Obligatorio I de Diseño de Aplicaciones I

La empresa Threat Level Midnight Entertainment desea lanzarse como competidora de los grandes proveedores de streaming de series y películas, para lo cual lo ha contratado para que desarrolle un *POC* (*Proof of Concept*) de su nueva plataforma.

Luego de varias rondas de discovery con diversos expertos en producto, su CEO ha decidido focalizar la plataforma exclusivamente en películas y se han relevado los requerimientos detallados a continuación.

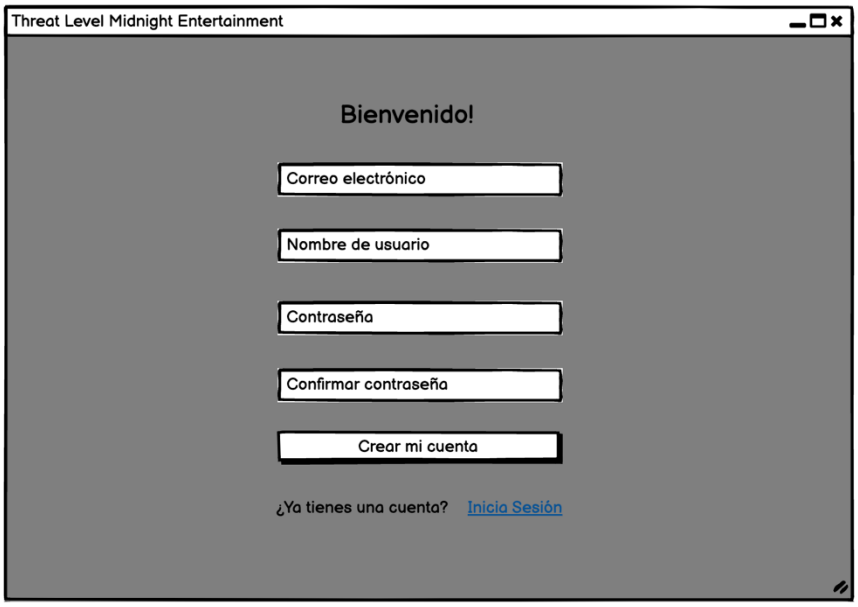
Requerimientos:

Requerimientos para todos los equipos:

1. **Registro e inicio de sesión de usuarios**

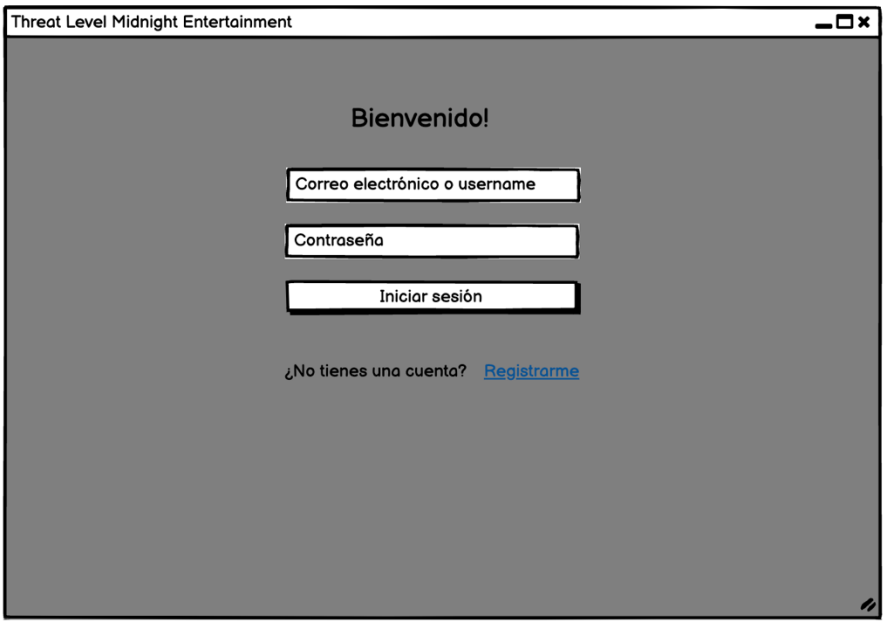
Un usuario se registra en la plataforma indicando

