

# DEKRA DIGITAL

**Presentación Final, Mayo 2022**

**Donde se  
encuentra**

C.Álamos, 7

29012 Málaga Centro



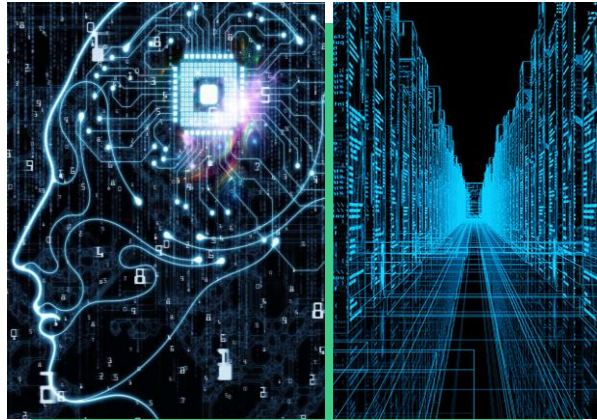
# Departamentos

## Inteligencia Artificial

Dedicado a la generación de algoritmos capaces de predecir y trabajar para facilitar todo tipo de tareas de grandes datos

Tutor: Xavier Valero

Integrantes:  
Alejandro Cueto y David Antúnez



## Big Data

Trabajando para gestionar datos a gran escala de gran valor y muy complejos.

Tutor: Cristian Mena Acedo

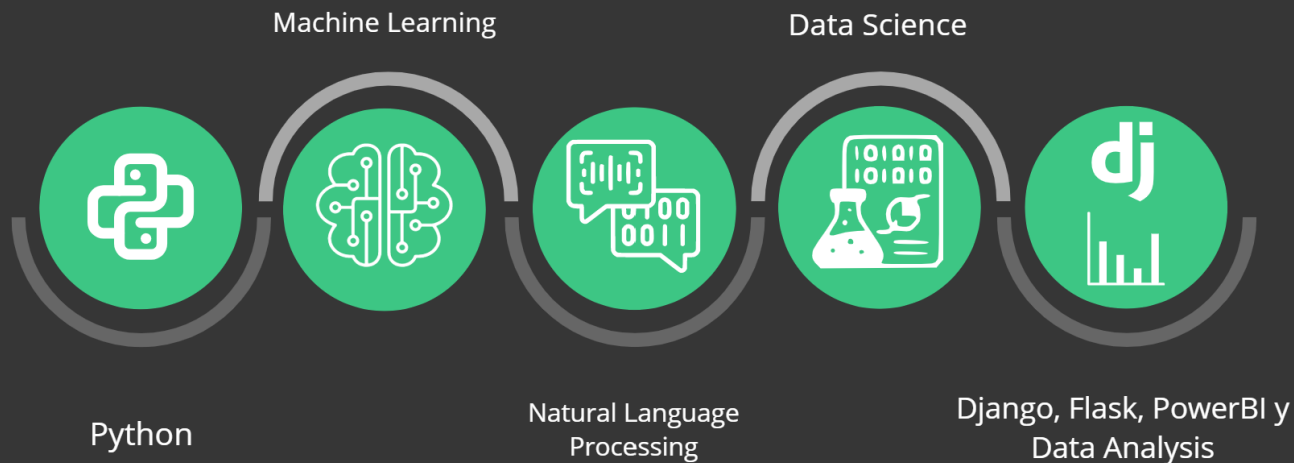
Integrantes:  
Jose Antonio Benitez y Alberto Moreno

