



2021

Profesores:

Martín Domínguez
Barragán, Pablo.
Madoery, Federico
Amarillo, Leandro

Diseño e Implementación de Estructuras de Datos

Guía 06 – Excepciones – Comparaciones - Lambdas

Contenido

Guía 06 – Excepciones – Comparaciones	2
Importar proyecto a Eclipse	3
Descripción del problema	6
Paso 01 – Capturar excepciones	7
Paso 02 – Implementar los métodos de la clase alumno.....	7
Paso 03 – Implementar los métodos de la clase Curso	7
Paso 04: Crear el método main en la clase APP para probar el sistema.....	8
Paso 05: Crear excepciones personalizadas	8
Paso 06: Modificar el método main	8

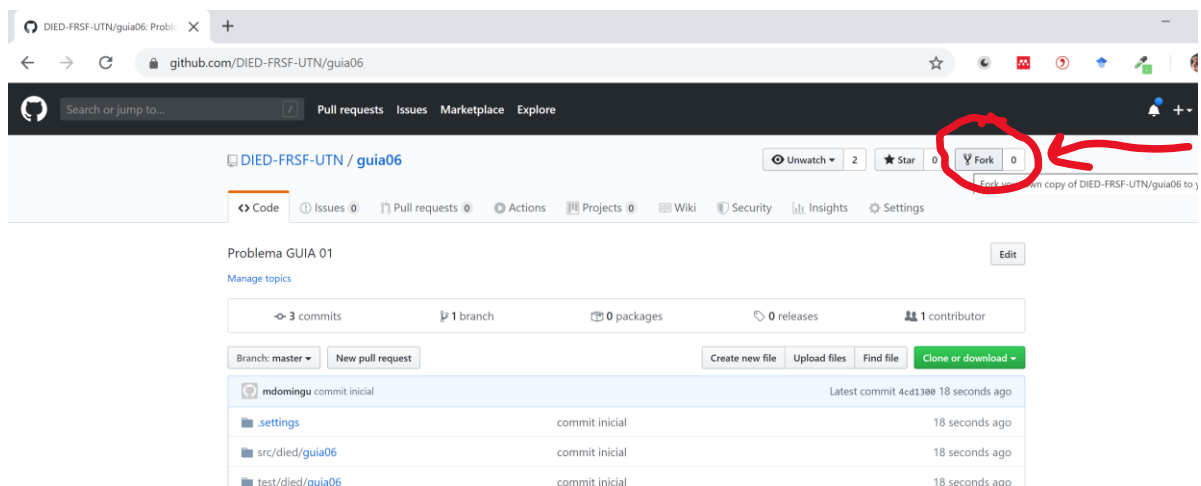
Guía 06 – Excepciones – Comparaciones

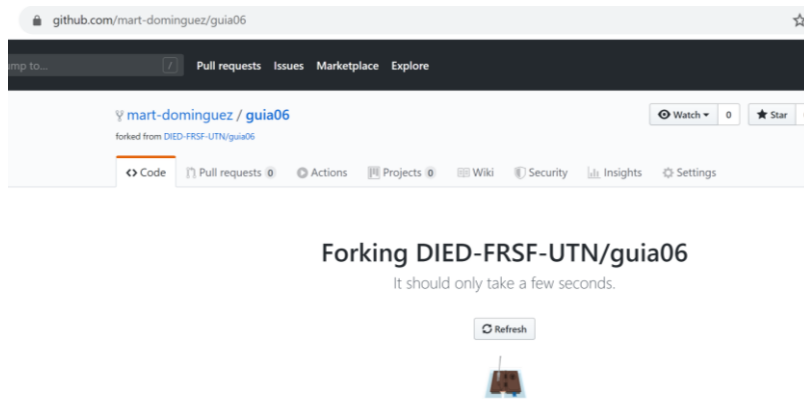
Se entrega un proyecto iniciado en ECLIPSE, en el siguiente repositorio GitHub.

Deberá realizar un “fork” del mismo (esto significa copiar el proyecto que se le entrega al espacio del repositorio de su usuario de github) y luego importarlo en eclipse.

