**Consignas para la resolución del examen**

1. **Conectarse a Miel.**
2. **Descargar el presente documento.**
3. **Crear un nuevo Proyecto en Eclipse.**
4. **Compartir el nuevo proyecto en un repositorio público de GitHub**
5. **Actualizar el proyecto siguiendo lo descripto en las siguientes páginas, usando la metodología TDD.**
6. **Entregar por prácticas de Miel la URL del repositorio público. Sólo se contemplarán los cambios subidos dentro del horario del examen.**

**Panini**

**Nos contrata Panini para desarrollar la versión digital del álbum de figuritas del mundial.**

**Cada figurita está compuesta por su código identificatorio, la letra del grupo al que pertenece, la selección, el nombre del jugador y el valor del jugador en el mercado.**

**El sistema puede tener dos tipos de usuarios registrados. Los administradores, quienes son los encargados de administrar toda la información de las figuritas que se van a comercializar, y los usuarios finales que son los que en definitiva jugarán.**

**Según el tipo de usuario es la acción por realizar, por ejemplo, el método agregarFigurita de un usuario administrador se encarga de dar de alta una nueva figurita con sus características. Ahora bien, el método agregarFigurita de un usuario final en realidad lo que hace es agregar esa figurita en el stock disponible para ese usuario.**

**Tanto la base de datos de figurita generada por los administradores como el stock de figuritas de cada usuario deben estar ordenados por grupo, selección y número de figurita en la selección.**

**Las figuritas tienen un identificador único que está compuesto por la selección y el número de figurita (cada selección tiene alrededor de 22 jugadores). Si un administrado intenta dar de alta un código ya existente, el sistema debe arrojar la excepción “CodigoExistente”.**

**A diferencia de lo que ocurre a nivel administrador, cuando un usuario final agregue una nueva figurita a su stock, estas pueden repetirse.**

**El usuario final tiene la posibilidad de pegarFigurita en su álbum. Si un usuario final intenta pegar una figurita que ya había pegado previamente, el sistema debe arrojar la excepción “FiguritaRepetida”.**

**Por último, el sistema debe permitir el intercambio de figuritas entre distintos usuarios. Si por alguna razón, un usuario intenta intercambiar una figurita que no posee o que YA HABIA PEGADO, el sistema debe lanzar la excepción “FiguritaNoDisponible”.**

**Imagen que contiene texto, marcador, parado, lado

Descripción generada automáticamente**

**Imagen 1**

**Condiciones de aprobación.**

* **Se considerará desaprobado todo proyecto que:**
  1. **no compile, o,**
  2. **no se identifiquen los atributos necesarios para resolver lo solicitado, o,**
  3. **los tipos de datos elegidos para los atributos no sean los adecuados, o,**
  4. **no cumpla con las condiciones mínimas descriptas a continuación.**
* **Se considerará aprobado, aquel examen que ejecute de manera correcta al menos el 70% basado en la ponderación de los siguientes test:**

|  |  |
| --- | --- |
| **queSePuedaCrearUnaFigurita** | **5%** |
| **queSePuedaCrearUnAdministrador** | **5%** |
| **queSePuedaCrearUnUsuarioFinal** | **5%** |
| **queUnAdministradorPuedaAgregarUnaFigurita** | **5%** |
| **queUnUsuarioFinalPuedaAgregarUnaFigurita** | **5%** |
| **queLasFiguritasAgregadasDeFormaDesordenadaQuedenOrdenadas** | **5%** |
| **queUnAdministradorNoPuedaAgregarUnaFiguritaExistente** | **5%** |
| **queUnUsuarioFinalSiPuedaAgregarFiguritasExistentes** | **5%** |
| **queUnUsuarioFinalPuedaPegarUnaFigurita** | **20%** |
| **queUnUsuarioFinalNoPuedaPegarUnaFiguritaRepetida** | **10%** |
| **queSePuedaRealizarElIntercambioDeFiguritasEntreDosUsuariosFinales** | **20%** |
| **queNoSePuedaIntercambiarUnaFiguritaDeUnUsuarioQueNoLaTenga** | **5%** |
| **queNoSePuedaIntercambiarUnaFiguritaDeUnUsuarioQueYaLaHayaPegado** | **5%** |
|  | **100%** |