UFAM

ICOMP
Instituto de Computação

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

(I) Brazilian-German Symposium on Learning Analytics
Mainz, May 12th 2023

LEARNING ANALYTICS IN INTRODUCTORY PROGRAMMING

Elaine Harada Teixeira de Oliveira

Research Group on Computing Education

Institute of Computing

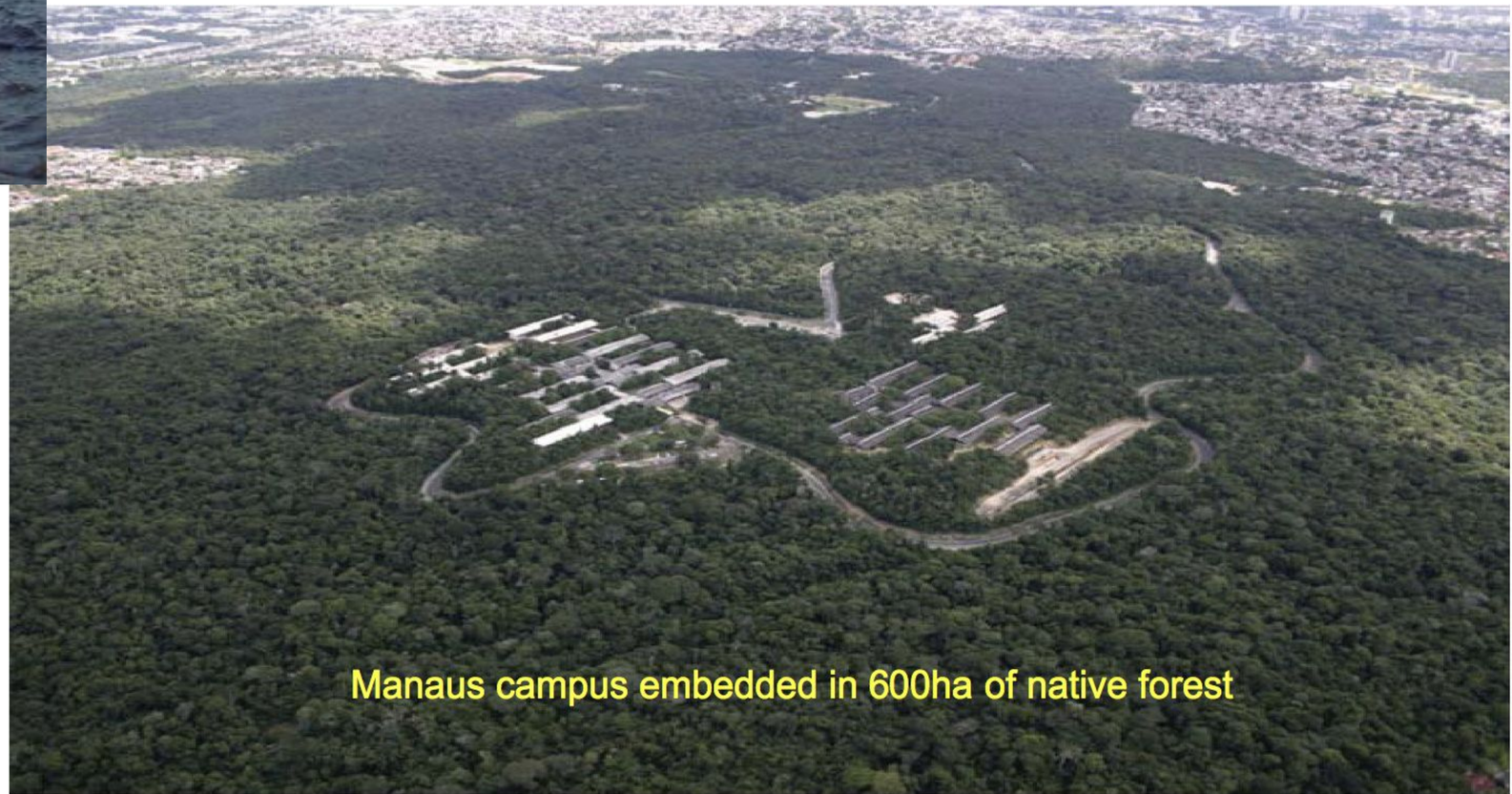
Federal University of Amazonas (UFAM)

