

Ejercicio sobre TDD (Test Driven Development)

Un *array* asociativo consiste en una tabla de pares <clave, valor> tal como se ilustra en la Figura 1, siendo tanto la clave como el valor cadenas de caracteres.

Un *array* asociativo soporta las operaciones que se enumeran a continuación:

- **void put(String clave, String valor):** crea una nueva entrada (un nuevo par) en la tabla. Si la clave existía toma el nuevo valor.
- **String get(String clave):** Busca la clave en la tabla y devuelve el valor asociado; si la clave no existe eleva una excepción **UndefinedKeyException**.
- **String getOrElse(String clave, String valorPorDefecto):** Busca la clave en la tabla y devuelve el valor asociado; si la clave no existe devuelve **valorPorDefecto**.
- **boolean containsKey(String clave):** Devuelve **true** si la clave existe y **false** en caso contrario.
- **boolean remove(String clave):** Borra el par asociado a la clave, devolviendo **true** o **false** según la operación haya tenido éxito o no.
- **int size():** Devuelve el número de entradas del *array* asociativo.

Clave	Valor
"nombre"	"luis"
"apellido"	"martin"
"dni"	"12345567P"
"pais"	"España"

Figura 1. Ejemplo de array asociativo

Aplicando TDD, desarrolla paso a paso una clase **AsocArray** que implemente un *array* asociativo representado mediante una estructura simplemente enlazada.

La evolución del desarrollo debe quedar **reflejada paso a paso en un repositorio git** alojado en GitHub. Para ello:

1. Comienza con un *commit* inicial que solo contenga la clase de test **AsocArrayTest** vacía.
2. Cada vez que escribas un nuevo test, realiza un *commit* seguido de un *push* para reflejar el estado actual del desarrollo. El mensaje de *commit* debe empezar por el prefijo "**green:**". Por ejemplo:

green: una tabla recién creada está vacía

3. Cada vez que escribas código de producción para pasar un test, realiza de nuevo *commit* seguido de un *push*. El mensaje de *commit* debe empezar por el prefijo "**red:**". Por ejemplo:

red: una tabla recién creada está vacía

4. Si en alguna ocasión tienes que refactorizar el código de producción o los tests, vuelve a dejar constancia mediante *commit* y *push*. El mensaje de *commit* debe empezar por el prefijo "**refactor:**". Por ejemplo:

refactor: añadido un @Before para crear un fixture común

Si el TDD ha requerido de T tests y R refactorizaciones, el número total de *commits* del repositorio git debe ser $2 \cdot T + R + 1$.

Al finalizar el TDD, usa el *plugin* **EclEmma** para comprobar si tus tests alcanzan el 100% de *branch coverage* y *complexity coverage*. Si no alcanzas alguna cobertura, añade el número mínimo de tests para alcanzarla.