- Su correo electrónico, que debe validarse que sea único en el sistema.
- Un nombre de usuario, alfanumérico, con mínimo de 10 caracteres y máximo de 20, que también debe validarse como único.
- Una contraseña, de mínimo 10 caracteres y máximo 30, la cual debe verificarse con un campo “Confirmar contraseña”.



Un usuario inicia sesión en la plataforma indicando

- Nombre de usuario o correo electrónico
- Contraseña



2. Creación de usuario administrador

Por defecto, al iniciar el sistema, debe estar creado un usuario de tipo administrador que estará habilitado para dar de alta y baja los contenidos dentro de la plataforma.

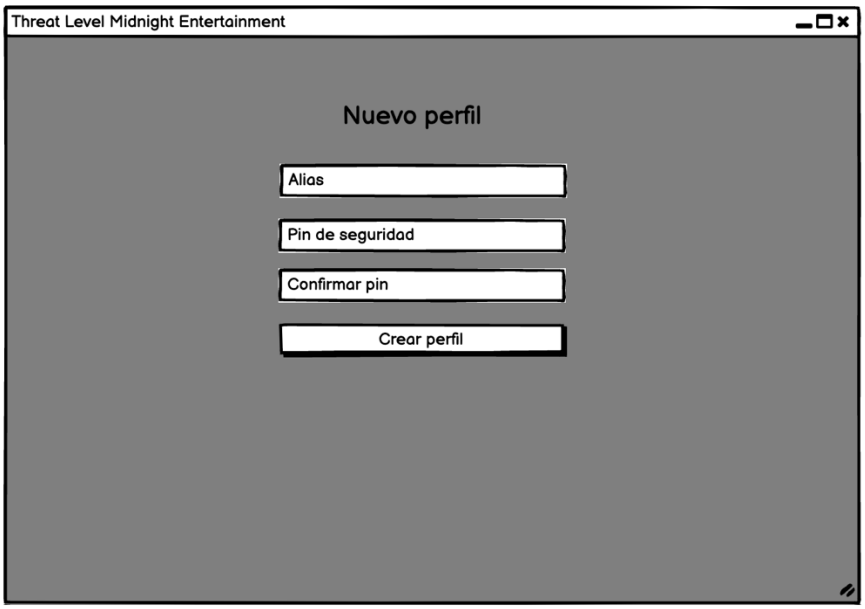
Vale aclarar que el inicio de sesión para el administrador es el mismo que el de un usuario normal, se ocultarán/mostrarán controles en la interfaz según el tipo de usuario detectado.
El correo electrónico y nombre de usuario del administrador queda a elección de los desarrolladores, pero deben estar **documentadas** sus credenciales.

3. Administración de perfiles de usuario

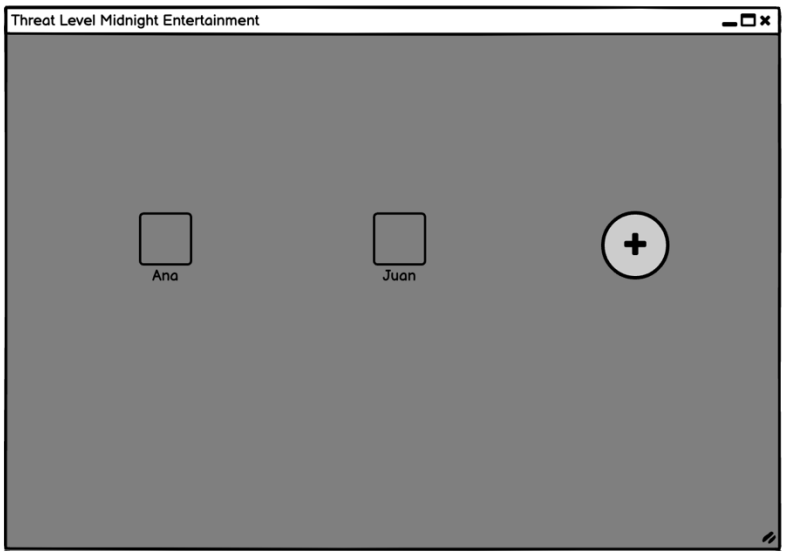
La primera vez que un usuario se registra debe crear su perfil en la plataforma, quedando este perfil como **owner** de la cuenta.