Manaus, Amazonas, Brazil

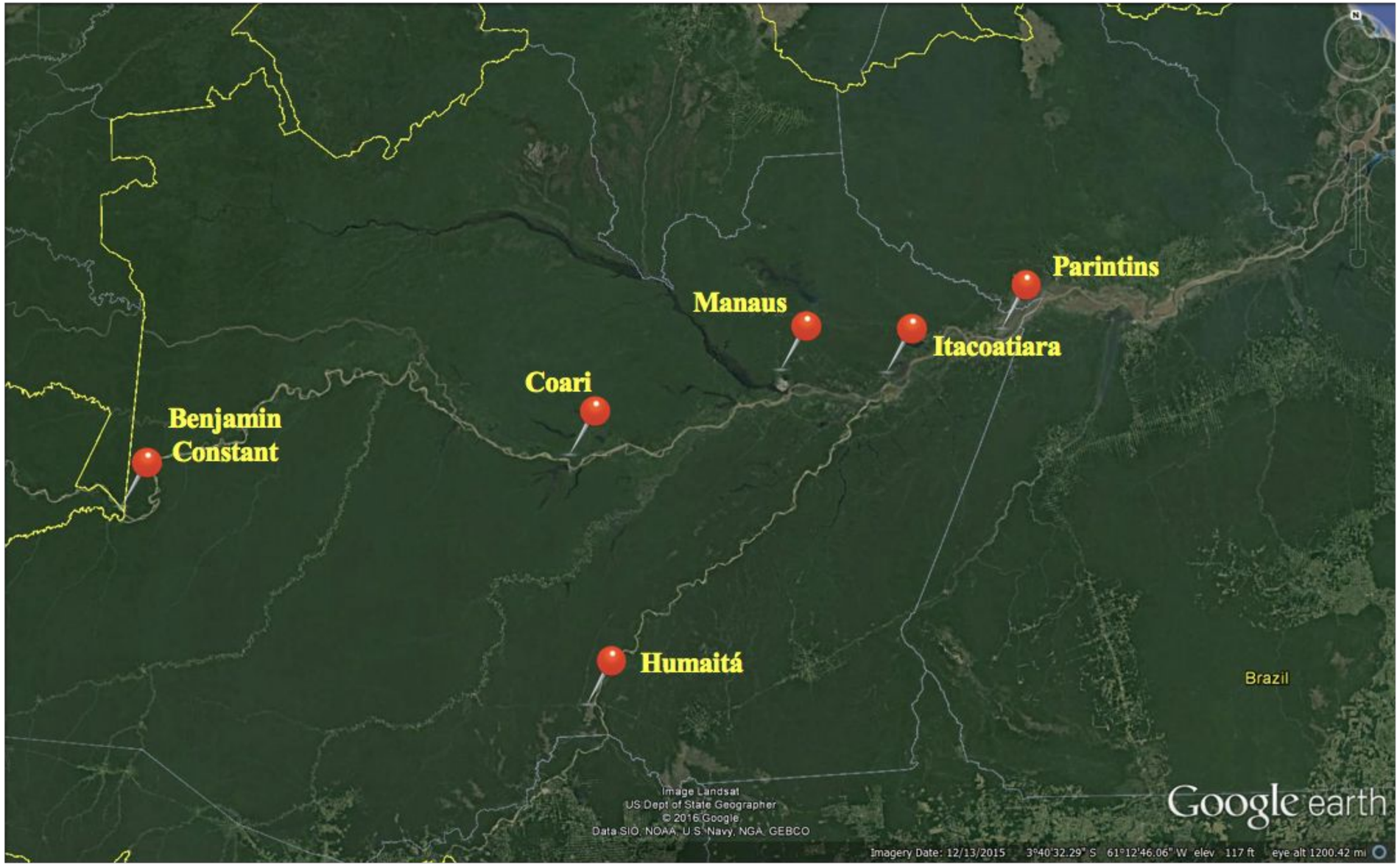
CAPES

Alexander von HUMBOLDT STIFTUNG

SUPER
PROJETO PARA EDUCAÇÃO E PESQUISA


Manaus campus embedded in 600ha of native forest

Amazonas state is our Campus

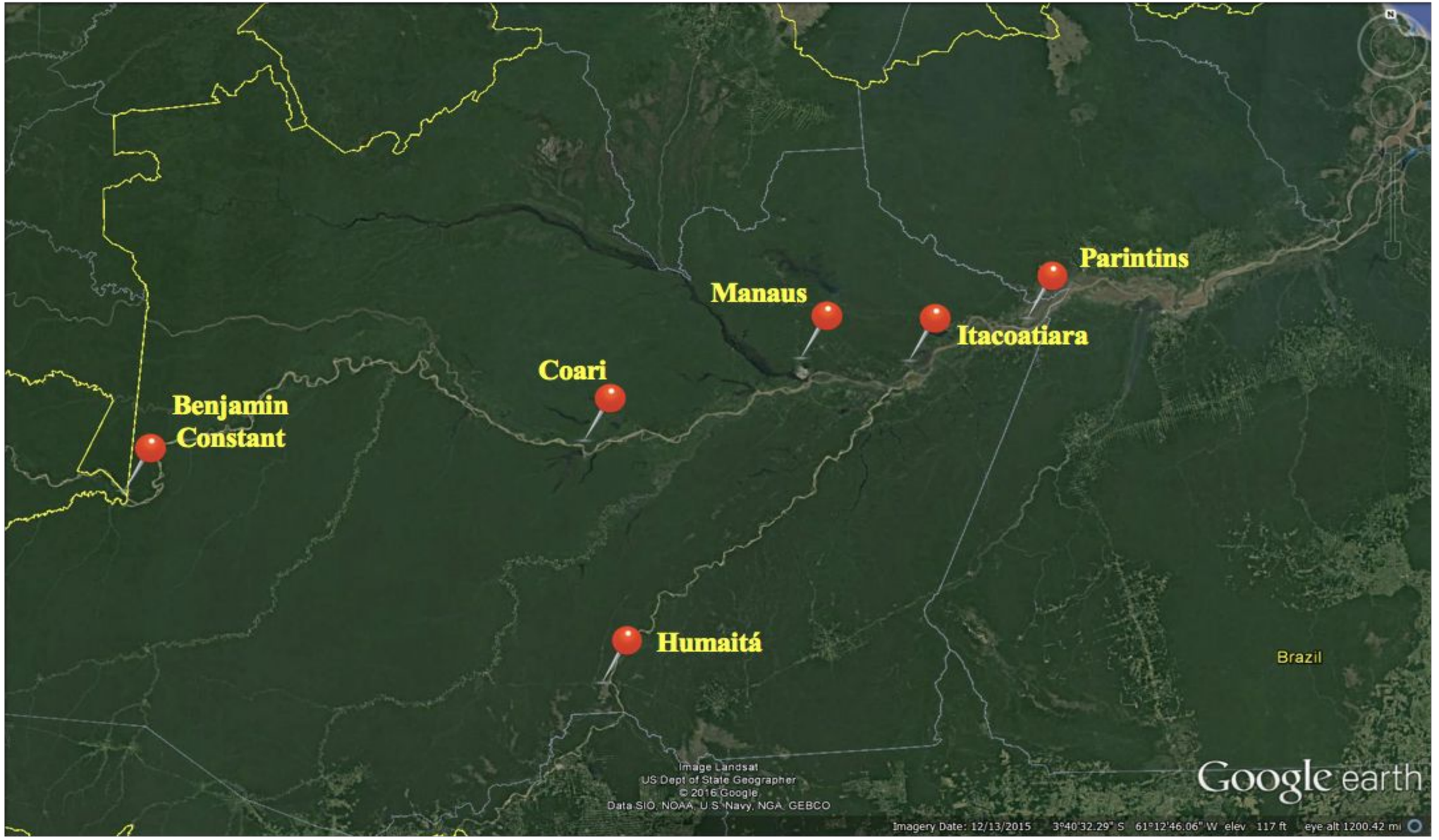


Acknowledgments to our former dean Prof. Hidembergue Ordozgoith da Frota

Amazonas state is our Campus

Acknowledgments to our former dean Prof. Hidembergue Ordozgoith da Frota

Amazonas state is our Campus



Acknowledgments to our former dean Prof. Hidembergue Ordozgoith da Frota