# Inteligencia Artificial

## Camino a seguir

### Secuencia de aprendizaje

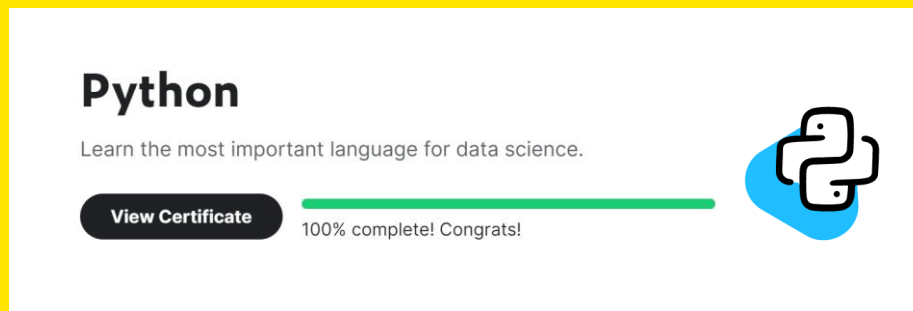
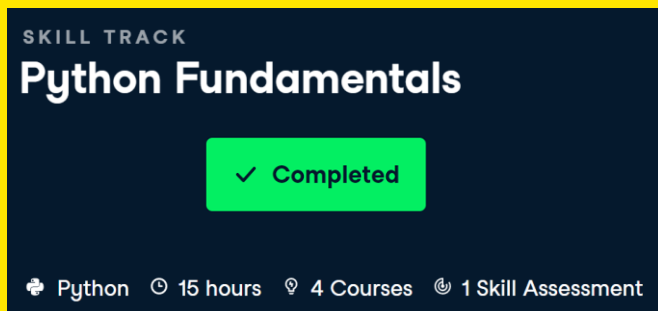
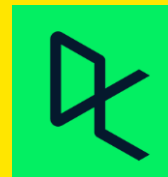
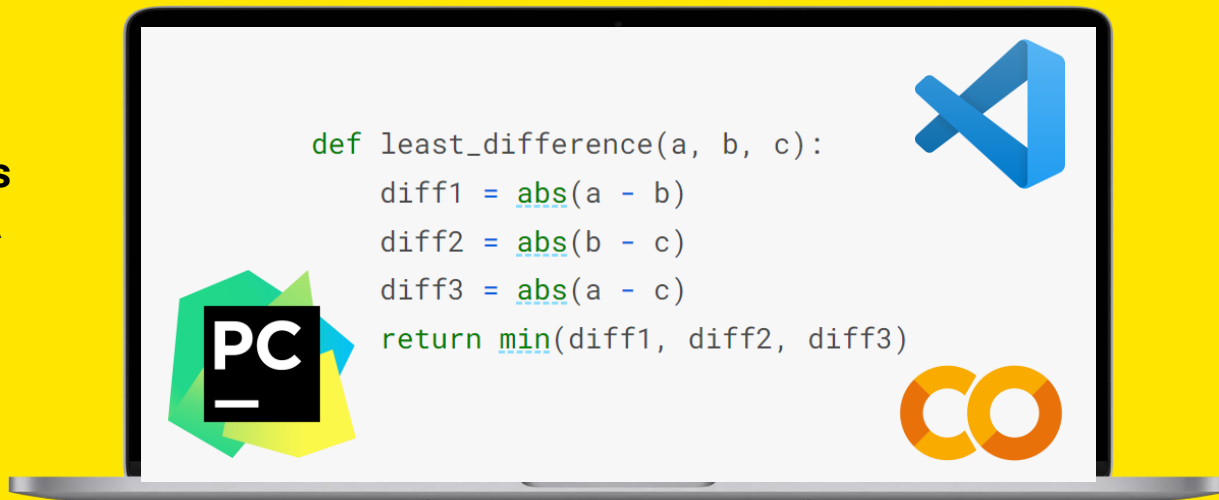
---



# PYTHON

## Uso de librerías orientadas a IA

Como por ejemplo  
Pandas, Matplotlib,  
Sklearn, nltk...



