

# SOFTWARE

## CICLO DE VIDA DE UN SOFTWARE



**Álvaro Gracia Aledo**

2/12/2023

1º DAM

# ÍNDICE

1. ANÁLISIS DE REQUISITOS.....	3
• ANÁLISIS FUNCIONALES.....	3
• ANÁLISIS NO FUNCIONALES.....	4
2. DISEÑO.....	4
2.1 ENTIDADES.....	4
2.2 SISTEMA GESTOR DE BASES.....	5
.....	5
2.3 LENGUAJE DE PROGRAMACIÓN.....	5
2.4 HARDWARE.....	5
2.5 CICLO DE VIDA.....	5
3. CODIFICACIÓN.....	6
3.1 LENGUAJE DE PROGRAMACIÓN.....	6
3.2 HERRAMIENTAS.....	6
3.3 PASOS DE CODIFICACIÓN.....	6
4. PRUEBAS.....	7
4.1 PRUEBAS UNITARIAS.....	7
4.2 PRUEBAS INTEGRADAS.....	7
5. DOCUMENTACIÓN.....	7
6. EXPLOTACIÓN.....	8
7. GESTIÓN DE MANTENIMIENTO.....	8
8. PLANNING.....	8
9. BIBLIOGRAFÍA.....	9

La empresa BK ha recibido un encargo de software. Se trata de diseñar una aplicación para llevar a cabo la gestión de notas de un centro de estudios formado por una cooperativa de profesores. El centro desea trabajar con software libre.

## 1. ANÁLISIS DE REQUISITOS.

### • ANÁLISIS FUNCIONALES.

- Consultar notas y datos personales del alumno.
  - Insertar nota / Editar nota / Nueva nota.
- Encuestas personales sobre la opinión de los alumnos.
- Modificar datos personales del profesorado, gestionar datos sobre los cursos que se imparten, modificar datos de los alumnos matriculados, gestión de notas y faltas de asistencias.
- Base de datos con los alumnos para su identificación.
  - Insertar NRE / USUARIO.
  - Insertar contraseña.
  - Cambiar contraseña / He olvidado la contraseña.
- Seguridad a la hora del registro, comprobar contraseñas, cambio de contraseñas.
  - 3 intentos de registro en caso de equivocación, posterior bloqueo y notificación.
- Identificar cursos con su nombre, año, número de alumnos, lista de alumnos, datos de los alumnos, etc.
- Vista general de datos por cursos impartidos, porcentaje de aprobados, suspensos, media de nota, etc.
- Base de datos de información personal, agrupada por categoría:
  - PROFESORADO: DNI, nombre completo, N° de la Seguridad Social, fecha de nacimiento, teléfono, localidad, NRE y especialidad.
  - ALUMNADO: DNI, nombre completo, fecha de nacimiento, N° de la Seguridad social, teléfono, localidad y NRE.
  - CURSO: Nombre, año, fecha de inicio y de fin, profesorado, lista de alumnos, notas y valoración de cada alumno.

## • ANÁLISIS NO FUNCIONALES.

- Software libre usado para la creación de la aplicación.
- App con interfaz rápida, intuitiva y sencilla de usar.
- Protección y seguridad de la privacidad y de la información.
- Asistencia a los usuarios para solución de problemas.
- Sincronización de datos automática.

## 2. DISEÑO.

**2.1 ENTIDADES.** Una vez claras las ideas principales para la creación de la aplicación procedemos a organizar las entidades, datos y tablas para una correcta gestión de datos.

BASE DE DATOS			
PROFESORADO	ALUMNADO	CURSOS	INICIO_SESION
DNI	DNI	Nombre	usuario / nre
Nombre_completo	Nombre_completo	Año	contrasena
Fecha_nacimiento	Fecha_nacimiento	Fecha_inicio	
N_segu_social	N_segu_social	Fecha_fin	
Teléfono	Teléfono	Profesorado	
Localidad	Localidad	N_alumnos	
NRE	NRE	Datos_alumnos	
Especialidad		notas	
		alu_aprobados	
		alu_suspensos	
		valoraciones	

## 2.2 SISTEMA GESTOR DE BASES.

- Según uno de los requisitos del cliente debemos usar un lenguaje libre, así que por ello usamos MySQL. Es una de las marcas más populares de software RDBMS, que implementa un modelo cliente-servidor y su GPL determina lo que puedes hacer según las condiciones. La versión con licencia comercial está disponible si necesitas una propiedad más flexible y un soporte avanzado. Aparte, es de alta velocidad, rendimiento y tiene capacidad para manejar grandes cantidades de datos.

## 2.3 LENGUAJE DE PROGRAMACIÓN.

- Para hacer la app decidimos usar Python, que es un lenguaje de programación de alto nivel y fácil de aprender. Es ampliamente utilizado en el desarrollo de aplicaciones web, científicas y de inteligencia artificial. Python tiene una gran comunidad de desarrolladores y una amplia variedad de bibliotecas y frameworks disponibles.

