

# Casos de uso de la aplicación

**Identificador:** Inicio de curso

Actor(es):

- Usuario
- Sistema

Objetivo:

- Iniciar el curso (por primera vez) en el que el usuario adquirirá nuevos conocimientos sobre alguna materia.

Precondiciones:

1. El curso existe y está integrado en el sistema.

Pasos:

1. El usuario accede a la aplicación, y accede a la ventana en la que se encuentra la selección de cursos.
2. El usuario selecciona el curso que quiere realizar, y la estrategia de aprendizaje.
3. El sistema carga el archivo JSON con la información del curso, y utiliza su contenido para comenzar.

Cuestiones:

¿Qué pasa si se produce un error a la hora de leer el archivo del curso?

Alternativas:

- 3.a: se produce un error a la hora de leer el archivo JSON: se muestra un mensaje de error en la aplicación, y el usuario regresa a la ventana de selección de curso.

**Identificador:** Pausa en el curso

Actor(es):

- Usuario
- Sistema

Objetivo:

- El usuario debe de poder cambiar de curso, o cerrar la aplicación, sin que se pierda su progreso.

Pasos:

1. El usuario está en medio de la realización de un curso, y selecciona la opción de salir.
2. El sistema crea un archivo de persistencia con las estadísticas y progreso actuales, podría incluso ser una sección en el archivo JSON.
3. La próxima vez que se seleccione el curso, el sistema revisará este archivo, y leerá su contenido para dejar el estado como antes.

Cuestiones:

¿Qué pasa si se produce un error a la hora de leer el archivo de persistencia?

¿Qué pasa si la aplicación se cierra de forma imprevista?

Alternativas:

- 3.a: se produce un error a la hora de leer el archivo de persistencia: se muestra un mensaje de error en la aplicación; el usuario tendrá que empezar desde el principio.
- 1.a: la aplicación se cierra sin que el sistema haya guardado: para minimizar pérdidas, cada cierto tiempo se realiza un guardado automático del progreso.

**Identificador:** Guardado de estadísticas

Actor(es):

- Usuario
- Sistema

Objetivo:

- El usuario debe de poder conservar sus estadísticas de uso de la aplicación, como tiempo de uso o mejores rachas.

Pasos:

1. El usuario usa la aplicación.
2. El sistema actualiza constantemente un archivo de persistencia con las estadísticas de la aplicación.
3. La próxima vez que se abra la aplicación, el sistema revisará este archivo, y leerá su contenido para dejar el estado como antes.

Cuestiones:

¿Qué pasa si se produce un error a la hora de leer el archivo de persistencia?

Alternativas:

- 3.a: se produce un error a la hora de leer el archivo de persistencia: se muestra un mensaje de error en la aplicación; el usuario tendrá que empezar desde el principio.

**Identificador:** Sistema de logros

Actor(es):

- Usuario
- Sistema

Objetivo:

- El usuario obtendrá logros (o medallas) cuando logre realizar ciertas hazañas. (Ej: racha de más de 20 días)

Pasos:

1. El usuario realiza una hazaña.
2. El sistema lo reconoce y muestra un mensaje de celebración.
3. El sistema añade el logro a la colección del usuario.
4. El sistema actualiza el archivo de persistencia.

**Identificador:** Compartir cursos

Actor(es):

- Usuario

Objetivo:

- El usuario podrá compartir un curso creado con otros usuarios.

Precondiciones:

- El curso ha sido creado previamente.

Pasos:

1. El usuario accede a la carpeta en la que tiene almacenados los cursos.
2. Envía a otro usuario el archivo JSON correspondiente al curso que desea compartir.
3. El otro usuario lo recibe y lo coloca en su carpeta.

## MODELADO DOMINIO

@startuml

```
class Usuario {  
  
    String nombre  
  
    String email  
}
```

```
        String contraseña

        void registrarse()

        void iniciarSesion()

        void iniciarCurso(Curso curso)

        void guardarProgreso(Curso curso)

        void guardarEstadisticas(Estadisticas estadisticas)
    }

    abstract EstrategiaAprendizaje {

    }

    class Secuencial extends EstrategiaAprendizaje {}

    class RepeticionEspaciada extends EstrategiaAprendizaje {}

    class Aleatoria extends EstrategiaAprendizaje {}


    class Curso {

        String nombre

        String dominio

        void responderPreguntas(BloqueContenido bloque, EstrategiaAprendizaje
estrategia)
    }

    class BloqueContenido {

        String titulo
    }

    abstract Pregunta {

        String enunciado
    }

    class PreguntaTest implements Pregunta {
```

```

        List<String> opciones

        void validarRespuesta(String respuesta)
    }

class Completar implements Pregunta {

    String respuestaCorrecta

    void validarRespuesta(String respuesta)
}

class Flashcard implements Pregunta {

    String informacion

    void mostrar()
}

class Estadisticas {

    int tiempoUso

    int mejorRacha

    void actualizarEstadisticas()

    void conseguirLogro()

    array logrosConseguidos
}

Usuario "*" -- "*" Curso : realiza

Curso "1" -- "*" BloqueContenido : contiene

BloqueContenido "1" -- "*" Pregunta : tiene

Pregunta <|-- PreguntaTest

Pregunta <|-- Completar

Pregunta <|-- Flashcard

EstrategiaAprendizaje <|-- Secuencial

EstrategiaAprendizaje <|-- RepeticionEspaciada

```

```

EstrategiaAprendizaje <|-- Aleatoria

Usuario "1" -- "1" Estadisticas : tiene

Curso "*" -- "1" EstrategiaAprendizaje : aplica

```

@endum1