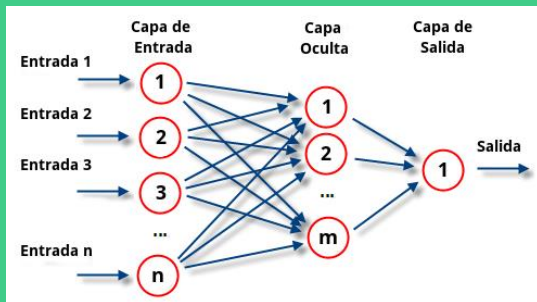
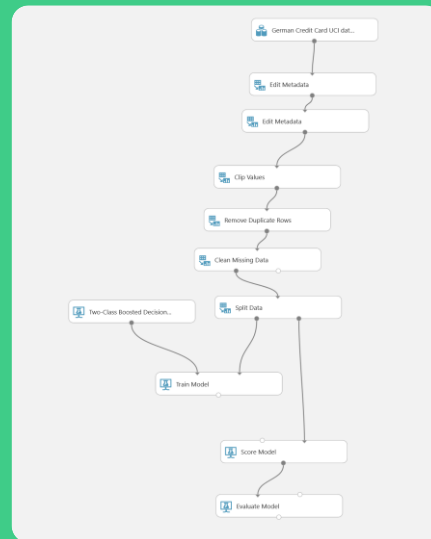
# Machine Learning

## Cognitive Services

Hemos aprendido en qué ramas se divide el Machine Learning y nos hemos centrado en los servicios cognitivos, como el procesamiento de lenguaje natural

## Data Classification

Hemos utilizado grandes datasets para generar algoritmos de predicción



age	gender	chest_pain	blood_pressu	cholesterol	slope	vessels_num	thall	covid_risk	
63	male	3	145	233	0	0		1 High	
37	male	2	130	250	0	0		2 High	
41	female	1	130	204	2	0		2 High	
56	male	1	120	236	2	0		2 High	
57	female	0	120	354	2	0		2 High	
57	male	0	140	192	1	0		1 High	
56	female	1	140	294	1	0		2 High	
44	male	1	120	263	2	0		3 High	
52	male	2	172	199	2	0		3 High	
57	male	2	150	168	2	0		2 High	
54	male	0	140	239	2	0		2 High	
48	female	2	130	275	2	0		2 High	
49	male	1	130	266	2	0		2 High	
64	male	3	110	211	1	0		2 High	

## Convertir el texto a valores clasificables

De esta forma podemos hacer análisis de los sentimientos del texto, clasificarlos...



```
def remove_symbols(df):
    pattern = '^[\0-9a-zA-Z]'
    for row in df['Description']: # Row's iteration and space removing
        df['Description'] = df['Description'].str.replace(pattern, ' ', regex=True)

def lower_case(df):
    df["Description"] = df["Description"].str.lower() ## Convert Description to lower case

# TOKENIZE DATASET
e_stopwords = stopwords.words("english")
df['Description'] = [i for i in df['Description'].str.split() if i not in e_stopwords]
```

Description	Labels
<p><b>**14-820845**</b>injured right wrist when involved in a accident on 271 nor</p> <p><b>**17-133500**</b>ee was walking on the dock looking through some freight closing trailer door after delivery left leading roller off track jammed door the ee finished his last move when he turned and ducked down to get out</p> <p><b>**14-861011**</b>the ee was using a crow bar when it slipped off of the nail slipped on ice fell on lunch box</p> <p>closing trailer door and shut it on my right hand. cut 3 fingers and fractured left wrist and right foot when exiting out of the trailer. stepped on dock during drilling operations, the c&amp;j service rig was preparing to make a core walking up driver side entrance. pickup truck was parked in front of dump strain or injury by-reaching;i needed to put a one check valve in a unit. the ee was hooking up a trailer and was cranking the dollies with a handle i was walking from moves back to the station. i slipped on a patch of ice as ee was climbing into the truck bed ee said after a couple minutes ee felt contact with-fire or flame;employee was vacuuming off coal dust from the driver stepped out of rear of trailer to the ground onto uneven ground. a feb 21 2017, throughout the course of the day a new worker with brand strain or injury by-lifting;i was lifting a ramp on trailer #r38564 when i fell</p>	