La creación del perfil implica

- Elegir un alias, texto, con un mínimo de 1 carácter y un máximo de 15.
- Un pin, código numérico de 5 dígitos, con el objetivo de proteger a los niños de que accedan a un perfil con películas que no son para su rango etario.

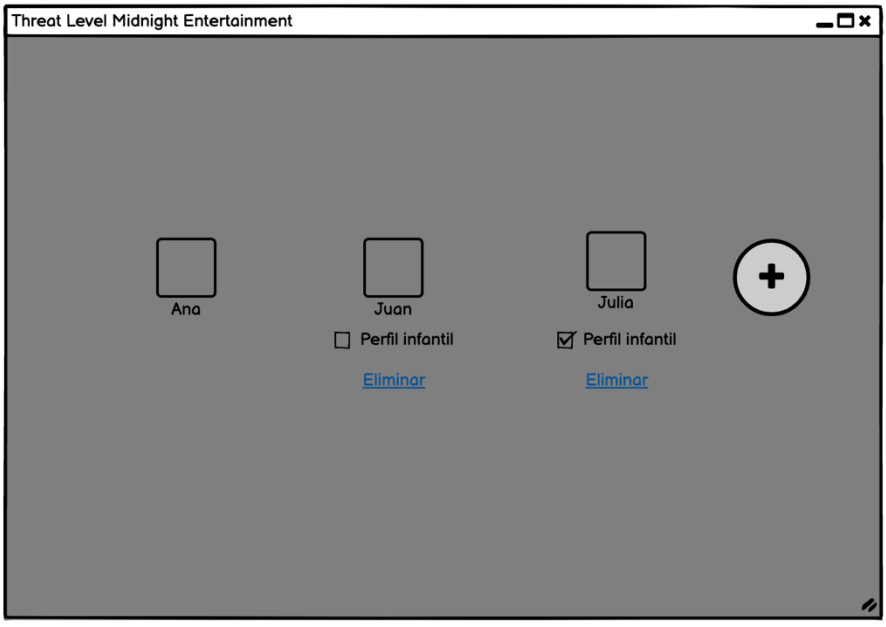


Luego, en el inicio, se visualizan los perfiles de la cuenta. Al elegir uno, debe ingresarse el pin para acceder.



El perfil que es **owner** de la cuenta es el único que puede eliminar o marcar que un perfil corresponde al de un niño (al marcarlo como que es de un niño no se le solicita más el pin de acceso).

Nota: El perfil de **owner** no puede ser marcado como un perfil de niño.



Cada cuenta puede tener como máximo 4 perfiles diferentes, incluyendo al **owner**.

4. Administración de películas

El usuario administrador es el encargado de la gestión del contenido de la plataforma. Como tal, puede dar de alta y eliminar películas. De las mismas se conoce

