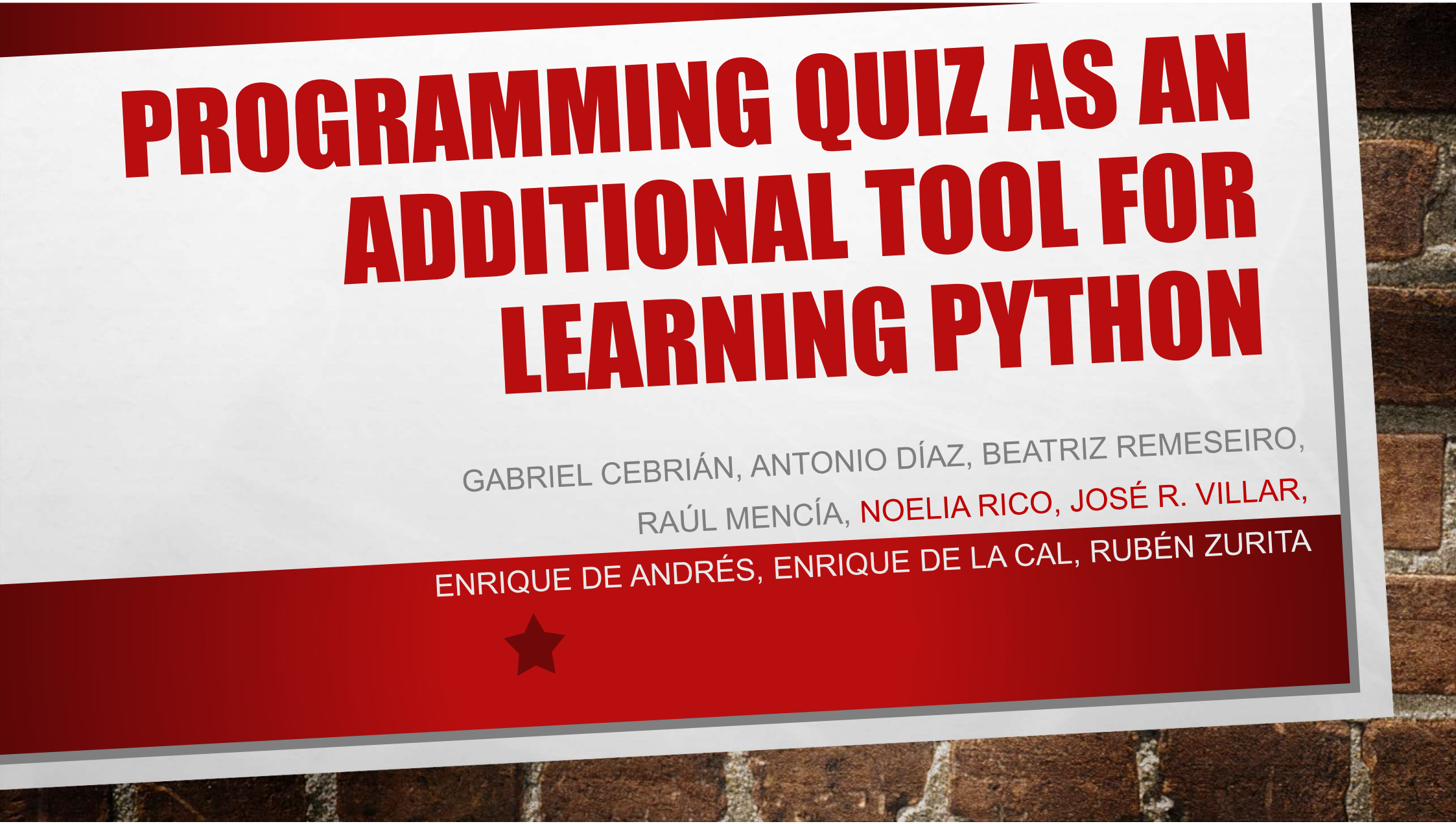
PROGRAMMING QUIZ AS AN ADDITIONAL TOOL FOR LEARNING PYTHON

GABRIEL CEBRIÁN, ANTONIO DÍAZ, BEATRIZ REMESEIRO,
RAÚL MENCÍA, NOELIA RICO, JOSÉ R. VILLAR,
ENRIQUE DE ANDRÉS, ENRIQUE DE LA CAL, RUBÉN ZURITA



EXPERIENCES IN GAMIFICATION AND TEACHING ENHANCEMENT

- This talk covers recent proposals for enhancing the teaching of the subject **Computer Basis** in different degrees taught at Engineering Schools with the University of Oviedo.
- On the one hand, the gamification test that was proposed to the students during the first semester of this year is introduced.
- On the second hand, then some experiences in gathering feedback from students and also from staff members are discussed.