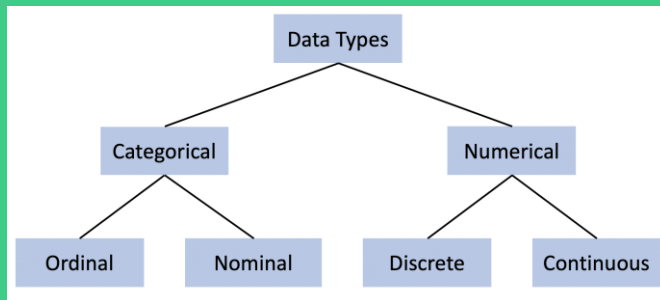




# DATA SCIENCE

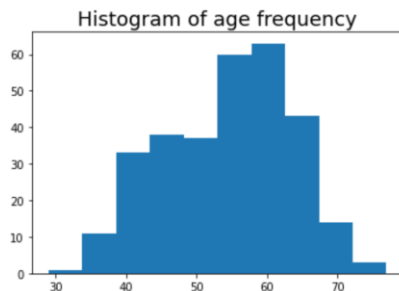
## Preprocesar, analizar y exponer datos

Hemos aprendido los  
pasos principales a  
seguir para el análisis del  
dataset previo a la  
creación del modelo



### ▼ Histograms for data analysis

```
[ ] import matplotlib.pyplot as plt
plt.hist(df['age'])
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of age frequency', loc = 'center', fontsize = 18)
plt.show()
```

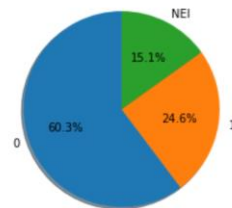


```
[ ] import matplotlib.pyplot as plt

a_pie = [zero_count, one_count, nei_count]
labels = ['0', '1', 'NEI']

plt.pie(a_pie, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)

[<matplotlib.patches.Wedge at 0x7f55c05ed8d0>,
 <matplotlib.patches.Wedge at 0x7f55c06013d0>,
 <matplotlib.patches.Wedge at 0x7f55c0601f50>],
[Text(-1.0433479486139872, -0.34846098508008755, '0'),
 Text(1.0872305292426, -0.16712204008704437, '1'),
 Text(0.5029307808382035, 0.9782947560349463, 'NEI')],
[Text(-0.5690988810621748, -0.1900696282255023, '60.3%'),
 Text(0.5930348341323273, -0.0911574764111151, '24.6%'),
 Text(0.27432588045720185, 0.533615321473607, '15.1%')]]
```



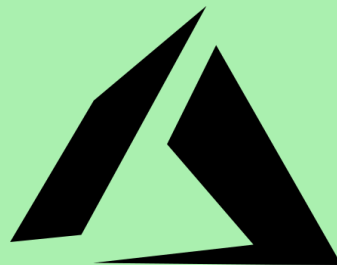


# APPLIED MODELS



## Django

Aprendizaje básico y creación de programas básicos para comprender conceptos



## Azure Sentiment Analysis

Uso del servicio MLaaS de Microsoft

## Flask

Creación de una Web Application de Azure para subida de datos a Azure Storage



## PowerBI

Creación de EDA's más detallados y complejos usando DAX

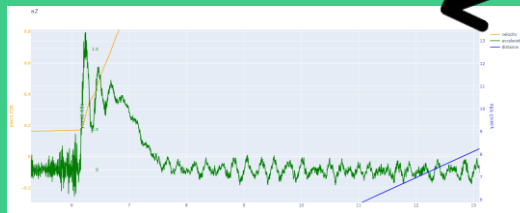
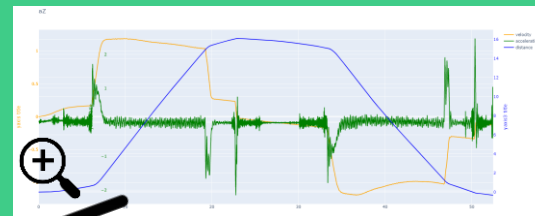
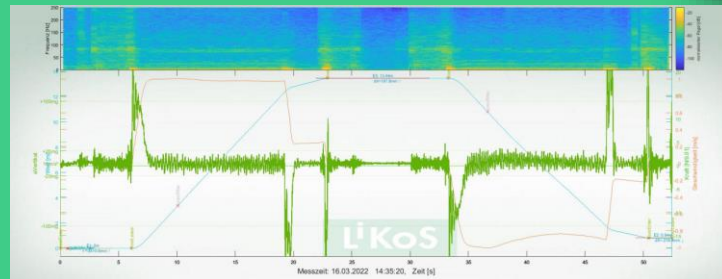