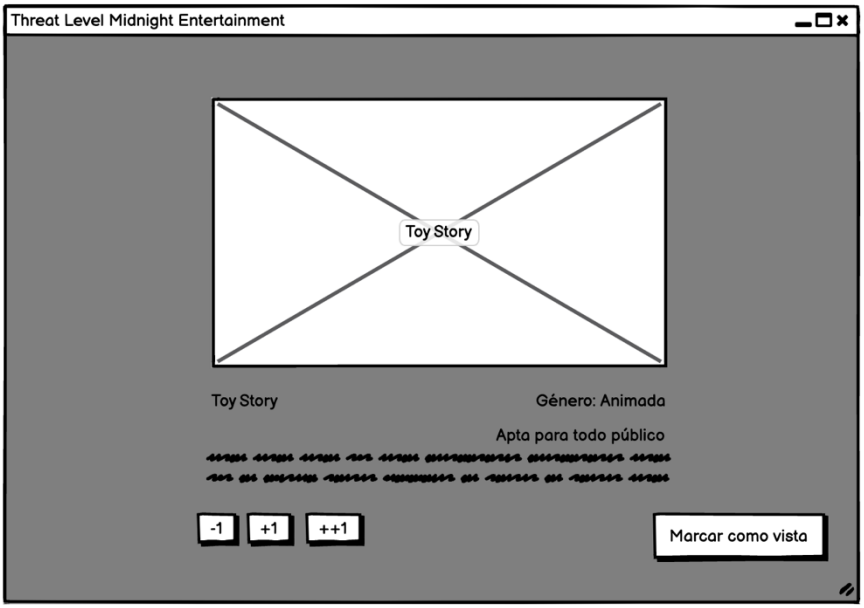
- Su nombre (no vacío)
- Géneros asociados (al menos uno que se denominará como **género principal**, pudiendo la película tener más géneros que serán los **géneros secundarios**)
- Poster de la película (imagen jpg o png)
- Descripción
- Si es una película apta para todo público
- Si la misma es patrocinada o no (por ejemplo, para el caso de las películas de contenido original de Threat Level Midnight Entertainment puede ser conveniente marcarlas como patrocinadas para que aparezcan destacadas en el ranking del usuario)

5. Administración de géneros

Los géneros se identifican por un nombre (no vacío) y tienen una descripción. Los mismos pueden ser eliminados exclusivamente si no tienen películas asociadas. **NO** es necesario implementar la modificación de géneros.

6. Calificación de películas

En la pantalla principal debe verse un listado de películas. Al clickear en el poster de una de ellas, el usuario podrá acceder a su vista de detalle para calificarla.



Una película puede ser calificada con **-1** como voto negativo, **+1** como voto positivo y **++1** como voto muy positivo.

7. Marcar película como vista

La reproducción de películas se simulará mediante un botón de **"Marcar película como vista"** como aparece en el boceto anterior. No será necesario el manejo de ningún tipo de video.

8. Ranking y personalización

Como fuese mencionado en el punto **6.** un usuario (aplica para cualquier usuario) debe ver un listado de películas al momento de ingresar en la pantalla principal. El orden de las mismas se determina según distintos criterios de ordenamiento, siendo los mismos

- El primer criterio, por defecto, ordena las películas por género (nombre) alfabéticamente y en caso de empate, es decir existen varias películas dentro de un mismo género, las mismas se ordenan también por nombre en orden alfabético.
- El segundo criterio sigue la base del primero, criterio alfabético, con el agregado de que las películas que fueron marcadas como patrocinadas aparecen al principio del listado, sin importar el género que tengan. Es decir, el patrocinio adquiere mayor prioridad frente al orden por defecto.

- El tercer y último criterio ordena las películas por género, pero el orden de los géneros es por puntaje descendente, y dentro de cada género las películas se ordenan por nombre. No se tiene en cuenta el patrocinio.

El cálculo del puntaje de un género sigue las siguientes reglas

- **Reproducir** una película incrementa el puntaje del **género principal** de la misma en 1
- **Votar** una película con **+1** se comporta igual que **Reproducir** (si ambas cosas suceden a la vez el puntaje se acumula, subiendo el puntaje del género en 2 puntos).
- **Votar** una película con **-1** disminuye el puntaje del **género principal** en 1
- Y finalmente, el voto **++1** incrementa el puntaje del **principal** en 2 y el de los **secundarios** en 1.

