crear\_desde\_linea | "03220367E | 1234567890 | 1A | 2024-05-01 | 2024-05-15 | Pendiente" | Objeto Prestamo con mismos datos | OK | Correcto |



Código de prueba unitaria con *unittest*.

Resultado correcto de pruebas automáticas.

Como se observa en la salida del test, todos los casos se han ejecutado correctamente sin errores, lo que indica que el comportamiento de las clases es el esperado.