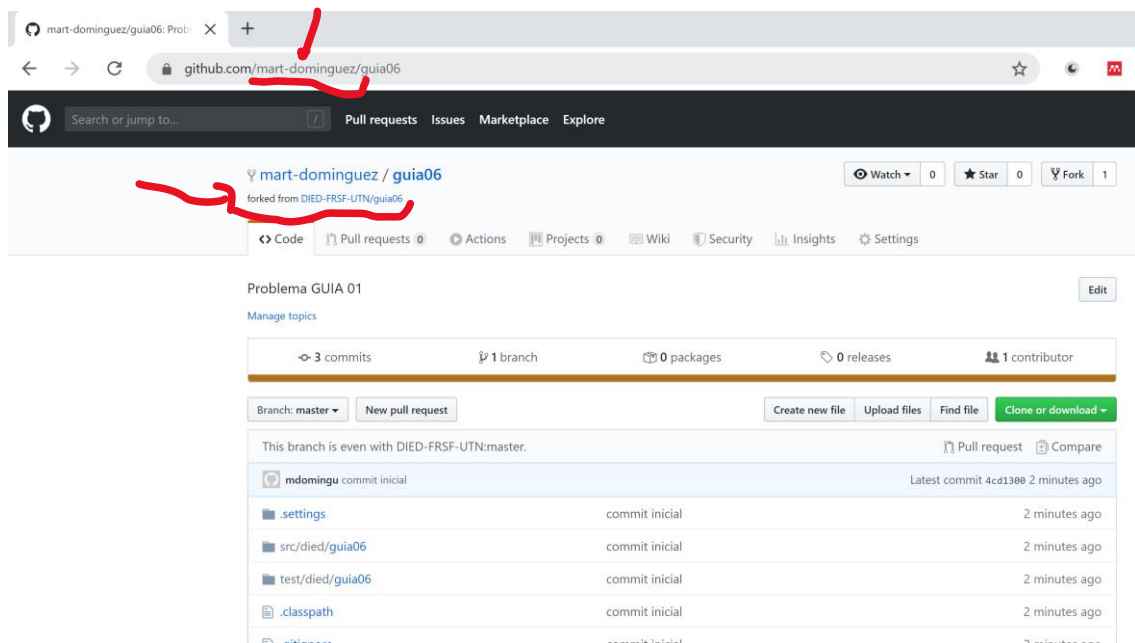
URL del repositorio <https://github.com/DIED-FRSF-UTN/guia06>

En primer lugar, realizar el **FORK**





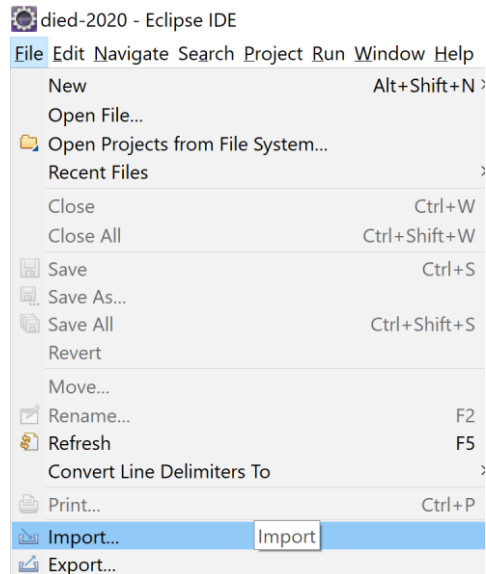
Luego de unos segundos tendrá la copia del repositorio dentro de su usuario de github



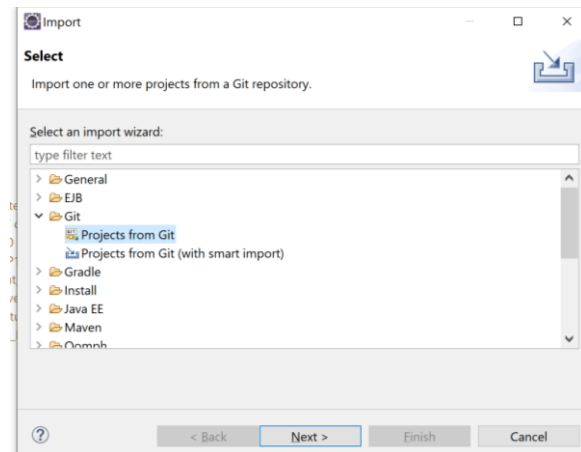
Importar proyecto a Eclipse

Para importar el proyecto git a eclipse, esta vez usaremos una funcionalidad visual de eclipse.