Amazonas state is our Campus

Acknowledgments to our former dean Prof. Hidembergue Ordozgoith da Frota


Manacapuru - Vila Sião
CEMEAM / SEDUC-AM


ESCOLAR

CS1 at UFAM

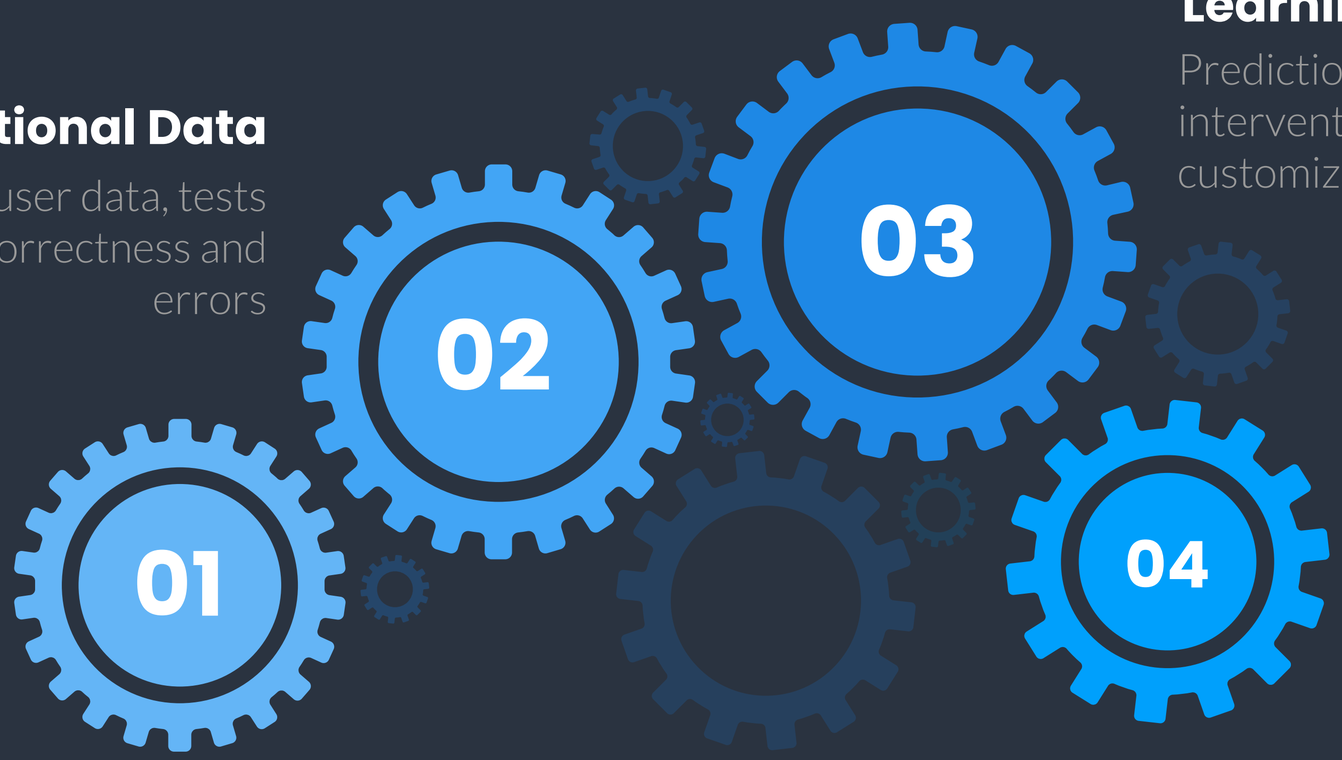
THE FOUR ENGINES

Educational Data

Interaction data, user data, tests and submissions, correctness and errors

Online Tools

Adoption of online tools for remote, hybrid and face-to-face teaching



Learning Analytics

Predictions, descriptions, interventions, recommendations, customization, etc.