# DEKRA HACKATON 2022

Car Background Removal



Interactive Liftcheck Report



DEKRA  
DIGITAL

# ASOCIACIÓN DE TRABAJO-ASIGNATURA

## PROGRAMACIÓN

- Uso de Python
- Utilización de librerías de Python orientadas al procesamiento de datos y la creación de algoritmos predictivos (scikitlearn, nltk, pandas, numpy...)
- Creación de algoritmos de ML y NLP

## BASE DE DATOS

Trabajo con grandes datasets

Entendimiento de la organización de los datos en dataframes y series

## SISTEMAS INFORMÁTICOS

- Uso de docker para Azure Web Apps
- Servicios de Azure (MLaaS, App Service, Storage...)

## ENTORNOS DE DESARROLLO

- Uso de la metodología Scrum
- Trabajo en proyectos por parejas y por equipos (en la hackaton)
- Realización Catch Ups diariamente
- Utilización de IDEs para Python (Google Collab, jupyter notebook, Pycharm, Vscode...)

## LENGUAJE DE MARCAS

- Creación de páginas con Bootstrap
- Uso de HTML, CSS y JS para creación de frontend de apps
- Utilización de Django y Flask para Web Apps

# CODE SNIPPETS

## LOGISTIC REGRESSION

```
# USING LOGISTIC REGRESSION MODEL | accuracy score : 0.7705882352941177
from sklearn.linear_model import LogisticRegression
lr_clf = LogisticRegression(random_state=0)
# setting up the Pipeline
vec_clf = Pipeline([('vectorizer', vectorizer), ('pac', lr_clf)])
# fitting the model
vec_clf.fit(X_train, y_train)
joblib.dump(vec_clf, '/lrClassifier.pk1', compress=3) # for an easily exporting process
```

## DECISION TREE

```
# USING DECISION TREE MODEL | accuracy score : 0.7823529411764706
from sklearn.tree import DecisionTreeClassifier
dt_clf = DecisionTreeClassifier(random_state=0)
# setting up the Pipeline
vec_clf = Pipeline([('vectorizer', vectorizer), ('pac', dt_clf)])
# fitting the model
vec_clf.fit(X_train, y_train)
joblib.dump(vec_clf, '/dtClassifier.pk1', compress=3) # for an easily exporting process
```

# CODE SNIPPETS

## RANDOM FOREST

```
# USING RANDOM FOREST CLASSIFIER | accuracy score : 0.788235294117647
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# depths_list = [25, 30, 32, 35, 40, 42, 50] CALCULATING THE BEST DEPTHs
# for depths in depths_list:
#     rfc_clf = RandomForestClassifier(max_depth=depths, random_state=0)
#     vec_clf = Pipeline([('vectorizer', vectorizer), ('pac', rfc_clf)])
#     vec_clf.fit(X_train, y_train)
#     y_preds=vec_clf.predict(X_test)
#     print(accuracy_score(y_preds,y_test))

rfc_clf = RandomForestClassifier(max_depth=2, random_state=0)
# setting up the Pipeline
vec_clf = Pipeline([('vectorizer', vectorizer), ('pac', rfc_clf)])
# fitting the model
vec_clf.fit(X_train, y_train)
joblib.dump(vec_clf, '/rfcClassifier.pk1', compress=3) # for an easily exporting process
```

