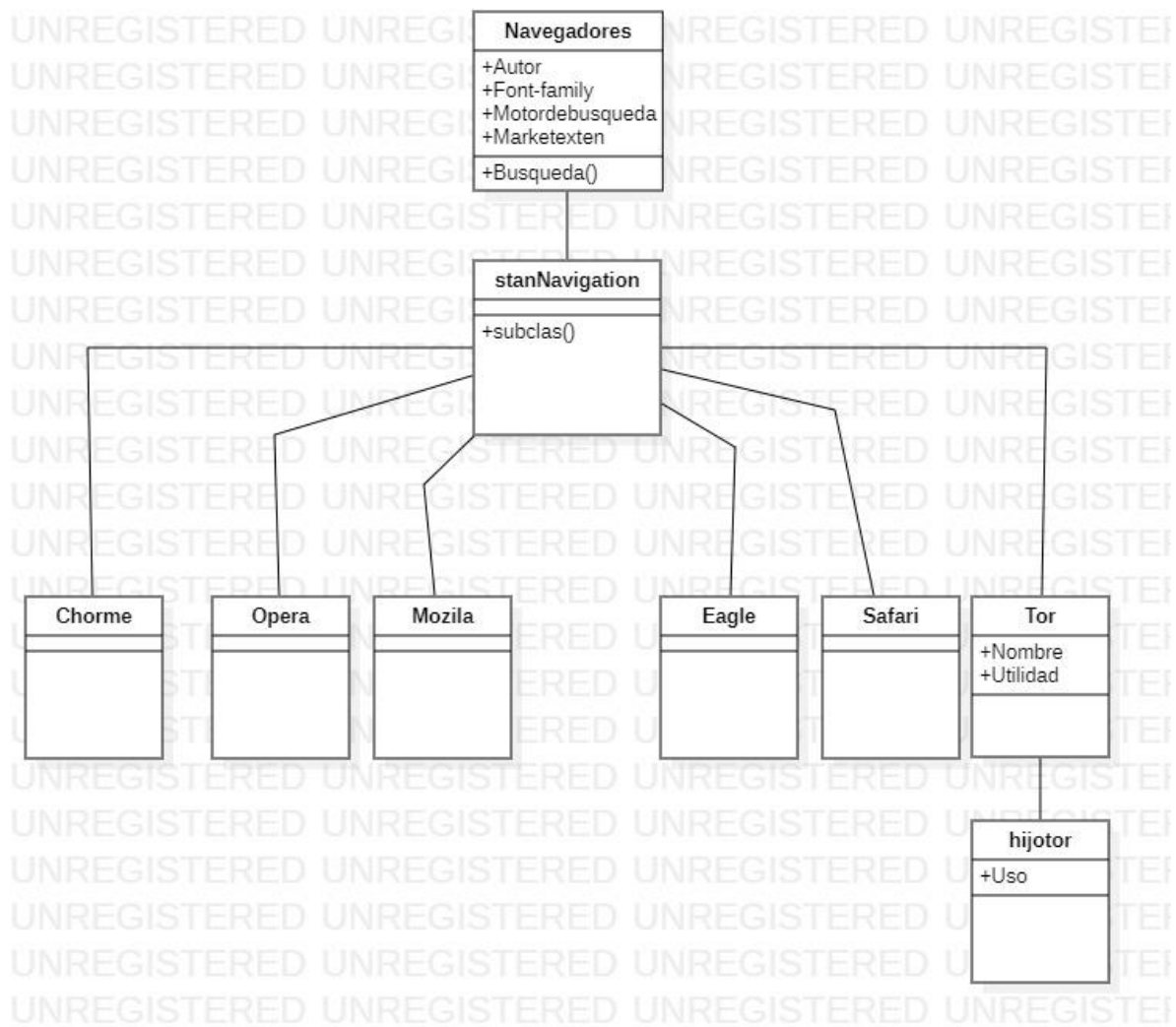


Programación orientada a objetos – Herencia

Johan Pachon Melo – ADSO

Instructor: Ing. Jose David Montesino

Diagrama de clase



Código en Python

```
#Navegadores super-class
#Superclass - herencia con parametros
class Navegadores:

    #Constructor
    def __init__(self,autor,font_family,
motorBusqueda,marketPlace):
        self.autor=autor
        self.font_family=font_family
        self.motorBusqueda=motorBusqueda
        self.marketplace=marketPlace

    #METODOS DE UTILIDADES
    def busqueda(self):
        print("Buscando....")

    def verificacionConexion(self):
        print(".....verifica")
        print(".....Utilidad")

    def errorNavegador(self):
        print("File not found")

class standNavigation(Navegadores):
    def subclas(self):
        print("Otra clase mas")

#Instancias de objetos con init
safari=standNavigation("Pedro pineda",
"Cursiva","Bin","playStore")
chorme=standNavigation("Andes perez",
"Roboto","Bin","operaStore")
opera=standNavigation("Manzana quintero",
"Arial","Bin","operaStore")
eagle=standNavigation("Samuel luque",
"Roboto","Bin","microsoftStore")

#SALIDAS METODOS

safari.verificacionConexion()
eagle.busqueda()
opera.errorNavegador()

#Salidas
print("Navegadores atributos: ")
print("safari: ", safari.autor,safari.
font_family,safari.motorBusqueda,safari.
marketplace)
print("chorme: ", chorme.autor,chorme.
font_family,chorme.motorBusqueda,chorme.
marketplace)
print("opera: ", opera.autor,opera.
font_family,opera.motorBusqueda,opera.
marketplace)
print("eagle: ", eagle.autor,eagle.
font_family,eagle.motorBusqueda,eagle.
marketplace)
print(
"-----
--"
)

#Clases de herencia sin init
class tor():
    Nombre="onion"
    utilidad="Cebolla"
    autor="Anonimos"

class hijotor(tor):
    Uso="Sub-terraneoweb"

#instancia sin init
objetos2=hijotor()

#Herencia sin init
print("Utilidad ", objetos2.Nombre,
objetos2.Uso,objetos2.utilidad)
```