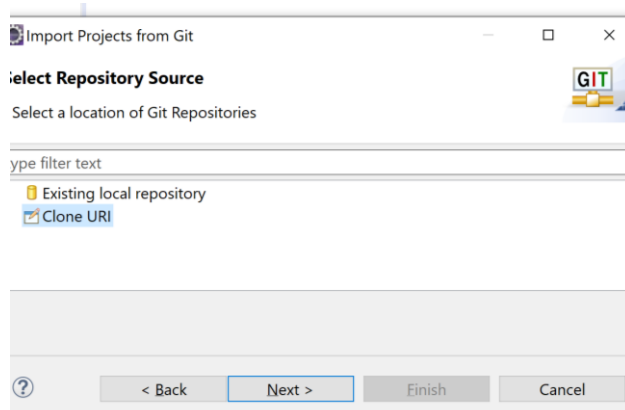
Seleccionar "File" → "Import"



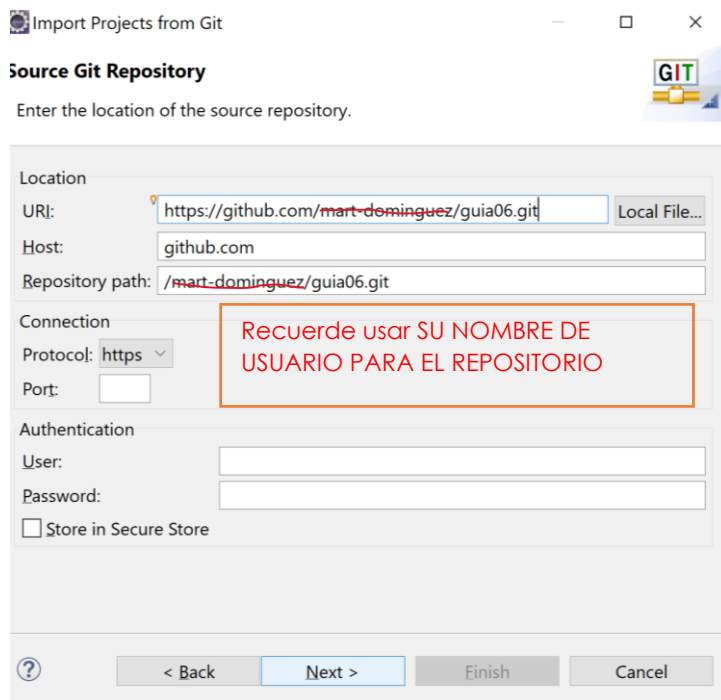
Luego seleccionar "projects from git"



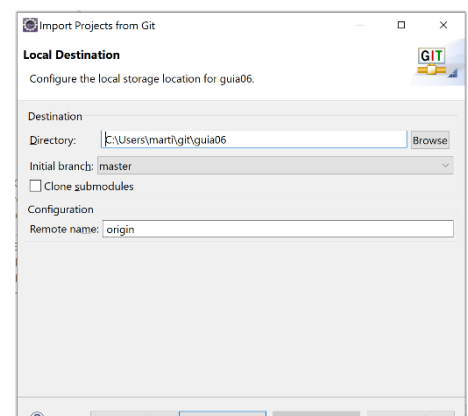
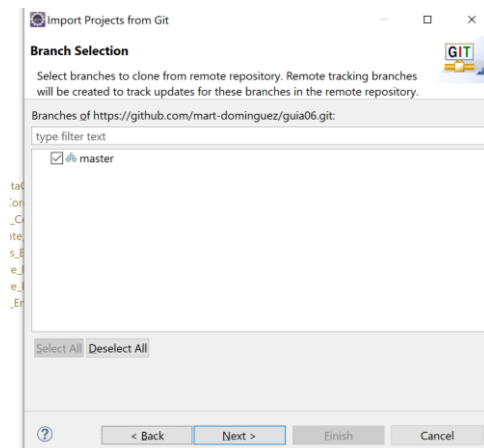
Seleccionar la opción clonar URI



