

**Eduardo De la Coteria
Martínez 1ºDAW**

Entornos de desarrollo

**Tema 12
Diagramas II**

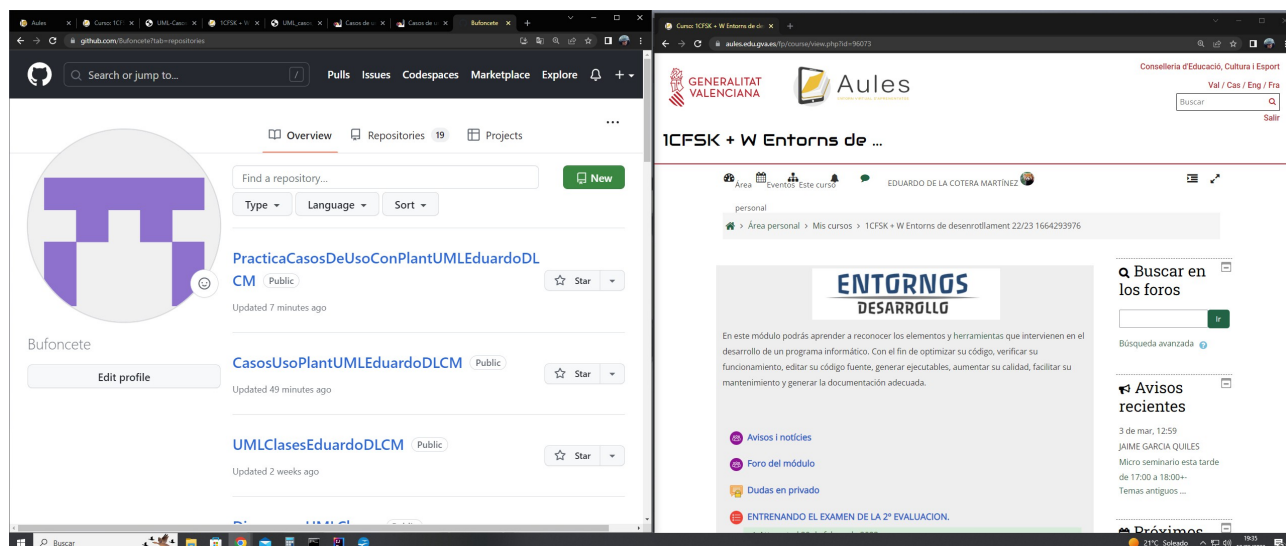
**Práctica Casos de uso
con PlantUml
Memoria**

<https://github.com/Bufoncete/PracticaCasosDUsoPlantUml.git>

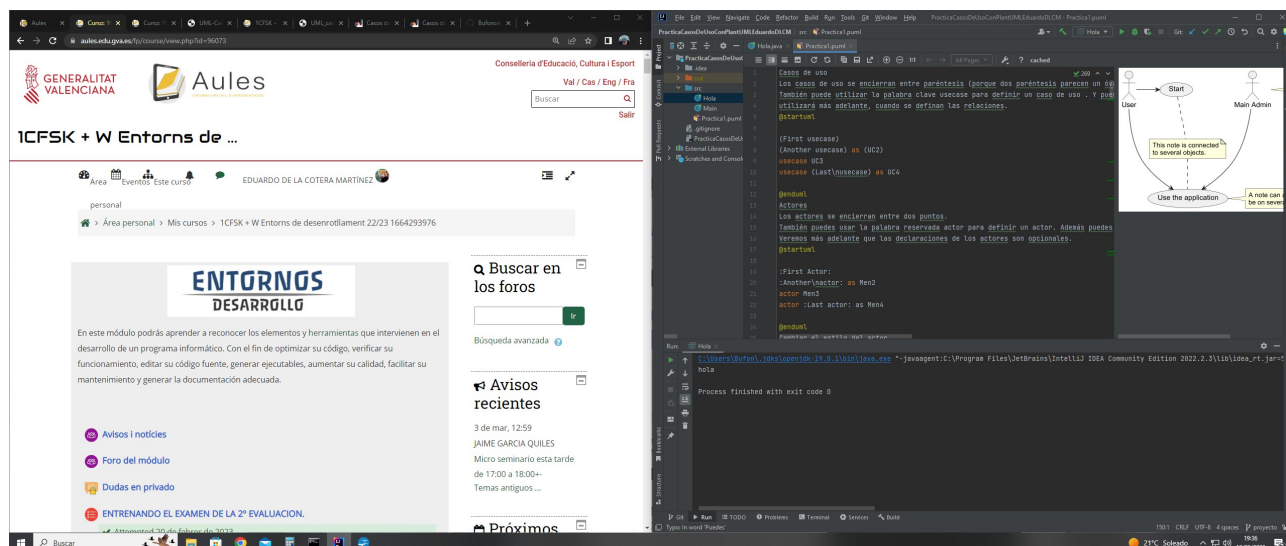
A continuación presento la memoria de la práctica por puntos, siguiendo los diferentes pasos del enunciado.