Dentro de un mismo género, de existir "empate" ordenando por nombre de la película el atributo final para la ordenación será el de la fecha de creación de la película dentro del sistema. Las creadas más recientemente aparecerán primero, esta regla aplica para los 3 criterios de ordenación.

El algoritmo de ordenamiento es configurado por el usuario administrador, pudiendo cambiar el mismo en tiempo de ejecución.

9. Control de contenido apto para todo público

Independientemente del orden establecido en el punto 8. deben filtrarse las películas que no sean aptas para todo público si el perfil actual es un perfil infantil.

Requerimientos adicionales para equipos de tres estudiantes:

10. Películas relacionadas

Una película puede estar relacionada con una o más películas, sin importar si son del mismo género o demás, esta es una relación que se establece manualmente por el administrador (queda a criterio del equipo hacerlo en la pantalla de administración del equipo o en otro lugar).

Al ingresar a la vista de detalle de una película, debajo de la misma deben verse un mini póster de las películas relacionadas, y debe ser posible ir al detalle de una película seleccionada.

11. Exportación de listado de películas

Debe ser posible exportar el listado total de las películas del sistema a un archivo .txt, teniendo en cuenta que a futuro puede solicitarse que la exportación sea a nuevos formatos como Excel.

Los mockups incluídos en la letra son meramente ilustrativos y puede diseñarse una interfaz de usuario diferente siempre que cumpla con la totalidad de los requerimientos funcionales.

Requerimientos no funcionales para todos los equipos:

1) Debe estructurar los componentes de la solución de forma que la lógica de negocio y la interfaz de usuario **estén separadas**.

2) Ambos componentes deben ser desarrollados usando C# en Visual Studio 2019, y utilizando GitHub como repositorio de código, **alojando el repositorio dentro de la organización de ORT-DA1 creada con dichos fines**.

3) Todos los componentes que contengan lógica de negocio deberán ser diseñados y construidos 100% **utilizando TDD**; y contener la implementación de todas las reglas de negocio que se especifican en la letra. **No se debe construir la interfaz de usuario siguiendo dicha metodología.**

Nota sobre requerimientos (todos los equipos):

Nota: **La totalidad y detalle de los requisitos serán relevados a partir de consultas en el foro correspondiente en aulas**, las que deben tener un título descriptivo de su contenido. Tener en cuenta al momento de planificar el trabajo que las respuestas en Aulas **pueden demorar hasta 48 horas**. Para evitar complejidades innecesarias se realizaron simplificaciones al dominio del problema real.

Entrega del obligatorio:

A continuación, se describen los requisitos esperados para la entrega del obligatorio:

Diseño e Implementación:

Se debe entregar una aplicación que contemple toda la funcionalidad descrita en este documento. **Para esta primera instancia la solución guardará toda la información en memoria**, es decir, no es necesario el uso de base de datos (esto se implementará en la segunda entrega).

El desarrollo de todo el obligatorio debe cumplir:

- Estar en un repositorio **Git**, siguiendo **Gitflow**.
- **El nombre del repositorio debe estar conformado de la siguiente forma**:
  - NumeroEstudiante1_NumeroEstudiante2
- Haber sido desarrollado utilizando **TDD** (desarrollo guiado por pruebas) lo que involucra otras dos prácticas: escribir las pruebas primero (Test First Development) y refactoring. De esta forma se utilizan las pruebas unitarias para dirigir el diseño.
- Cumplir los lineamientos de **Clean Code** (capítulos 1 al 10, y el 12), utilizando las técnicas y metodologías ágiles presentadas para crear código limpio.
- Utilizar las convenciones de codificación ("idioms") de C#: https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions
- Se requiere escribir los casos de prueba automatizados con el framework de unit tests visto en el curso, **documentando** y **justificando** las pruebas realizadas (ver documentación).
- Como parte de la evaluación se revisará el **nivel de cobertura** de los test sobre el código entregado. Por lo que se debe entregar un reporte y un **análisis de la cobertura** de las pruebas justificando cada uno de los valores obtenidos de acuerdo al diseño elaborado para la solución.