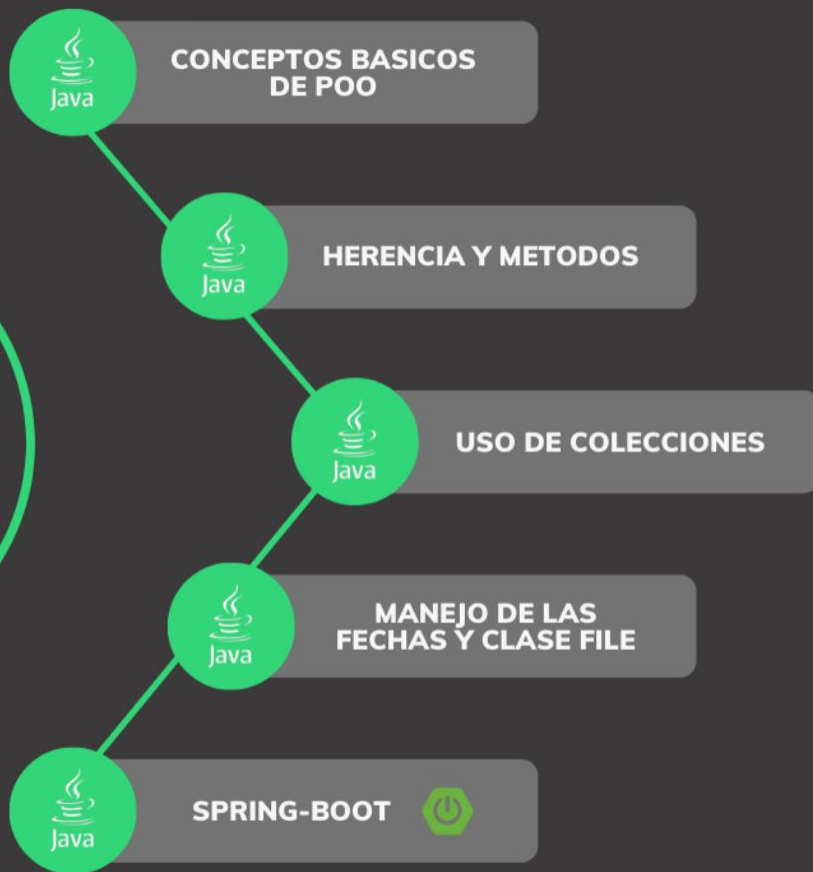
## LINEAR SVC

```
svm_clf = svm.LinearSVC(C=0.1)

vec_clf = Pipeline([('vectorizer', vectorizer), ('pac', svm_clf)])

vec_clf.fit(X_train, y_train)
joblib.dump(vec_clf, '/svmClassifier.pk1', compress=3)
```

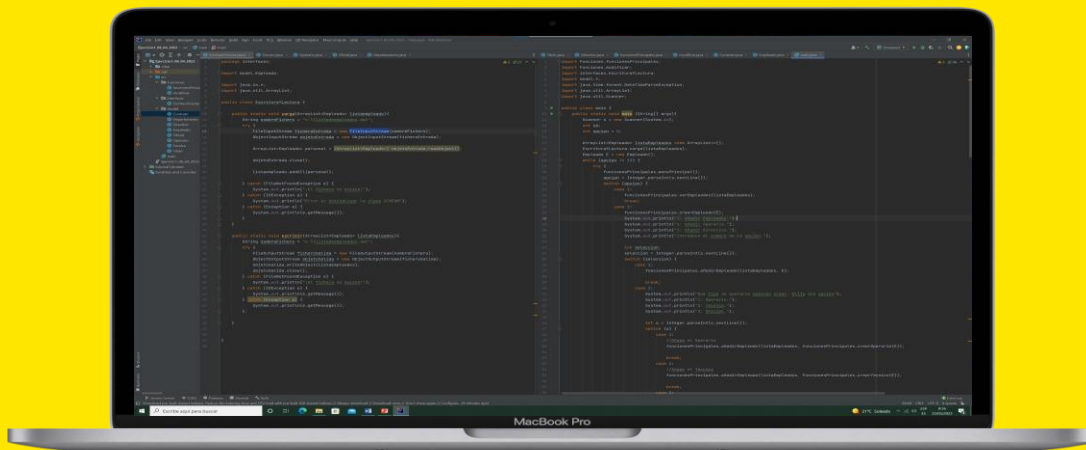
# BIG DATA



# Java

## *Inicio de la Dual.*

Durante la primera semana, estuvimos asimilando conceptos de java, ya que nuestro nivel no estaba a nivel empresarial.