Para acreditar la autenticidad del autor de las capturas, añadiré junto a las mismas una captura de mi cuenta de Aules abierta para evitar confusiones.

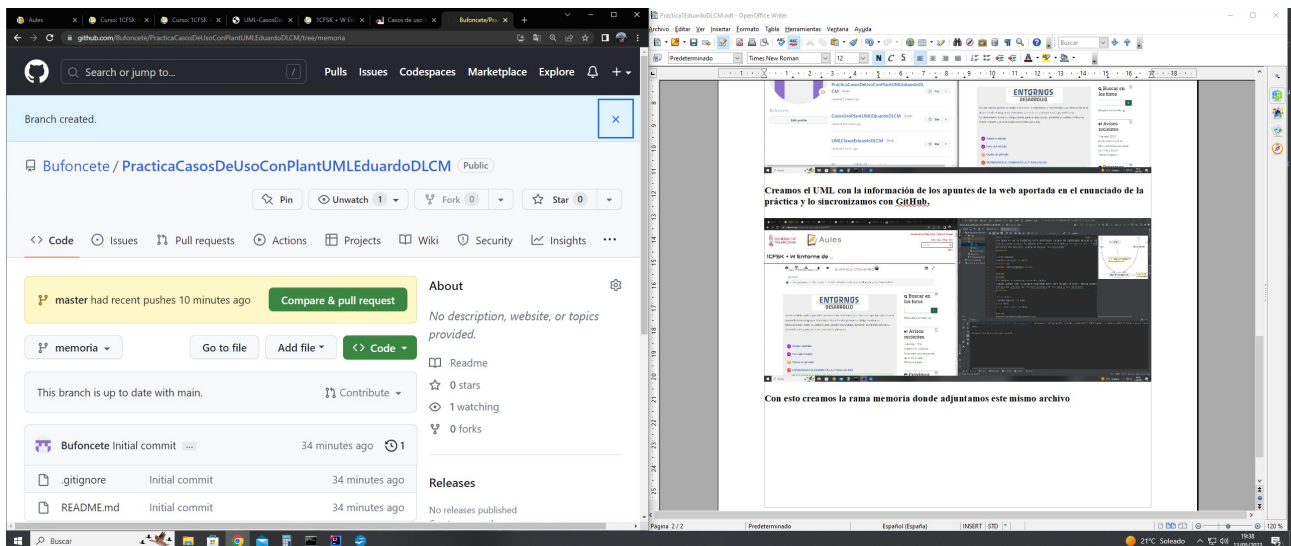
1º Creamos el repositorio y lo clonamos en IntelliJ



Creamos el UML con la información de los apuntes de la web aportada en el enunciado de la práctica y lo sincronizamos con GitHub.



Con esto creamos la rama memoria donde adjuntamos este mismo archivo.



Y con esto terminamos el ejercicio.

//Código completo de UML

Casos de uso

Los casos de uso se encierran entre paréntesis (porque dos paréntesis parecen un óvalo).

También puede utilizar la palabra clave usecase para definir un caso de uso . Y puede definir un alias, utilizando la palabra clave as. Este alias se utilizará más adelante, cuando se definan las relaciones.

```
@startuml
(First usecase)
(Another usecase) as (UC2)
usecase UC3
usecase (Last\nusecase) as UC4
@enduml
```

Actores

Los actores se encierran entre dos puntos.

También puedes usar la palabra reservada actor para definir un actor. Además puedes definir un alias, usando la palabra reservada as. Este alias será usado más adelante, cuando definamos relaciones.

