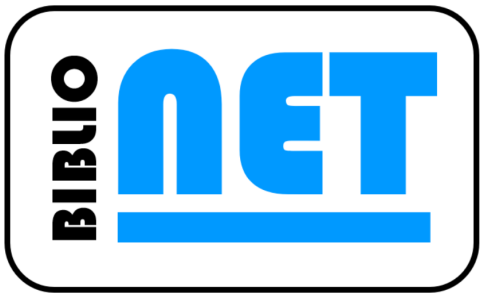
DP1 2020-2021
Documento de Requisitos y Análisis del Sistema

Proyecto BiblioNET



github.com/gii-is-DP1/dp1-2020-g2-02.git

Miembros:

- Fernando José González García
- Isabel de Vergara de Orellana
- Beatriz María Llamas Sainz-Pardo
- Iván Romero Costa
- Jorge Luis Ruiz-Henestrosa Ruiz
- Alejandro Sanabria Cabeza

Tutor: José Antonio Parejo Maestre

GRUPO G2-02

Versión 3
20 de enero de 2021

Historial de versiones

Fecha	Versión	Descripción de los cambios	Sprint
21/12/2020	V1	● Creación del documento	3
23/12/2020	V1	● Creación H16 ● Añadido campo “Descatalogado” a libro ● Pequeñas mejoras generales de la página	3
26/12/2020	V1	● Cambios y mejoras en la interfaz de la página ● Creación H21,H22	3
27/12/2020		● Avances y mejoras relacionados con las entidades Libro y Prestamo ● Cambios en las entidades Ejemplar y Miembro	3
31/12/2020	V1	● Cambios en listLibro y editEjemplar ● Introducción del documento	3
3/01/2021		● Creación H5 ● Modificación H21 ● Modificación restricciones simples (Entregable1) ● Creación algunas restricciones simples	3
5/01/2021	V1	● Modificadas las historias de usuario H1,H7 ● Creación H9,H20 ● Creación diagramas UML ● Creación RN3	3
7/01/2021	V1	● Modificación H21,H20 ● Mejoras de la vista relacionada con Libro ● Revisión general ● Patrones de diseño y arquitectónicos ● Creación de decisiones de diseño	3
8/01/2021	V1	● Modificación y mejora H9 ● Modificación diagramas UML ● Modificación patrones de diseño y arquitectónicos ● Avances en decisiones de diseño	3
9/01/2021	V1	● Pequeñas mejoras y algunos cambios ● Revisión general proyecto	3

Contents

Introducción

La página web que vamos a crear se va a encargar de facilitar la gestión de una biblioteca.
A través de esta web podremos registrarnos como miembros de la biblioteca para poder pedir y devolver préstamos de libros.
Además el bibliotecario tendrá fácil acceso a todos los datos relacionados con los miembros, al estado de todos los libros y podrá comunicarse con un proveedor encargado de cubrir todas las necesidades de la biblioteca.

Diagrama(s) UML:

Diagrama de Dominio/Diseño

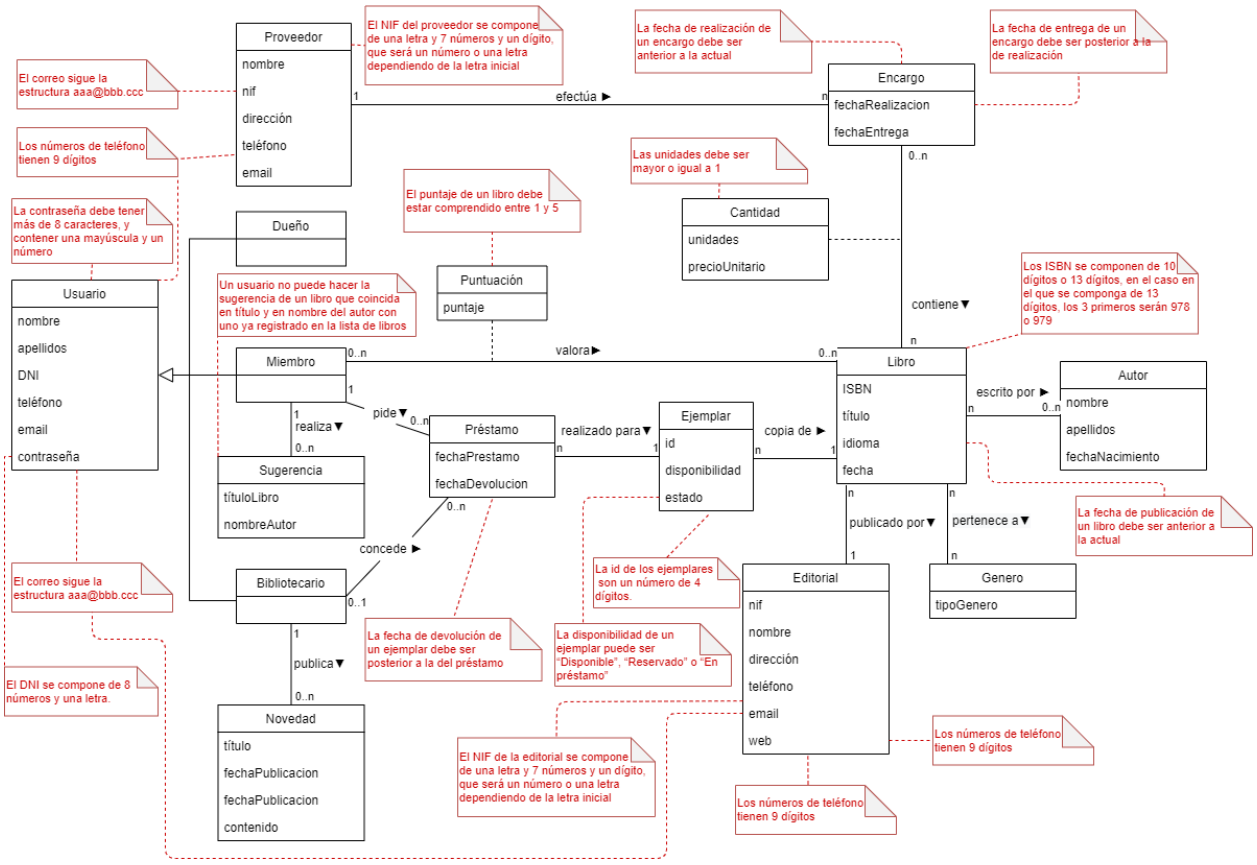
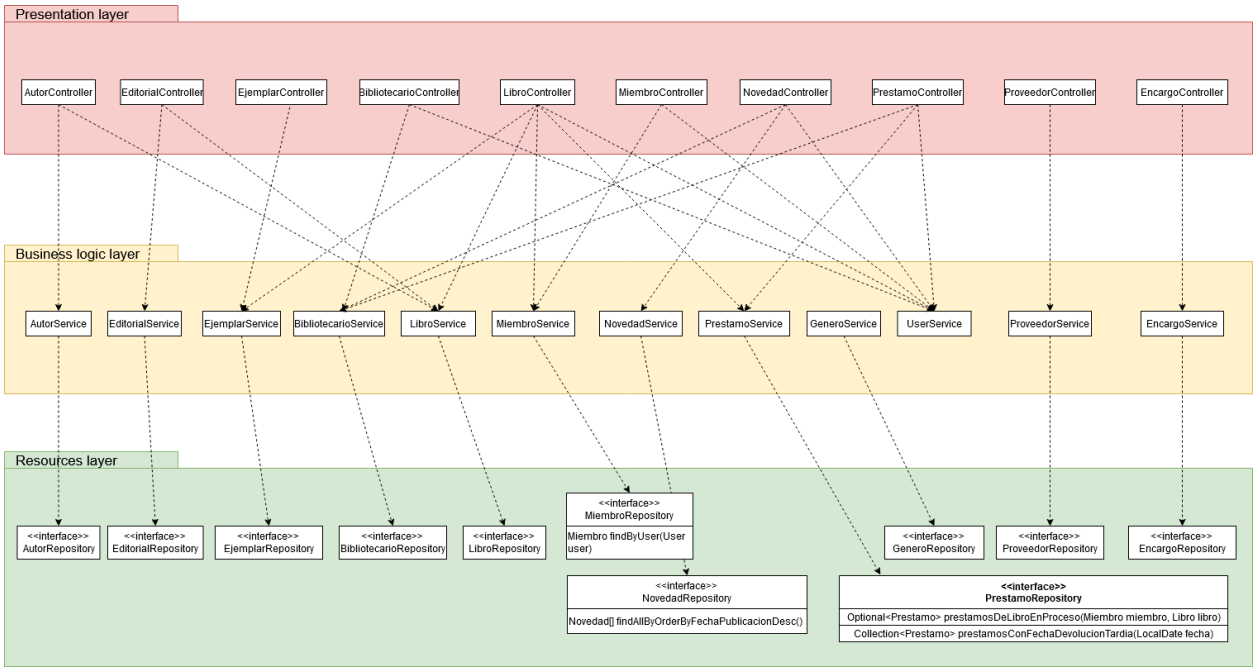


Diagrama de Capas (incluyendo Controladores, Servicios y Repositorios)



Patrones de diseño y arquitectónicos aplicados

Patrón: Modelo Vista Controlador

Tipo: Arquitectónico

Contexto de Aplicación

Al ser un patrón arquitectónico se usa en toda la aplicación.

Clases o paquetes creados

Mantenemos los paquetes de PetClinic creando en ellos las clases necesarias para el proyecto.

Ventajas alcanzadas al aplicar el patrón

Hemos separado claramente entre modelos, vistas y controladores lo que nos ha facilitado la realización de pruebas unitarias además de aportar cohesión y disminuir el acoplamiento.