# Java

## Portal Gestion Dual

Nos asignaron la tarea de realizar un programa tipo CRUD en el cual trabajaríamos durante toda la Dual.

## Herencia y Métodos

Una vez las bases y los conceptos claves estaban comprendidos, nos pusimos a estructurar el proyecto que nos propusieron, para ello jugamos con las distintas clases que nos propusieron y su herencia.

[illegible]

```
package model;

public class Empleado {
    private String codigo;
    private String nombre;
    private String apellido1;
    private String apellido2;

    /*= CONSTRUCTOR =*/

    public Empleado(String codigo, String nombre, String apellido1, String apellido2) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.apellido1 = apellido1;
        this.apellido2 = apellido2;
    }

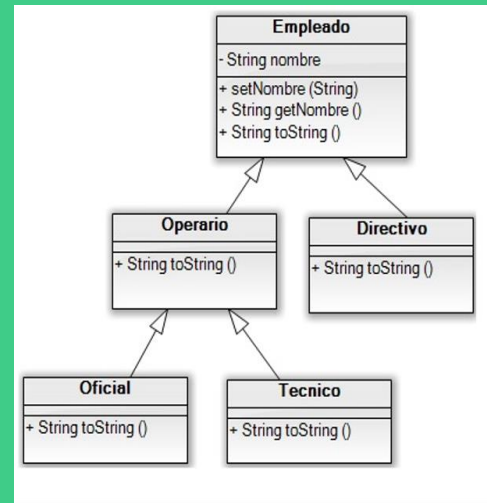
    /*= GETTERS Y SETTERS =*/

    public Empleado() {}

    public String getCodigo() {return codigo;}

    public void setCodigo(String codigo) {this.codigo = codigo;}

    public String getNombre() {return nombre;}
}
```



# Java

## Uso de Colecciones

Estuvimos viendo los distintos tipos de colecciones existentes y probando su uso, para posteriormente implementarlas en el proyecto.



## Manejo de Fechas

Usando las clases  
LocalDateTime y Period  
pudimos trabajar y  
operar con fechas y  
periodos de tiempo  
entre ellas.

```
public class Contrato extends Empleado {
    private int idContrato;
    private String tipoContrato;
    private LocalDateTime fechaInicioContrato;
    private LocalDateTime fechaFinContrato;

    public Contrato() {
    }

    public int getIdContrato() {return idContrato;}

    public void setIdContrato(int idContrato) {this.idContrato = idContrato;}

    public String getTipoContrato() {return tipoContrato;}

    public void setTipoContrato(String tipoContrato) {this.tipoContrato = tipoContrato;}

    public LocalDateTime getFechaInicioContrato() {return fechaInicioContrato;}

    public void setFechaInicioContrato(LocalDateTime fechaInicioContrato) {
        this.fechaInicioContrato = fechaInicioContrato;
    }

    public LocalDateTime getFechaFinContrato() {return fechaFinContrato;}

    public void setFechaFinContrato(LocalDateTime fechaFinContrato) {this.fechaFinContrato = fechaFinContrato;}

    public Contrato(int id, int idEmpleado, String tipoContrato, LocalDateTime fechaInicioContrato, LocalDateTime fechaFinContrato) {
        super(id); //Se setea el empleado
        this.idContrato = idContrato;
        this.tipoContrato = tipoContrato;
        this.fechaInicioContrato = fechaInicioContrato;
        this.fechaFinContrato = fechaFinContrato;
    }

    @Override
    public String toString() {
        return "El contrato N°.....de idContrato = " + getIdContrato() + " tiene un tipo = " + getTipoContrato() + " fecha de inicio = " + getFechaInicioContrato() + " fecha de fin = " + getFechaFinContrato();
    }
}
```

```
package model;

import java.io.Serializable;
import java.time.LocalDate;

public class Titulo implements Serializable {
    int id;
    String nombre;
    LocalDate expedicion;

    public Titulo(int id, String nombre, LocalDate expedicion) {
        this.id = id;
        this.nombre = nombre;
        this.expedicion = expedicion;
    }

    public Titulo() {
    }

    public int getId() {return id;}

    public void setId(int id) {this.id = id;}

    public String getNombre() {return nombre;}

    public void setNombre(String nombre) {this.nombre = nombre;}

    public LocalDate getExpedicion() {return expedicion;}

    public void setExpedicion(LocalDate expedicion) {this.expedicion = expedicion;}

    @Override
    public String toString() {
        return "Titulo: " + getNombre() + " Expedicion: " + getExpedicion() + "W";
    }
}
```

```
//Instancia una fecha.
public static LocalDate darAltaFecha(String fechaEntrada) {
    DateTimeFormatter formato = DateTimeFormatter.ofPattern("dd/MM/yyyy");
    LocalDate fecha = LocalDate.parse(fechaEntrada, formato);
    return fecha;
}
```

```
//Da la edad de un empleado
public static int edad(LocalDate fechaEntrada) {
    LocalDate actual = LocalDate.now();

    Period periodo = Period.between(fechaEntrada, actual);

    return periodo.getYears();
}
```