Veremos más adelante que las declaraciones de los actores son opcionales.

```
@startuml
:First Actor:
:Another\nactor: as Men2
actor Men3
actor :Last actor: as Men4
@enduml
```

Cambiar el estilo del actor

Puedes cambiar el estilo del actor de hombre==== palo ==== (==== por defecto ====) a:

un hombre impresionante con el comando skinparam actorStyle awesome;

un hombre hueco con el comando skinparam actorStyle hollow .

Hombre palo (por defecto)

```
@startuml
:User: --> (Use)
"Main Admin" as Admin
"Use the application" as (Use)
Admin --> (Admin the application)
@enduml
```

Hombre==== impresionante ====

```
@startuml
skinparam actorStyle awesome
:User: --> (Use)
"Main Admin" as Admin
```

```

"Use the application" as (Use)
Admin --> (Admin the application)
@enduml
Hombre==== hueco ====
@startuml
skinparam actorStyle Hollow
:User: --> (Use)
"Main Admin" as Admin
"Use the application" as (Use)
Admin --> (Admin the application)
@enduml

Descripción de Casos de uso
Si quiere realizar una descripción en varias líneas, puede usar citas (" ").
También puede usar los siguientes separadores: --..==__. Y puede introducir
títulos dentro de los separadores.
@startuml
usecase UC1 as "You can use
several lines to define your usecase.
You can also use separators.
--
Several separators are possible.
==
And you can add titles:
..Conclusion..
This allows large description."
@enduml

Utilice el paquete
Puede utilizar paquetes para agrupar actores o casos de uso
@startuml
left to right direction
actor Guest as g
package Professional {
    actor Chef as c
    actor "Food Critic" as fc
}
package Restaurant {
    usecase "Eat Food" as UC1
    usecase "Pay for Food" as UC2
    usecase "Drink" as UC3
    usecase "Review" as UC4
}
fc --> UC4
g --> UC1
g --> UC2
g --> UC3
@enduml

Puede utilizar rectangle para cambiar la visualización del paquete
@startuml
left to right direction
actor "Food Critic" as fc
rectangle Restaurant {
    usecase "Eat Food" as UC1
    usecase "Pay for Food" as UC2
    usecase "Drink" as UC3
}
fc --> UC1
fc --> UC2
fc --> UC3
@enduml

Ejemplo básico
Para relacionar actores y casos de uso, la flecha --> es usada.
Cuanto más guiones - en la flecha, más larga será la misma. Puedes añadir una
etiqueta en la flecha, añadiendo el carácter : en la definición de la flecha.
En este ejemplo, puedes ver que User no ha sido definido, y es usado como un

```

```

actor.
@startuml
User -> (Start)
User --> (Use the application) : A small label
:Main Admin: ---> (Use the application) : This is\nyet another\nlabel
@enduml
Extensión
Si un actor/caso de uso extiende a otro, puedes usar el símbolo <|--.
@startuml
:Main Admin: as Admin
(Use the application) as (Use)
User <|-- Admin
(Start) <|-- (Use)
@enduml
Usando notas
Puedes usar las palabras claves: note left of , note right of , note top of ,
note bottom of, para añadir notas relacionadas a un objeto en particular.
También se puede añadir un nota solitaria con la palabra clave note, y después
realacionarla con otro objeto usando el símbolo .. .
@startuml
:Main Admin: as Admin
(Use the application) as (Use)
User -> (Start)
User --> (Use)
Admin ---> (Use)
note right of Admin : This is an example.
note right of (Use)
    A note can also
    be on several lines
end note
note "This note is connected\nto several objects." as N2
(Start) .. N2
N2 .. (Use)
@enduml
Estereotipos
Puedes añadir estereotipos mientras defines actores y casos de uso, usando << y
>>.
@startuml
User << Human >>
:Main Database: as MySql << Application >>
(Start) << One Shot >>
(Use the application) as (Use) << Main >>
User -> (Start)
User --> (Use)
MySql --> (Use)
@enduml
Cambio de dirección de las flechas
Por defecto, los enlaces entre clases tienen dos guiones -- y están orientados
verticalmente. Es posible utilizar un enlace horizontal poniendo un solo guión
(o punto) como este
@startuml
:user: --> (Use case 1)
:user: -> (Use case 2)
@enduml
También se puede cambiar la dirección invirtiendo el enlace:
@startuml
(Use case 1) <.. :user:
(Use case 2) <- :user:
@enduml
También es posible cambiar la dirección de la flecha añadiendo las palabras
clave left, right, up o down dentro de la flecha
@startuml
:user: -left-> (dummyLeft)
:user: -right-> (dummyRight)