Gamification

To encourage and engage

LEARNING ANALYTICS

THE FOUR ENGINES

In computer courses, **online judges are essential tools to support** teaching and learning activites

Online Tools

Adoption of online tools for remote, hybrid and face-to-face teaching

nal Data

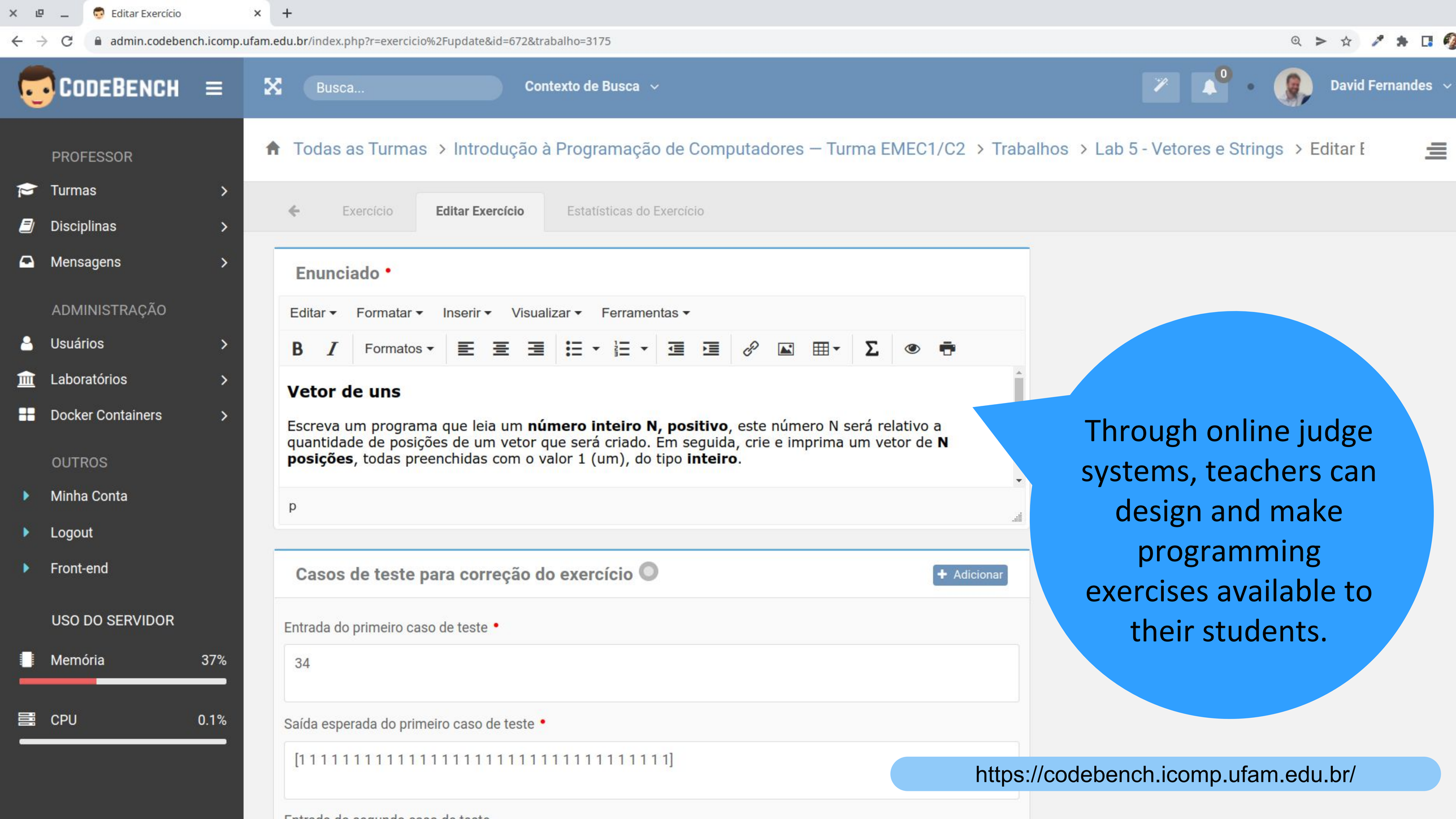
Interaction data, user data, tests and submissions, correctness and errors

Learning Analytics

Predictions, descriptions, interventions, recommendations, customization, etc.