Documentación:

La documentación entregada debe ser en **un solo** documento digital (**límite 15 páginas, sin contar anexos)**, que contenga la siguiente información ordenada e indexada:

1. Carátula de documento como indica el documento 302. **Debe incluirse un link al repositorio de la entrega.**
2. Descripción general del trabajo y del sistema: si alguna funcionalidad no fue implementada o si hay algún error conocido (bug) deben ser descritos aquí.
3. Descripción y justificación de diseño. Para ello se debe incluir:
    a. Diagrama(s) de paquetes mostrando la organización general de la aplicación (**no es necesario incluir el paquete de pruebas**).
    b. Diagramas de clases: al menos uno por paquete. Los diagramas deberán ser lo más completos y formales posible **respecto a lo que se desea comunicar**.
    c. **Explicación de los mecanismos generales y descripción de las principales decisiones de diseño tomadas**. A modo de ejemplo, se debería describir la forma en que la interfaz de usuario interactúa con el dominio, cómo se almacenan los datos, el manejo de errores y excepciones, o la utilización de polimorfismo.
    d. **Breve análisis de los criterios seguidos para asignar las responsabilidades**.
4. Cobertura de pruebas unitarias con su debido análisis y justificaciones. En caso que la cobertura de líneas de código no supere el 90% para alguna funcionalidad, se deberá incluir un análisis sobre porque, explicando que faltó en los test para que no se ejercitara la parte de código no cubierto
5. **Para las funcionalidades de <u>Control de contenido apto para todo público</u> y <u>Administración de películas</u>** se debe documentar los casos de prueba elaborados, incluyendo: valores inválidos, valores límites, ingreso de tipos de datos erróneos, datos vacíos, datos nulos, omisión de campos obligatorios, formato inválidos, pruebas de las reglas del negocio, etc. **Se debe entregar un reporte que muestre evidencia del resultado de ejecutar estos casos de prueba, que puede ser registrada mediante capturas de pantalla (como anexo en el documento) o realizando uno o más videos cortos publicados en Youtube, en los cuales se pueda apreciar claramente la ejecución de las pruebas**. **Los links a estos videos deben incluirse en este documento y se debe verificar que se hayan compartido correctamente y que un tercero pueda verlos.**

Importante: se espera que la descripción y justificaciones sigan un orden lógico, **intercalando diagramas y explicaciones según sea necesario**. Las condiciones de entrega serán evaluadas como si se le estuviese entregando a un cliente real: **prolijidad, claridad, profesionalismo**, etc.

Entrega:

La entrega debe estar compuesta por los siguientes elementos contenidos en el repositorio de GitHub:

- Una carpeta con la aplicación compilada.
- Código fuente de la aplicación (merge a main), incluyendo el proyecto que permita probar y ejecutar.
- Documentación digital (incluyendo modelado UML).

La entrega de los obligatorios será en formato digital. **Los principales aspectos a destacar sobre la entrega online de obligatorios son:**

1. **La entrega de la documentación** se realizará desde gestion.ort.edu.uy (además de estar en el repositorio), el código fuente **NO** se sube.
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. Sugerimos realizarlo con anticipación.
3. Uno de los integrantes del grupo de obligatorio será el administrador del mismo y es quien formará el equipo y subirá la entrega
4. Cada equipo debe entregar un único archivo en formato zip o rar (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
5. El archivo a subir debe tener un tamaño máximo de 40mb.
6. Les sugerimos realicen una 'prueba de subida' al menos un día antes, donde conformarán el 'grupo de obligatorio'.
7. La hora tope para subir el archivo será las 21:00 del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc)
9. Aquellos que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta antes de las 20:00hs. del día de la entrega

En caso de tener una situación particular de fuerza mayor, es necesario dirigirse con suficiente antelación al plazo de entrega, a la Coordinadora de Cursos o el Secretario Docente.

