

Apellidos y nombre: _____

Se desea desarrollar en **Yii 2 framework** una aplicación web de catalogación musical, estilo Spotify.

En Spotify tenemos usuarios y también tenemos álbumes, canciones y artistas. Entre *álbumes-canciones* y entre *artistas-canciones* se establecen relaciones de muchos a muchos.

1. (1,5 puntos) Generar el CRUD básico de álbumes, canciones y artistas, teniendo en cuenta que se debe impedir borrar un álbum, una canción o un artista si ello provocara una violación de la integridad referencial.
2. (2 puntos) Implementar en `site/index` un buscador integrado que, usando **una única barra de búsqueda**, localice aquellos álbumes, canciones y artistas (las tres cosas a la vez) que contengan la cadena a buscar. Los resultados incluirán enlaces a las correspondientes acciones `view` de cada elemento.
3. (2 puntos) En `albumes/index`, mostrar también la duración total del álbum como una columna más que admita filtrado y ordenación y que esté formateada como *minutos:segundos* (por ejemplo, 135:27).
4. (1,5 puntos) En `albumes/view`, mostrar todos los datos del álbum, incluyendo su duración total, los datos de las canciones que lo forman (con paginación y ordenación) y los artistas que intervienen en él (sin paginación ni ordenación).
5. (3 puntos) En `canciones/create` y `canciones/update`, diseñar una UI (**preferentemente mediante AJAX**) que permita incorporar con facilidad la canción a uno o varios álbumes sobre la marcha.

Aunque no haría falta decirlo, la aplicación no debe mostrar al usuario final ningún dato interno a la misma, incluyendo principalmente los identificadores o *id*.

La evaluación de cada ejercicio se llevará a cabo siguiendo la siguiente rúbrica:

% del total	Evaluación
0 %	No entregado, apenas empezado o manifiestamente incorrecto.
25 %	Se esboza un mínimo planteamiento pero no se acerca a una solución funcional.
50 %	Hay una solución incompleta que no cumple todos los requisitos.
75 %	La solución es completa pero ineficiente, poco elegante o mal documentada.
100 %	Solución completa, eficiente, elegante y bien documentada.

Se tendrá en cuenta, especialmente:

- Aplicación coherente del patrón MVC.
- Uso de comentarios adecuados en el código.
- Documentación del mismo en formato PHPDoc.
- Aplicación de los estándares de codificación y reglas de estilo usados por el framework.
- Elección adecuada de las rutas de acceso a la funcionalidad de la aplicación.
- Formato apropiado de los datos.
- Uso de mecanismos de comunicación asíncrona.
- Escalabilidad.