Gamification

To encourage and engage


Editar Exercício
admin.codebench.icomp.ufam.edu.br/index.php?r=exercicio%2Fupdate&id=672&trabalho=3175
CODEBENCH
Busca...
Contexto de Busca
David Fernandes
PROFESSOR
Turmas
Disciplinas
Mensagens
ADMINISTRAÇÃO
Usuários
Laboratórios
Docker Containers
OUTROS
Minha Conta
Logout
Front-end
USO DO SERVIDOR
Memória 37%
CPU 0.1%
Todas as Turmas > Introdução à Programação de Computadores — Turma EMEC1/C2 > Trabalhos > Lab 5 - Vetores e Strings > Editar E
Exercício
Editar Exercício
Estatísticas do Exercício
Enunciado •
Editar Formatar Inserir Visualizar Ferramentas
Formatos
Vetor de uns
Escreva um programa que leia um número inteiro N, positivo, este número N será relativo a quantidade de posições de um vetor que será criado. Em seguida, crie e imprima um vetor de N posições, todas preenchidas com o valor 1 (um), do tipo inteiro.
p
Casos de teste para correção do exercício
Adicionar
Entrada do primeiro caso de teste •
34
Saída esperada do primeiro caso de teste •
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Through online judge systems, teachers can design and make programming exercises available to their students.
https://codebench.icomp.ufam.edu.br/

CODEBENCH

Introduction to Computer Programming

Programming Exercises

Home › ES01 › Homeworks

Start | Homeworks 2 | Materials | Messages

Start
- 01
- 02
- 03
- 04
- 05

Grades

Description

Write a program that checks whether a string entered by the user is palindrome or not.

Tips

A palindrome is a word, phrase, or other sequence of characters that reads the same backward as forward, such as level or radar.

Input/Output Example

Input	kayak
Output	True

File Edit Search Run Tools

Python 3 | main.py

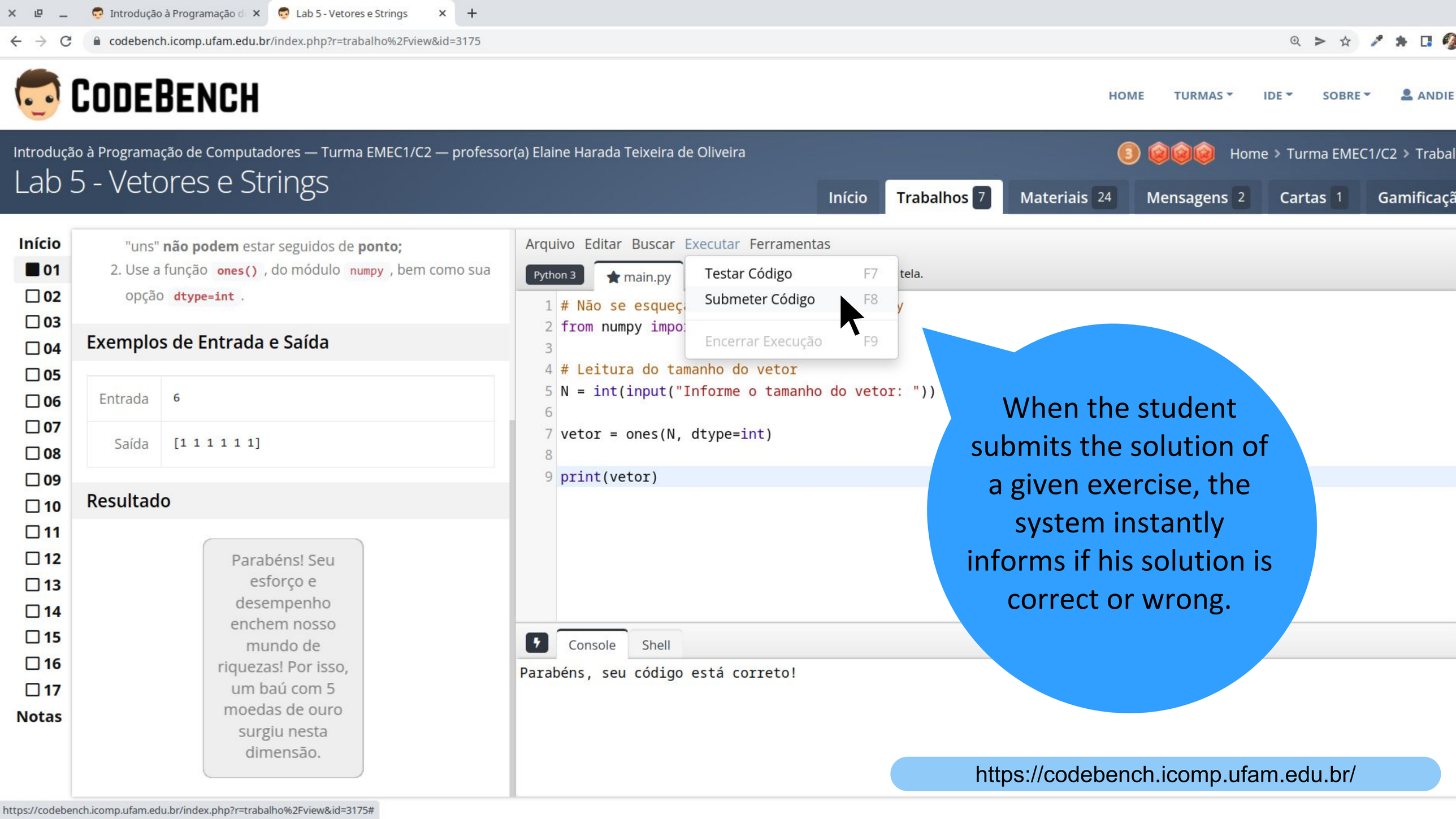
```
1  def is_palindrome(string):
2      left_pos = 0
3      right_pos = len(string) - 1
4
5      while right_pos >= left_pos:
6          if not string[left_pos] == string[right_pos]:
7              return False
8          left_pos += 1
9          right_pos -= 1
10     return True
11
12 str = input("Please enter a string: ")
13 print(is_palindrome(str))
```