- COMPUTER BASIS IN UNIOVI
- GAMIFICATION AND THE PAY-OFF
  - DEGREES INVOLVED
  - THE GAME
  - OBTAINED RESULTS
- FEEDBACK IS WORTHY
  - FINEXT2019
  - WELL-SORTED STAGES
  - CONCLUSIONS

COMPUTER BASIS IN UNIOVI

- Computer basis (CB) is a common subject for all the engineering degrees in the University of Oviedo's catalog of undergraduate degrees.
- This subject is included in the first term, with a total charge of 6 credits.
- The content of this subject includes
  - i) basic concepts and knowledge -information representation, logical operators, etc.-,
  - ii) a light introduction to computers -both hardware and software-, with an overview to operating systems,
  - iii) an introduction to databases design, and
  - iv) an introduction to computer programming.

- DOUBLE DEGREE IN CIVIL ENGINEERING AND ENGINEERING OF MINING AND ENERGY RESOURCES
- DEGREE IN CIVIL ENGINEERING
- DEGREE IN ENGINEERING OF MINING AND ENERGY RESOURCES
- GRADO EN INGENIERÍA DE ORGANIZACIÓN INDUSTRIAL (OFERTA CURSO 2019 -2020)
- DEGREE IN ENGINEERING OF INDUSTRIAL TECHNOLOGIES
- DEGREE IN ENGINEERING OF MINING TECHNOLOGIES
- DEGREE IN ELECTRIC ENGINEERING
- DEGREE IN INDUSTRIAL ELECTRONIC AND AUTOMATIC ENGINEERING
- GRADO EN INGENIERÍA EN GEOMÁTICA
- DEGREE IN ENGINEERING OF TECHNOLOGIES AND SERVICES FOR TELECOMMUNICATION
- DEGREE IN FORESTRY AND NATURAL RESOURCES
- DEGREE IN GEOMATIC ENGINEERING AND TOPOGRAPHY
- DEGREE IN SOFTWARE COMPUTER ENGINEERING
- DEGREE IN IT ENGINEERING
- DEGREE IN MECHANICAL ENGINEERING
- DEGREE IN CHEMICAL ENGINEERING
- DEGREE IN INDUSTRIAL CHEMISTRY ENGINEERING
- DEGREE IN MARINE
- DEGREE IN NAUTICAL AND MARINE TRANSPORTATION

COMPUTER BASIS STRUCTURE

- Lectures
  - 28 → 1-hour sessions.
  - Time for each chapter varies from one degree to another.
  - In Computer Science, the contents vary slightly.
  - In the remaining engineering degrees: there is a more or less common Schedule.
- Labs
  - 14 → 2-hours sessions.
  - Plus 2 → 1-hour sessions for solving doubts and extra exercises.
  - In general: 2 EXCEL, 2 ACCESS, then PYTHON.
  - But in Computer Science: 2 EXCEL, then PYTHON.

Nowadays, each degree can have its own FLAVOUR of Computer Basis

BRIEFLY SPEAKING, COMPUTER BASIS IS A

https://www.stockio.com/free-icon/security-icons-danger

- For the students
  - Out of opus
  - Complex
  - Too many things, some barely overview
- For the teaching staff
  - Too many things, some barely overview
  - Discouraged students
  - Too many degrees with the exactly same contents?