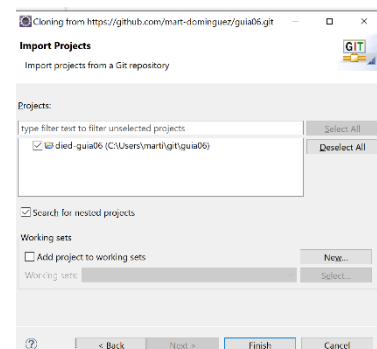
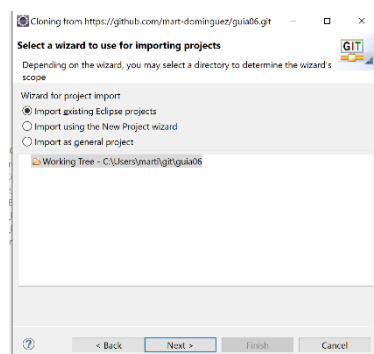
Luego indicar la URL del repositorio GIT de **SU USUARIO**



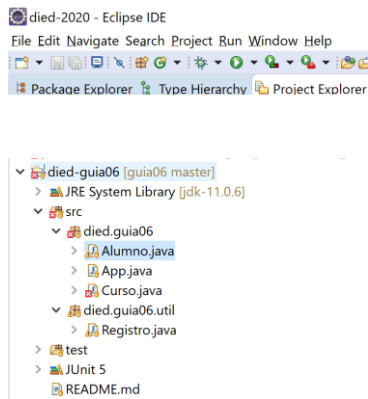
Luego seleccione la rama master (**en este proyecto solo trabajaremos en la rama master**) y **a continuación** Elegir el directorio local donde guardará el proyecto



Y finalmente importar el Proyecto a eclipse



Deberá ver el Proyecto en eclipse de la siguiente manera



Descripción del problema

Deberá modelar el caso de una organización que brinda carreras de capacitación a alumnos a través de cursos a los cuales los alumnos se pueden inscribir si cumplen con las situaciones previstas.

De cada Curso se guarda un valor numérico como Identificador (id), el nombre y el ciclo lectivo. Además, cada curso conoce la cantidad de créditos que otorga a quienes lo aprueban. Por otra parte, cada curso tiene una cantidad de créditos mínimo que el alumno tiene que haber obtenido (aprobandando otros cursos) para poder inscribirse. Algunos cursos no requieren créditos mínimos para inscribirse por lo que dicho valor es de cero (0). Además, todo curso tiene un cupo, y cuando el mismo es alcanzado no se pueden inscribir más alumnos.

Por otra parte, de los métodos que se espera que pueda resolver un curso son:

- Inscribir un alumno: Este método, verifica si el alumno se puede inscribir y si es así lo agrega al curso, y además agrega el curso a la lista de cursos en los que está inscripto el alumno y retorna el valor booleano verdadero. En caso contrario retorna falso y no agrega el alumno a la lista de inscriptos ni el curso a la lista de cursos en los que el alumno está inscripto. Para poder inscribirse un alumno debe
 - o a) tener como mínimo los créditos necesarios
 - o b) el curso debe tener cupos disponibles
 - o c) un alumno puede estar inscripto en simultáneo a no más de 3 cursos del mismo ciclo lectivo.
- Imprimir un curso: se debe poder imprimir el listado de inscriptos ordenados alfabéticamente, por número de libreta universitaria, o por cantidad de créditos obtenidos.

Además, en cada método de la clase curso "registrar" se invoca a una clase que deja registros de auditoria en un archivo del sistema operativo.

A su vez el alumno se modela como una clase que tiene los atributos nombre y nro. de libreta universitaria. A su vez cada alumno una vez que le es aceptada la inscripción a un curso, se le agrega dicho curso a su lista de cursos que está

cursando. Por otro lado, cada vez que aprueba un curso, se quita el curso de la lista de cursos que está cursando y se agrega a la lista de cursos aprobados.

El comportamiento **mínimo** esperado para la clase alumno es:

- Poder determinar la cantidad de créditos que posee (esto se determina como la suma de los créditos que otorgan los cursos que ya aprobó).
- Poder registrar que se inscribe a un curso.
- Poder registrar que aprobó un curso (y ya no está cursando)

Paso 01 – Capturar excepciones

La clase `died.guia06.util.Registro` posee el método `registrar` que lanza una excepción, deberá capturarla en los métodos de la clase `Curso`.