Console | Shell

```
$ python3 main.py
Please enter a string: madam
True
```

Students code solutions for the proposed exercises in our system

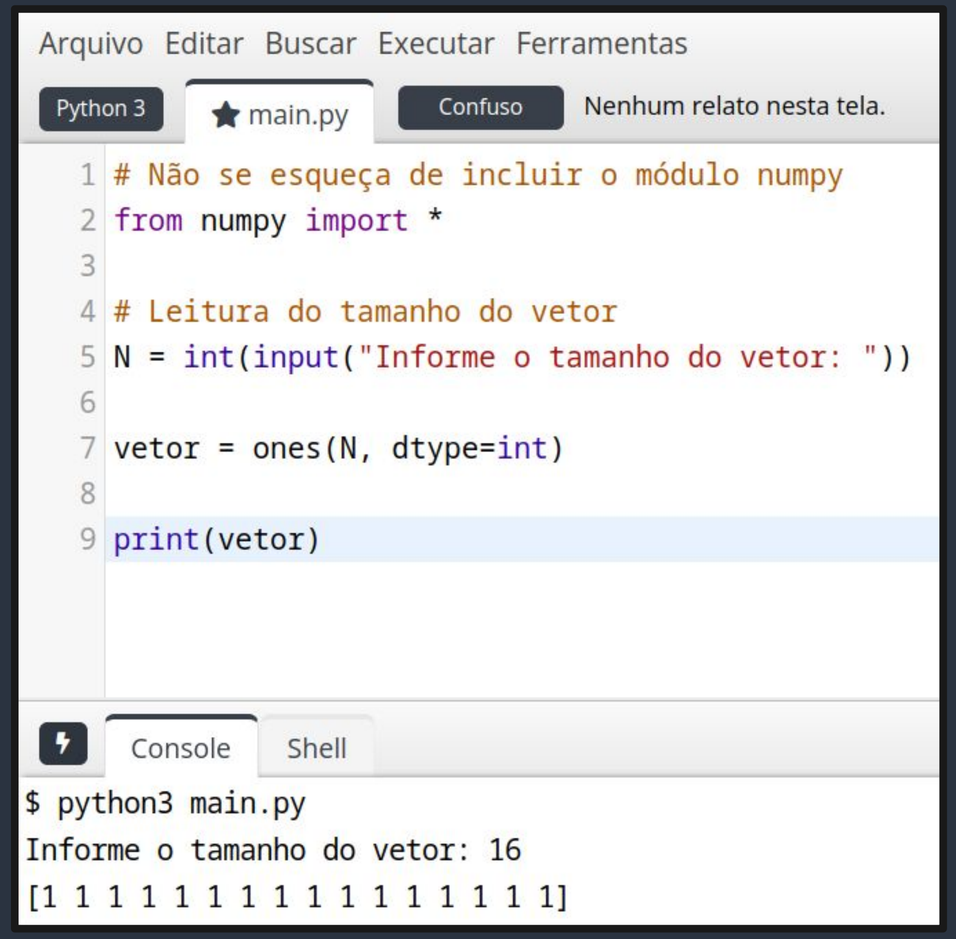
https://codebench.icomp.ufam.edu.br/


Introdução à Programação d
Lab 5 - Vetores e Strings
codebench.icomp.ufam.edu.br/index.php?r=trabalho%2Fview&id=3175
CODEBENCH
HOME
TURMAS
IDE
SOBRE
ANDIE
Introdução à Programação de Computadores — Turma EMEC1/C2 — professor(a) Elaine Harada Teixeira de Oliveira
Home › Turma EMEC1/C2 › Trabal
Lab 5 - Vetores e Strings
Início
Trabalhos 7
Materiais 24
Mensagens 2
Cartas 1
Gamificaçã
Início
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
Notas
"uns" não podem estar seguidos de ponto;
2. Use a função ones() , do módulo numpy , bem como sua opção dtype=int .
Exemplos de Entrada e Saída
Entrada
6
Saída
[1 1 1 1 1 1]
Resultado
Parabéns! Seu esforço e desempenho enchem nosso mundo de riquezas! Por isso, um baú com 5 moedas de ouro surgiu nesta dimensão.
Arquivo Editar Buscar Executar Ferramentas
Python 3
main.py
Testar Código F7
Submeter Código F8
Encerrar Execução F9
tela.
1 # Não se esqueç
2 from numpy impo
3
4 # Leitura do tamanho do vetor
5 N = int(input("Informe o tamanho do vetor: "))
6
7 vetor = ones(N, dtype=int)
8
9 print(vetor)
Console
Shell
Parabéns, seu código está correto!
When the student submits the solution of a given exercise, the system instantly informs if his solution is correct or wrong.
https://codebench.icomp.ufam.edu.br/
https://codebench.icomp.ufam.edu.br/index.php?r=trabalho%2Fview&id=3175#