SO, WE CAN BE HEROES! JUST FOR ONE DAY

- At least, that should have been what the universities authorities thought when designing this subject.
- There should be something we can do to endure and fight the inherent problems with Computer Basis!
- Here are two actions developed during this year

https://www.beaconresort.com.au/superherofest

https://coreaxis.com/when-to-use-gamification/

https://jalacoste.com/lecciones-liderazgo-socrates

GYMKHANA: GAMING AND LEARNING

http://gonewiththetwins.com/new/good-the-bad-and-the-ugly-1968/

https://www.cerpsur.org/trabajo-en-equipo-juegos-de-integracion/13896229468294062280

https://allhtaccess.info/25-gif-images-related-to-programming/

THE AIM

ENCOURAGE THE STUDENTS TO WORK ON THE SUBJECT IN DAILY BASIS

THE MEAN

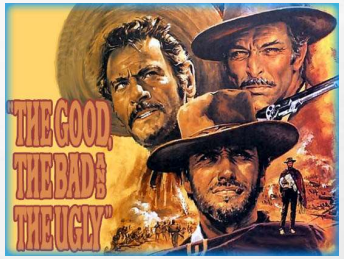
A PROGRAM DEVELOPED IN PYTHON THAT ASKS THE STUDENTS THE SOLUTION TO SEVERAL PROGRAMMING PROBLEMS

AND

THE UGLY

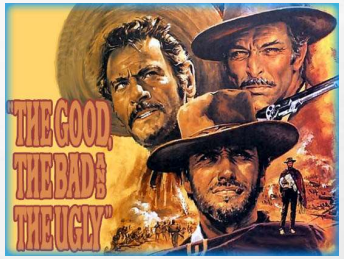
THE NUMBER OF TRIES AND THE NUMBER OF SOLVED EXERCISES WERE USED FOR ASSESSMENT

GYMKHANA: THE AIM



- To help the students reinforce their programming skills using gamification.
  - Also, to introduce the same activities in all the tutorial sessions.
- It was OPTIONAL.
- The higher the effort, the better the reward.
- Up to 5 staff members and three degrees:
  - Computer Science
  - Chemistry Engineering
  - Industrial Technologies Engineering
- University of Oviedo's Project on Innovation in Education PINN-18-A-010

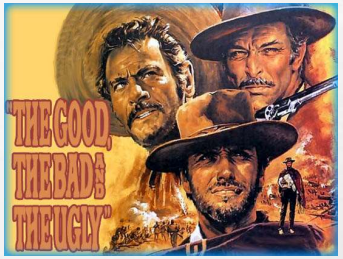
GYMKHANA: THE MEANS



- Basically, each student should connect to a ssh server, where a Python program runs automatically. It was a Shell-oriented program.
- This Python program asks the student to solve each of the designed exercises, one by one.
- An exercise is a randomly generated instance of a problem.
  - If the student successfully solves the exercise, then the next exercise is shown
  - Otherwise, the student must try again
- For each student, a log with all the solved exercises and the number of tries for each was registered.
  - In this release, the time spent in solving an exercise was not considered.
- The final mark:

$$P = \sum_{i=1}^{n} \frac{1}{N_i}$$

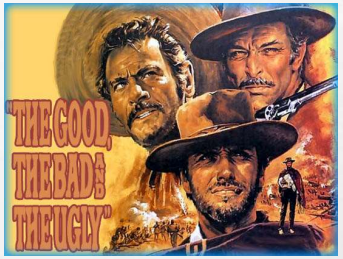
GYMKHANA: THE MEANS




```python
def main():
    if (check_deadline_finished()):
        finished()
        time.sleep(3)
        return

    uo_number = raw_input('Type your UO number (format: UOxxxxx): ')
    if (uo_number.lower()=='register'):
        uo_number = register()

    if (uo_number.lower()=='prof_results'):
        calculate_results()
        raw_input('Press enter to finish...')
        time.sleep(1)
        return

    if (uo_number==''):
        return

    uo_number = check_uo(uo_number)
    while (uo_number==''):
        uo_number = raw_input('Wrong UO number format.\nType your UO number (format: UOxxxxx): ')
        uo_number = check_uo(uo_number)

    if not (os.path.exists(uo_number + '.txt')):
        print 'User does not exist.'
        time.sleep(2)
        return

    uo_pass = hashlib.md5(uo_number.encode()).hexdigest()[:5]
```

```python
    solved_ex = parse_file(uo_number+'.txt')
    user_file = open(uo_number+'.txt', 'a')
    random.seed(int(uo_number[2:]))
    time.sleep(0.85)
    result = 1

    ejercicios = [ex1_ant, ex2_ant, ex3_ant, ex4_ant]  # iniciales
    ejercicios += [exB1, gcm_ex1, exN1, exN2]  # binario, operadores basicos, variables
    ejercicios += [jv_ex1, exN7, exN8, gcm_ex2, gcm_ex3, exB2]  # operadores logicos, condiciones, if-else
    ejercicios += [gcm_ex4, gcm_ex5, gcm_ex6, exB3, gcm_ex7, exB4]  # bucles
    ejercicios += [jv_ex3, jv_ex4, jv_ex5, jv_ex6, jv_ex7, jv_ex9, exN3, exN4, exB6, exB7]  # listas
    ejercicios += [jv_ex2, jv_ex8, jv_ex10, exN5, exN6, exB5, exB8]  # cadenas
    current_ex = 1

    while solved_ex < len(ejercicios) and (result != -2 and result != -1):
        result = ejercicios[solved_ex](solved_ex>=current_ex)
        check_result(result, user_file)
        solved_ex = solved_ex + (result > 0)
        current_ex = current_ex + 1

    if (solved_ex==len(ejercicios)):
        print '\nCongratulations, you have completed the Gymkhana!'
        print 'Your score is ' + str(calculate_single_result(uo_number)) + '\n'
        time.sleep(3)
```