Paso 02 – Implementar los métodos de la clase alumno

Además de los métodos de negocio, debe

- Implementar el método `equals`, estableciendo que 2 alumnos son iguales si tienen el mismo número de libreta universitaria.
- Implementar la interface `comparable`, estableciendo que la comparación de 2 alumnos se hace alfabéticamente por el atributo `nombre`.
- Además, puede crear otros métodos que considere necesarios (getters / setters / constructores).

Paso 03 – Implementar los métodos de la clase Curso

Implementar los métodos de negocio de un curso y además puede crear otros métodos que considere necesarios (getters / setters / constructores).

Además, para poder implementar el ordenamiento de los alumnos inscriptos por distintos criterios, podría necesitar crear nuevas clases que implementen la interface `Compartor<Alumno>`

Paso 04: Crear el método main en la clase APP para probar el sistema.

Debe crear los cursos y alumnos necesarios para representar las situaciones descriptas.

Además, en caso de ser necesario el método main debe manejar la gestión de errores.

Paso 05: Crear excepciones personalizadas

Agregar a curso el método

- **public** void inscribirAlumno(Alumno a) { ... }

Este método ejecutará correctamente si el alumno se pudo inscribir. Caso contrario lanzará una excepción personalizada para cada una de las siguientes situaciones:

- Una excepción si el alumno no tiene los creditos requeridos
- Una excepción que represente que el curso tiene el cupo cubierto
- Una excepción que represente que el alumno ya tiene todas las materias de cursado regular.
- Además, este método debe capturar la excepción de entrada salida de la clase registro y en caso de que ocurra relanzarla como una nueva excepción del tipo personalizado "RegistroAuditoriaException"

Paso 06: Modificar el método main

Modificar el método main para que use el método para inscribir alumnos que gestiona excepciones, capture las excepciones correspondientes e imprima por consola los mensajes apropiados.

Problema 2

1. Se necesita modelar información de una **liga** de futbol que tiene hasta 10 equipos. Para ello, se registran **Equipos**. De cada Equipo se conoce el nombre, y la lista de **jugadores**. Cada equipo puede tener inscriptos hasta

21 Jugadores. La liga, registra en una lista de **Fechas**, cada uno de los **Partidos**, donde se enfrentan el equipo local y el equipo visitante. De cada partido se registra información del Día y hora del partido, y quienes fueron los jugadores 11 jugadores titulares y 7 suplentes que firmaron la planilla ese día. De cada partido también se registran **Eventos**, indicando el tipo de evento, que jugador está asociado a dicho evento y en que minuto del partido fue el evento.

2. Los tipos de evento que se registran son
 - a) Infracción: jugador que la cometió jugador que la recibió.
 - b) Tarjeta Amarilla o Roja: jugador que la recibió y una descripción con el motivo
 - c) Tiro de esquina: Jugador que lo ejecutó y costado en que se produjo (derecho o izquierdo)
 - d) Tiro Libre: Jugador que lo ejecutó y una descripción con el resultado de la acción.
 - e) Gol: jugador que lo marco, jugador que asistió, y en caso de ser un gol en el arco propio, se marca con un flag en true
3. Crear la clase Jugador donde se registre información del jugador, (nombre, dni, fecha de nacimiento, posición, altura, peso, si patea con el pie izquierdo o derecho)
4. Crear la clase Equipo que permita registrar:
 - a) Información del equipo
 - b) Agregar hasta 21 jugadores
 - i. .
5. Crear la clase Liga a la que se le pueda:
 - a) Agregar, quitar, consultar, y modificar Equipos.
 - b) Agregar una lista con las fechas a disputar y los encuentros que se disputan en cada fecha.
 - c) Registrar en cada fecha los partidos.
 - d) Registrar en cada partido los eventos que se han producido.

Métodos a implementar

- a) Imprimir la tabla de posiciones y guardarla en un archivo con la fecha de impresión.
- b) Imprimir la tabla de goleadores.
- c) Imprimir la lista de jugadores por el siguiente orden
 - a. Partidos jugados
 - b. Goles realizados
 - c. Amonestaciones recibidas.
 - d. Altura
 - e. Peso
- d) Imprimir la lista de equipos ordenados por
 - a. Con mayor cantidad de goles a favor
 - b. Con mayor diferencia de gol
 - c. Con mayor cantidad de amonestados y expulsados
 - d. Con mayor cantidad de infracciones.