La entrega final del obligatorio debe además estar claramente identificada (mediante un tag con el texto "**entrega_1**") en la rama **main** en el repositorio del grupo de **GitHub** dentro de la organización **ORT-DA1**. Debe incluir:

- **Código fuente** de la aplicación, incluyendo el proyecto que permita compilar, probar, ejecutar y ejecutar **pruebas unitarias**.
- Carpeta con la aplicación compilada en **release**.
- Documentación digital en formato **PDF** (incluyendo modelado **UML**).
- **Importante: La documentación debe incluir la url del repositorio del grupo, de forma que los docentes puedan asociar al grupo con su trabajo en github. Este es un requerimiento mandatorio, y no se revisará el código si no se cumple.**
- No se aceptará que a main se suba un archivo comprimido con la entrega final (ej: Zip, Rar, otros), el contenido de main deberá ser la **integración** de la rama **develop** a la **main**.

Rúbrica de evaluación:

Puntaje total: 20 puntos.

	Evaluación de	Pts.	Criterios de corrección
Funcionalidad	Implementación de la funcionalidad pedida y calidad de la interfaz de usuario.	6	**Excelente**: Se implementa toda la funcionalidad, tiene buena usabilidad y no hay errores importantes de funcionamiento. **Correcto**: Falta algún punto no central de la funcionalidad, existen detalles no bloqueantes de funcionamiento o la aplicación no es fácil de usar. **No suficiente**: Faltan partes importantes (por alcance o dificultad) de la funcionalidad. Existe gran cantidad de errores o funcionalidades no usables.
Diseño y documentación	Diagramas de paquetes Diagramas de clases Justificación del diseño Calidad del diseño Informe de cobertura de los casos de prueba Evidencia de pruebas Claridad de la documentación Organización de la documentación (debe tener un orden lógico y un índice) Completitud de la documentación	6	**Excelente**: Se presenta un documento completo en función de lo que se pide, ordenado y prolijo, que explica la solución entregada en todos sus aspectos. Se utiliza la notación vista en clase, de manera formal y correcta, para presentar los aspectos importantes del diseño. Se intercalan diagramas y explicaciones, que permiten comprender el diseño de la solución, las decisiones tomadas y la motivación detrás de las mismas. **Correcto:** Se presenta un documento prolijo que describe el diseño de la solución. Se describen los aspectos más importantes, pero hay errores en la

	Prolijidad de la documentación		nomenclatura. Falta describir aspectos importantes. El orden de la documentación no permite seguir un hilo descriptivo. **No suficiente**: Falta detallar parte importante de la solución. No se usa la notación vista en clase, o se usa en forma equivocada o incompleta en repetidas ocasiones. Se presenta la documentación como una sucesión de diagramas sin explicación.
Calidad del código y metodología de desarrollo	Desarrollo guiado por las pruebas (TDD) y técnicas de refactoreo de código Buenas prácticas de estilo y codificación y su impacto en la mantenibilidad (Clean Code) Correcto uso de las tecnologías Claridad del código Concordancia con el diseño	8	**Excelente**: El código es prolijo, fácil de entender y cumple con los puntos cubiertos en el curso sobre Clean Code. Lo construido coincide con lo detallado en la documentación de diseño. Se evidencia el uso de TDD mediante los commits en el repositorio. Hay buena cobertura de casos de prueba. **Correcto**: El código es prolijo en general, aunque presenta detalles a mejorar. Lo construido se corresponde en términos generales con lo presentado en el documento. Hay pruebas unitarias, aunque no se evidencie el uso de TDD. **No suficiente**: El código es desprolijo o difícil de entender. No cumple en repetidos casos lo propuesto por Clean Code. No hay pruebas unitarias.

Información importante

Lectura de obligatorio: 05-09-2022
Plazo máximo de entrega: 13-10-2022
Puntaje mínimo / máximo: 0 / 20 puntos