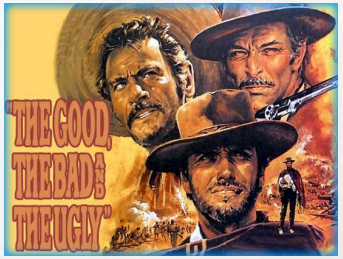
GYMKHANA: THE MEANS



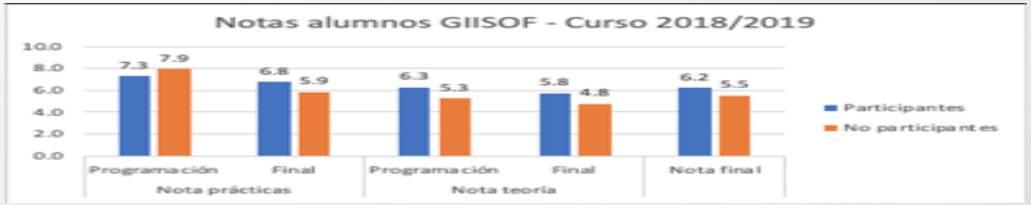

```python
#8.- comprobar si el valor contenido en una variable está en una lista
def jv_ex8_params():
    v = chr(random.randint(ord('a'), ord('m')))
    l = chr(random.randint(ord('n'), ord('z')))
    while l == v:
        l = chr(random.randint(ord('n'), ord('z')))
    s = "Artificial amateurs aren't at all amazing\n\tAnalytically, I assault, animate things"
    return [v, l, s]


def jv_ex8(slvd):

    p = jv_ex8_params()

    if slvd:
        return -3

    ansRE =  re.compile("^%s([ ]+)in([ ]+)%s([ ]+)$"%(p[0], p[1]))

    msg = """=====================================
Teclear  >>> exit    para salir del programa.

Dado el siguiente codigo:
    %s = 'Alphabet Aerobics'
    %s = "%s"

Escribe la expresion que evalua si %s contiene el valor almacenado en %s.
>>>"""%(p[0],p[1],p[2],p[1],p[0])

    #print(msg)

    answer = input(msg) #input(msg)
```

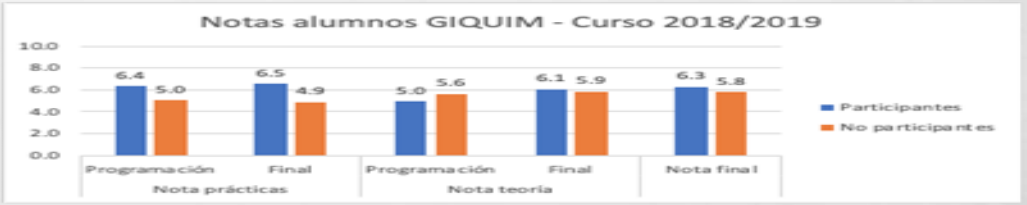
- There were three options: Python2 in English, Python 3 in English an Python3 in Spanish.
- All the members proposed exercises.
- Exercises were grouped according to the topic.
  - Some were based on eval or exec.
- The main part were based on lexical and syntactical parsing to avoid security problems.
- Still pending to implement with ***ast*** and ***sandbox***.