DATA COLLECTION IN ONLINE JUDGES

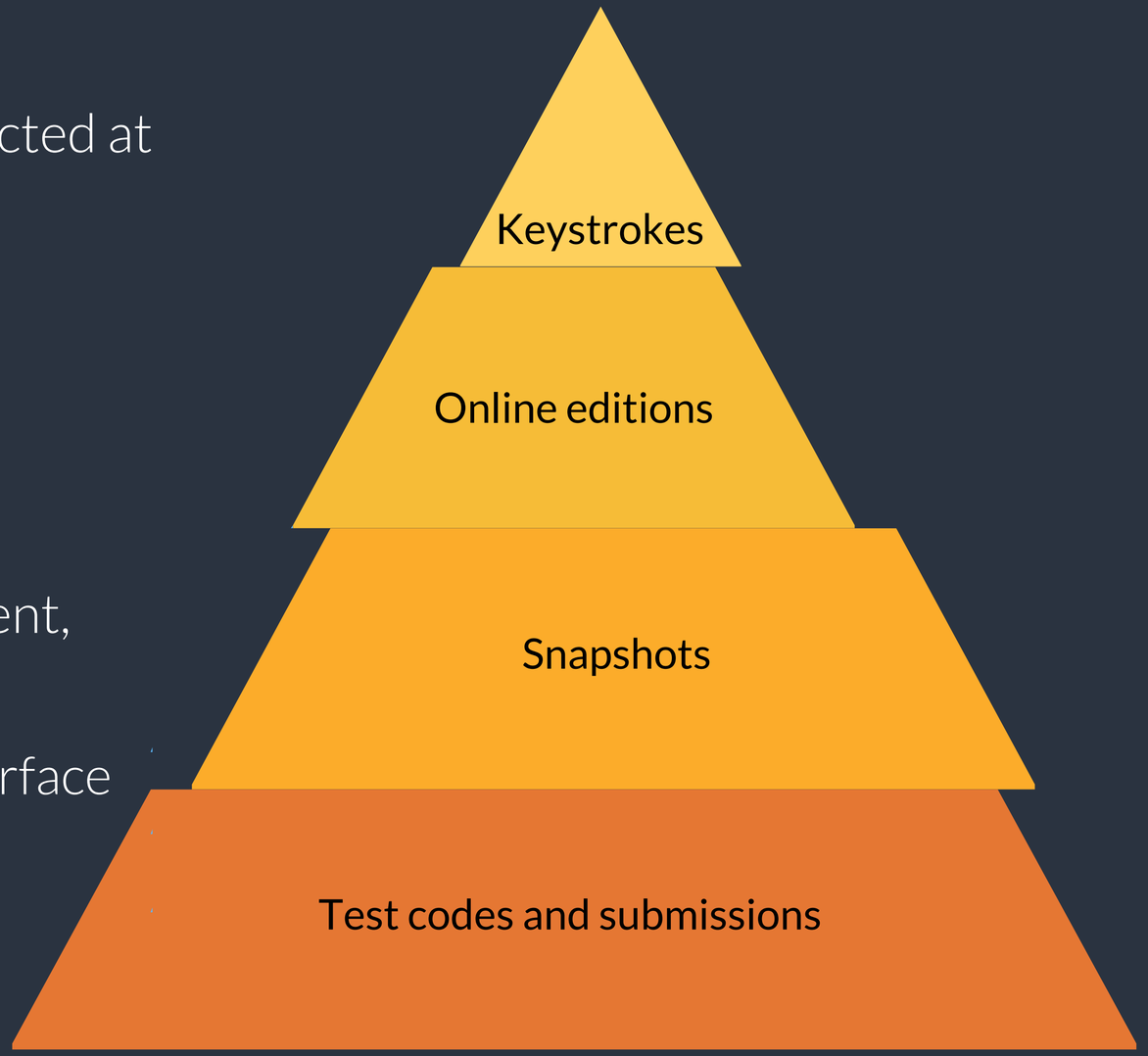
- Data generated by online judges differs from data generated by other types of web applications
  - Tests and submissions
  - Compiler/interpreter error messages
  - Code development process

- **Research opportunities on the use of Learning Analytics in online judge systems are wide and diverse**



DATA GRANULARITY

- The code development process can be collected at different levels of granularity
- Each level is associated with a data size and collection frequency
- In **CodeBench** , this collection is done at keystroke granularity
- In addition to the activities in the environment, the following are collected:
  - Mouse movements and clicks in the interface
  - Learning resources accessed
  - Logins and logouts
  - Activities on the gamification platform



CodeBench Dataset

codebench.icomp.ufam.edu.br/dataset/

banco imagens

CODEBENCH
EDUCATIONAL MINING DATASET

https://codebench.icomp.ufam.edu.br/dataset

CodeBench (http://codebench.icomp.ufam.edu.br) is a Programming Online Judge developed by the Institute of Computing (IComp) of the Federal University of Amazonas, Brazil. Through Codebench, teachers can provide lists of programming exercises to their students, who in turn must develop solutions for each exercise through an embedded IDE. Once a student submits a source code for a given exercise, the system instantly notifies the student whether him/her solution is correct or not.

The CodeBench automatically logs all actions performed by students on embedded IDE during their attempts to solve the proposed exercises. This dataset contains all logs collected from CS1 students during 2016 to 2021.