Decisiones de diseño

En esta sección describiremos las decisiones de diseño que se han tomado a lo largo del desarrollo de la aplicación que vayan más allá de la mera aplicación de patrones de diseño o arquitectónicos.

Decisión 1: Descatalogación de libros

Descripción del problema:

Planteamos funciones para eliminar libros y miembros de la base de datos, sin embargo descartamos estas opciones debido a que no queríamos perder posibles datos importantes al borrar en cascada.

Alternativas de solución evaluadas:

Alternativa 1.a: Creamos las alternativas de descatalogar libros y desactivar ejemplares en lugar de eliminarlos

Ventajas:

- Es cómodo de usar para el bibliotecario.
- No se borran posibles elementos importantes de la base de datos.

Inconvenientes:

- Al no eliminar los elementos de la base de datos el tamaño de esta aumentará considerablemente.

Justificación de la solución adoptada

Nos decidimos por la alternativa 1.a al ser la más práctica para nuestro sistema.

Decisión 2: Préstamos manuales

Descripción del problema:

Nos gustaría que el usuario pudiese tomar en préstamo ejemplares automáticamente(a no ser que estos ejemplares ya estuviesen reservados) sin pasar por manos del bibliotecario.El problema es que la realización de este sistema automatizado aumentaba considerablemente la complejidad de nuestro código.

Alternativas de solución evaluadas:

Alternativa 1.a: Hemos añadido una vista donde el bibliotecario tiene un listado de todos los préstamos, en esta vista puede concederlos, rechazarlos y finalizar los devueltos, esto le da control total de los ejemplares y facilita las funciones del bibliotecario al reunirse en una sola vista.

Ventajas:

- Aumenta la eficacia del código
- El bibliotecario tiene control total de los préstamos
- Reúne en una sola vista todas las responsabilidades con los préstamos del bibliotecario.

Inconvenientes:

- Es el bibliotecario el que tiene que lidiar con las peticiones de préstamos manualmente

Justificación de la solución adoptada

Nos decidimos por la alternativa 1.a al ser la más práctica para nuestro sistema.

Decisión 3: Rol de los proveedores en la aplicación

Descripción del problema:

Planteamos considerar a los proveedores como usuarios en la aplicación, de forma que estos pudiesen insertar encargos e informar de cuando estaban siendo entregados. Sin embargo, desechamos la idea ya que probablemente estos dispongan de su propia aplicación para la gestión de sus encargos.

Alternativas de solución evaluadas:

Alternativa 1.a: Proveedor como usuario de la aplicación

Ventajas:

- Ahorra trabajo al personal de la biblioteca.

Inconvenientes:

- Puede resultar demasiado confuso para el proveedor, y que este acabe por no usar la aplicación.

Alternativa 1.b: El proveedor es ajeno a la aplicación, el bibliotecario se encarga de almacenar los datos sobre sus encargos.

Ventajas:

- Ahorra trabajo al proveedor.

Inconvenientes:

- El bibliotecario deberá insertar manualmente los datos de los encargos.

Justificación de la solución adoptada

Nos decidimos por la alternativa 1.b al considerarla la más adaptada a la realidad.

Decisión 4: Implementación del estado de los préstamos

Descripción del problema:

Es necesario registrar de alguna manera el estado de los préstamos y los ejemplares objetos de los préstamos. Los préstamos pasan por distintas fases, primero, el miembro indica por la aplicación que está interesado en un libro, de forma que se le reserva un ejemplar, luego, el miembro pasa por la biblioteca donde un bibliotecario le entregará dicho ejemplar, y finalmente, el miembro devolverá el ejemplar dando el préstamo por finalizado.

Alternativas de solución evaluadas:

Alternativa 1.a: Añadir un campo de disponibilidad para la entidad Ejemplar, incluyendo como mínimo, los estados de disponible, reservado y en préstamo. Además añadir un campo booleano en la entidad Préstamo que indique si este ha finalizado.

Ventajas:

- Mayor eficacia en el código a la hora de buscar ejemplares disponibles, al poder comprobar directamente su disponibilidad.

Inconvenientes:

- Se necesita un campo en cada entidad, aumentando ligeramente el tamaño de los datos.

Alternativa 1.b: Añadir un campo de estado en la entidad Préstamo, incluyendo como mínimo los estados reservado, en préstamo, y finalizado.

Ventajas:

- Menos datos al tener solo un campo.
- Implementación más sencilla.

Inconvenientes:

- Menor eficacia en el código a la hora de buscar ejemplares disponibles, ya que para ello habría que recorrer primero todas las entradas de préstamos, para saber qué ejemplares están ocupados por estos.

Justificación de la solución adoptada

Nos decidimos por la alternativa 1.b, ya que además de ser más eficaz, se necesitaba igualmente un campo extra en la entidad Ejemplar para indicar los ejemplares descatalogados, por lo que no hay impacto en el tamaño de la base de datos.