# Java

## *Empleo de la clase File*

Tratamos la lectura y escritura de objetos dentro de ficheros. Gracias a ello pudimos cargar un ArrayList usando esos objetos, como si fuera una base de datos.

```
public static void carga(ArrayList<Empleado> listaempleado){
    String nombreFichero = "c:\\listadeempleados.dat";
    try {
        FileInputStream ficheroEntrada = new FileInputStream(nombreFichero);
        ObjectInputStream objetoEntrada = new ObjectInputStream(ficheroEntrada);

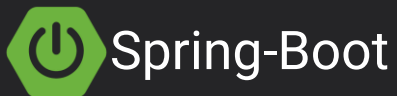
        ArrayList<Empleado> personal = (ArrayList<Empleado>) objetoEntrada.readObject();

        objetoEntrada.close();

        listaempleado.addAll(personal);

    } catch (FileNotFoundException e) {
        System.out.println("El fichero no existe!");
    } catch (IOException e) {
        System.out.println("Error al inicializar la clase STREAM");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

public static void escribir(ArrayList<Empleado> listaEmpleados){
    String nombreFichero = "c:\\listadeempleados.dat";
    try {
        FileOutputStream ficheroSalida = new FileOutputStream(nombreFichero);
        ObjectOutputStream objetoSalida = new ObjectOutputStream(ficheroSalida);
        objetoSalida.writeObject(listaEmpleados);
        objetoSalida.close();
    } catch (FileNotFoundException e) {
        System.out.println("El fichero no existe!");
    } catch (IOException e) {
        System.out.println(e.getMessage());
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```



Spring Boot es un framework de Java que nos permite crear aplicaciones auto contenidas, con esto nos podemos olvidar de la arquitectura y enfocarnos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar nuestro servicio o aplicación a un servidor de aplicaciones y enfocarnos desarrollo.



# Asociación de trabajo – asignatura.

- Java. Vista en profundidad y consolidación de conocimientos sobre el lenguaje.
- Spring/Spring-Boot

## Programación

- Creación de base de datos en memoria H2
- Uso de sentencias LDD y LMD

## Base de Datos

- IDE IDEA IntelliJ. Utilizado para la realización del proyecto.
- PostMan. Para probar el back-end de la aplicación

## Entornos de Desarrollo



Experiencia Personal:



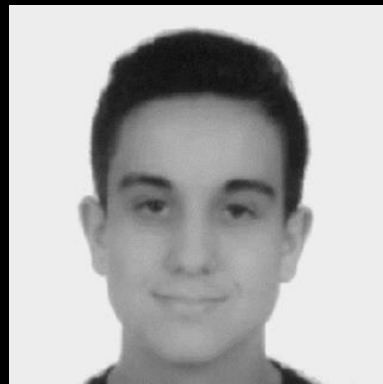
## REALIZADO POR



Alberto Moreno González



Alejandro Cueto Jiménez



David Antúnez Pérez



Jose Antonio Benitez

DEKRA DIGITAL

innovating safety