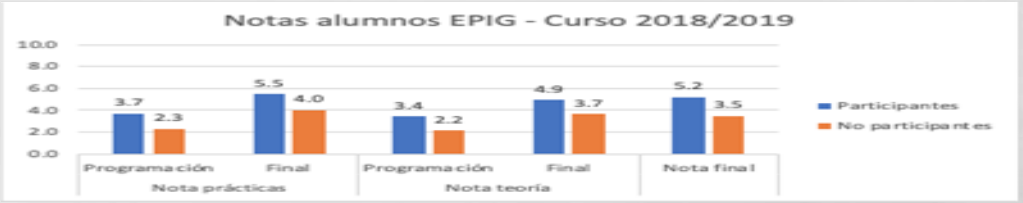
GYMKHANA: THE UGLY



- The students must solve the problems if they want get some reward.
- There played four roles, at least:
  - Those who worked hard
  - Those who worked really hard
  - Those who just worked but didn't stress
  - Those who didn't bother
- In the future, we plan to split the exercises in bunches based on the topics that are being explained in the lectures. Also, to consider the time spent to solve each exercise.







FINEXT 2019

A WORKSHOP DEVOTED TO IDENTIFY SUCCESSFUL STRATEGIES

FINEXT 2019

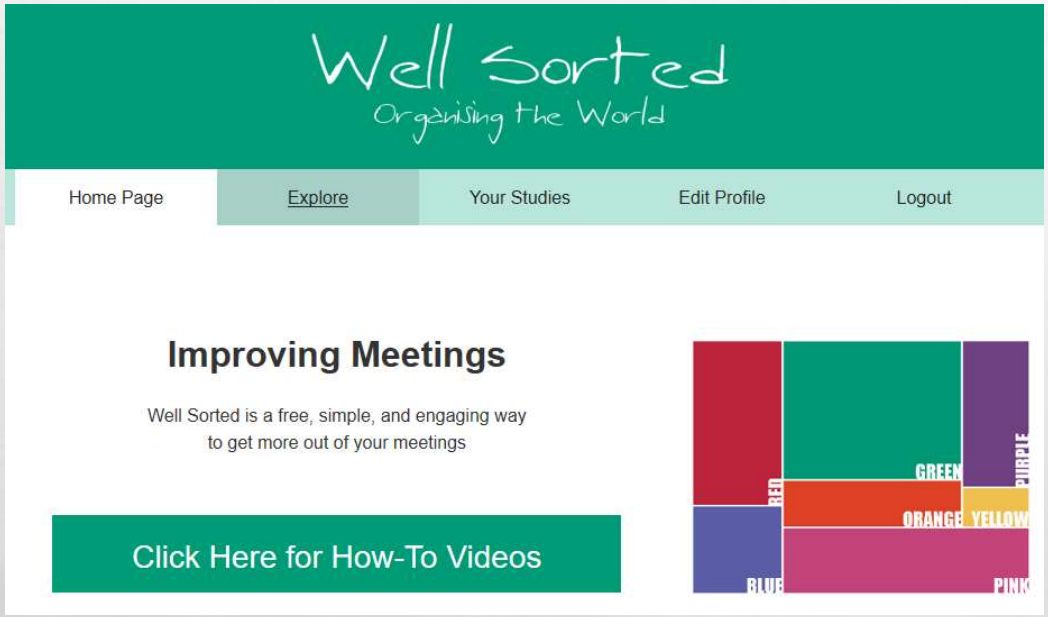
GOALS

- The aim of this experience is to share the problems perceived by the staff members when teaching the subject and to propose imaginative and feasible solutions.
- This process of ideas sharing might promote new collaborations in future innovative projects to enhance the teaching and improve the students' results.

STEPS

- A Well-Sorted enquiry to point out the main problems.
- A Well-Sorted grouping stage to sort out all the answers.
- A workshop to discuss the groups and propose ideas.