```

```

:user: -up-> (dummyUp)
:user: -down-> (dummyDown)
@enduml
Puede acortar la flecha utilizando sólo el primer carácter de la dirección (por
ejemplo, -d- en lugar de -down- ) o los dos primeros caracteres (-do-).
Tenga en cuenta que no debe abusar de esta funcionalidad : Graphviz suele dar
buenos resultados sin retoques.
Y con el left to right direction parámetro
@startuml
left to right direction
:user: -left-> (dummyLeft)
:user: -right-> (dummyRight)
:user: -up-> (dummyUp)
:user: -down-> (dummyDown)
@enduml
Dividiendo los diagramas
La palabra clave newpage divide su diagrama en varias páginas o imágenes.
@startuml
:actor1: --> (Usecase1)
newpage
:actor2: --> (Usecase2)
@enduml
Dirección: de izquierda a derecha
El comportamiento general cuando se construye un diagrama, es top to bottom.
@startuml
'default
top to bottom direction
user1 --> (Usecase 1)
user2 --> (Usecase 2)
@enduml
Puede cambiar a left to right usando el comando left to right direction. En
ocasiones, el resultado es mejor con esta dirección.
@startuml
left to right direction
user1 --> (Usecase 1)
user2 --> (Usecase 2)
@enduml
Personalización (Skinparam)
Puedes usar el comando skinparam para cambiar los colores y las fuentes de los
dibujos
Puedes usar este comando:
En la definición del diagrama, como cualquier otro comando,
En un archivo incluido,
En un archivo de configuración, proporcionado en la consola de comandos o en el
ANT task.
Puedes definir colores y fuentes específicas para los actores y casos de uso
estereotipados.
@startuml
skinparam handwritten true
skinparam usecase {
BackgroundColor DarkSeaGreen
BorderColor DarkSlateGray
BackgroundColor<< Main >> YellowGreen
BorderColor<< Main >> YellowGreen
ArrowColor Olive
ActorBorderColor black
ActorFontName Courier
ActorBackgroundColor<< Human >> Gold
}
User << Human >>
:Main Database: as MySql << Application >>
(Start) << One Shot >>
(Use the application) as (Use) << Main >>
User -> (Start)

```

```

User --> (Use)
MySQL --> (Use)
@enduml
Un ejemplo completo
@startuml
left to right direction
skinparam packageStyle rectangle
actor customer
actor clerk
rectangle checkout {
    customer -- (checkout)
    (checkout) .> (payment) : include
    (help) .> (checkout) : extends
    (checkout) -- clerk
}
@enduml
Business Use Case
You can add / to make Business Use Case.
Business Usecase
@startuml
(First usecase)/
(Another usecase)/ as (UC2)
usecase/ UC3
usecase/ (Last\nusecase) as UC4
@enduml
Business Actor
@startuml
:First Actor:/
:Another\nactor:/ as Man2
actor/ Woman3
actor/ :Last actor: as Person1
@enduml
Change arrow color and style (inline style)
You can change the color or style of individual arrows using the inline
following notation:
#color;line.[bold|dashed|dotted];text:color
@startuml
actor foo
foo --> (bar) : normal
foo --> (bar1) #line:red;line.bold;text:red : red bold
foo --> (bar2) #green;line.dashed;text:green : green dashed
foo --> (bar3) #blue;line.dotted;text:blue : blue dotted
@enduml
Change element color and style (inline style)
You can change the color or style of individual element using the following
notation:
#[color|back:color];line:color;line.[bold|dashed|dotted];text:color
@startuml
actor a
actor b #pink;line:red;line.bold;text:red
usecase c #palegreen;line:green;line.dashed;text:green
usecase d #aliceblue;line:blue;line.dotted;text:blue
@enduml
Display JSON Data on Usecase diagram
Simple example
@startuml
allowmixing
actor Actor
usecase Usecase
json JSON {
    "fruit":"Apple",
    "size":"Large",
    "color": ["Red", "Green"]
}

```

@endum1