## 2.4 HARDWARE.

- Para la parte del Hardware necesitaremos un servidor o un equipo designado para ello dentro de la empresa. Dispositivo que debe permanecer encendido y activo durante todo el horario laboral y estar configurado en la red interna del centro. La configuración en la red garantizará que todos los equipos del centro tengan acceso a la base de datos alojada en dicho servidor. Así tanto profesores como alumnos podrán acceder al software.
- Por otra parte se necesitarán equipos en todo el centro conectados la misma red que el servidor. También se recomienda tener un 'SAI', que es un sistema de alimentación ininterrumpida para el servidor, ya que esto evita que se apague en caso de fallo eléctrico en el centro.

## 2.5 CICLO DE VIDA.

- Para esta aplicación usaremos un modelo de cascada debido a la claridad de los requisitos desde el inicio. Y al ser una aplicación de instituto los cambios pueden ser a muy largo plazo, por lo que este modelo es perfecto.

### 3. CODIFICACIÓN.

#### 3.1 LENGUAJE DE PROGRAMACIÓN.

- Como anteriormente dijimos para esta app usaremos python, lenguaje muy utilizado por las grandes empresas. Consta de una sintaxis simple y muy legible, una amplia biblioteca de módulos y funciones, es compatible con multiplataforma y consta de una gran comunidad y seguidores, por lo hay mucha gente en constante ayuda.

#### 3.2 HERRAMIENTAS.

- Para el desarrollo del software utilizaremos el VSC, entorno de desarrollo integrado (IDE) de Microsoft. Esta facilita la identificación y corrección de errores mediante herramientas de depuración avanzadas, como puntos de interrupción, inspección de variables y seguimiento de ejecución.

#### 3.3 PASOS DE CODIFICACIÓN.

- **Código Fuente:** Son las bases del programa hechas con el lenguaje de Python. Su desarrollo lo hemos llevado a cabo con el entorno de desarrollo integrado (IDE) Visual Studio.
- **Código Objeto:** El compilador de Python traduce el código fuente directamente al Lenguaje Intermedio de Microsoft (MSIL). Este código intermedio se almacena en ensamblados, identificados por archivos con extensión .dll (bibliotecas de clases).
- **Ejecutable:** La Máquina Virtual de .NET ejecuta el MSIL, y durante la ejecución, se compila Just-In-Time.

## 4. PRUEBAS.

### 4.1 PRUEBAS UNITARIAS.

- Debido al seguimiento del proyecto se harán pruebas individuales en cada componente del software. Se usarán las herramientas integradas de J-Unit en VSC para evaluación del funcionamiento. Únicamente queremos comprobar la interfaz y los errores.

### 4.2 PRUEBAS INTEGRADAS.

- Una vez pasemos los test unitarios y verificamos que todo actúa correctamente pasaremos a las pruebas de integración con MS test de VSC.

Una vez todo esté correcto y antes de 5 días, hablaremos con el cliente para que pruebe una beta, y con eso, las opiniones y los comentarios terminariamos de solucionar aquellos pequeños fallos que pueda haber.

## 5. DOCUMENTACIÓN.

- **Guía Técnica:** Destinada al personal técnico especializado, esta documentación abarca las pruebas técnicas realizadas, así como información detallada sobre la codificación y diseño ASD subyacente de la aplicación.
- **Manual de Instalación:** Destinado al personal encargado de la instalación y a los clientes, esta guía proporciona instrucciones específicas sobre el proceso de instalación, configuración inicial, operación del sistema y consideraciones de seguridad.
- **Manual de Usuario:** Enfocado en los usuarios finales, este manual ofrece una introducción práctica, ejemplos de uso, descripción detallada de las funciones disponibles, requisitos mínimos del sistema y soluciones a problemas comunes.

## 6. EXPLOTACIÓN.

- Una vez la beta le parezca correcta al cliente haremos la fase de configuración e instalación de la aplicación. Pasaremos la app al equipo del cliente y terminaremos de unirlo al servidor, para una conexión cliente-servidor.

Una vez instalado, haremos unas pruebas más complejas para hacer una instalación más completa y así comprobar que todo va correcto en el equipo del cliente.

## 7. GESTIÓN DE MANTENIMIENTO.

- En esta fase se termina de corregir, mejorar y optimizar el software. Su duración es la mayor en todo el ciclo de vida de la aplicación, incluyendo actualizaciones y evoluciones futuras. Este mantenimiento quedará incluido en el contrato de servicio de mantenimiento de la revisable cada 3 años.

## 8.PLANNING.

	SEMANAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ANÁLISIS															
DISEÑO															
CODIFICACIÓN															
PRUEBAS															
DOCUMENTACIÓN															
EXPLOTACIÓN															
GESTIÓN															

	AVANCE DEL PROYECTO.
	DEMORA EN LA ETAPA.



## 9. BIBLIOGRAFÍA.

- App Escolar: <https://aplicacionescolar.com/>
- Aula Virtual Murcia: <https://aulavirtual.murciaeduca.es/index.php>
- Plumier:  
<https://educas.murciaeduca.es/cas/login?TARGET=https%3A%2F%2Fmirador.murciaeduca.es%2Fmirador%2Fmain.ctrl>