FINEXT 2019



Please, write down an idea (and a companion brief) about:

- Problems found in teaching the subject
- Concerns about teaching
- Ideas and proposals
- Any subject not included above

The second step involves grouping the answers

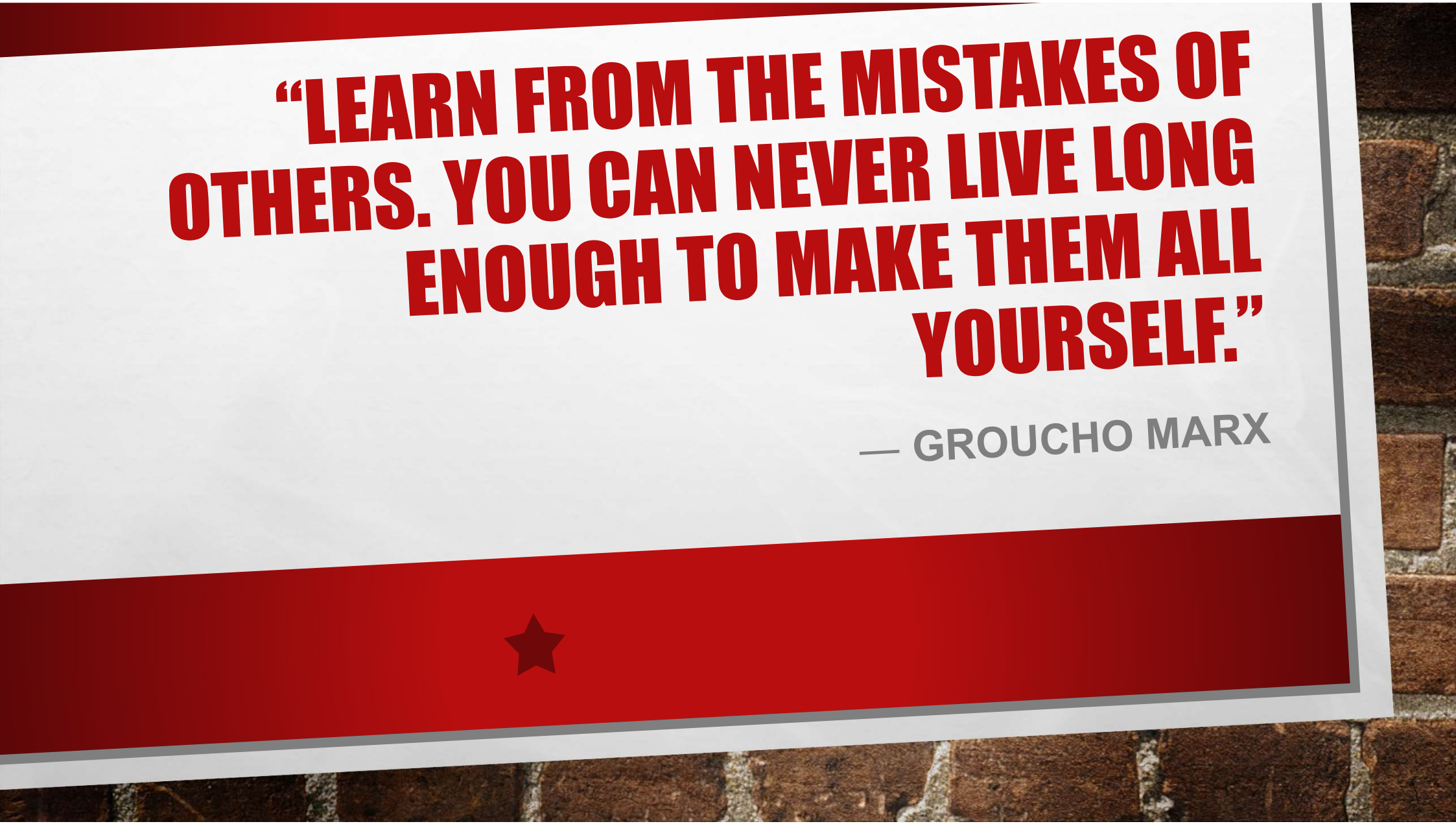
- Then, Well-sorted generates the results!

FINEXT 2019 Results

FINEXT 2019

CONCLUSIONS FROM THE WORKSHOP

- Propose exercises close to the degree, contact other staff members
- Students' feedback is worthy
- Using graphical programming languages might help when computing is the outlier
- But keep pencil and paper close to you!
- Merge the lectures and labs when possible
- FAQ's
- Rid of part of the contents
  - Choose the best tools, not the common ones
- Use good on-line tools


“LEARN FROM THE MISTAKES OF OTHERS. YOU CAN NEVER LIVE LONG ENOUGH TO MAKE THEM ALL YOURSELF.”
— GROUCHO MARX