Releases

CodeBench Dataset 1.6

- This release of the dataset contains all logs collected from CS1 students during the years 2016 to 2021. Each academic year is divided into two semesters.
- Obs: Because of the COVID-19 pandemic that started in 2020, the semesters 1 and 2 of such year occurred throughout 2021. In 2020, there was a single semes
  Emergency Remote Education), where only 3 classes of IPC were taught.
- Release date: 2022-10-21
- Download options:
  - A file per semester: 2016-1, 2016-2, 2017-1, 2017-2, 2018-1, 2018-2, 2019-1, 2019-2, 2020-ERE, 2020-1, 2020-2, 2021-1 and 2021-2.
- Release statistics:

	2016-1	2016-2	2017-1	2017-2	2018-1	2018-2	2019-1	2019-2	2020-ERE	2020-1	2020-2	2021-1	2021-2	Total
CS1 classes	9	5	10	5	9	5	9	7	3	7	6	7	8	**90**
Total number of students	471	172	463	177	465	180	489	297	492	228	200	224	219	**4077**
Homework exercises	681	447	1278	556	1550	893	1559	1116	420	1008	882	962	1180	**12532**
Exam exercises	124	110	163	103	182	107	176	138	8	12	8	154	195	**1480**

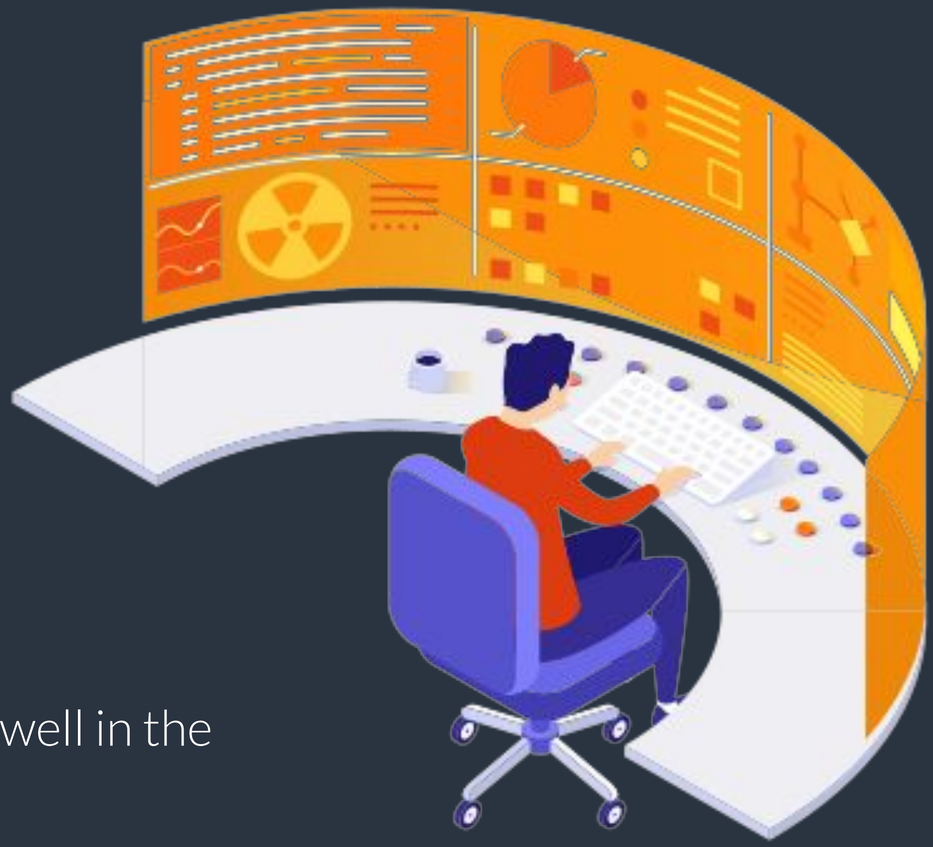
Since 2013, CodeBench has been used by about 6700 students and has more than 5600 exercises.

https://codebench.icomp.ufam.edu.br/dataset

```
2017-3-13 17:28:44.965#viewportChange#0
2017-3-13 17:28:46.733#change#{"from":{"line":2,"ch":0},"to":{"line":2,"ch":0},"text":["p"],"removed":[""],"origin":"+input"}
2017-3-13 17:28:46.885#change#{"from":{"line":2,"ch":1},"to":{"line":2,"ch":1},"text":["r"],"removed":[""],"origin":"+input"}
2017-3-13 17:28:47.109#change#{"from":{"line":2,"ch":2},"to":{"line":2,"ch":2},"text":["i"],"removed":[""],"origin":"+input"}
2017-3-13 17:28:47.373#change#{"from":{"line":2,"ch":3},"to":{"line":2,"ch":3},"text":["n"],"removed":[""],"origin":"+input"}
2017-3-13 17:28:47.605#change#{"from":{"line":2,"ch":4},"to":{"line":2,"ch":4},"text":["t"],"removed":[""],"origin":"+input"}
2017-3-13 17:28:50.369#change#{"from":{"line":2,"ch":5},"to":{"line":2,"ch":5},"text":["()"],"removed":[""],"origin":"+input"}
2017-3-13 17:28:50.369#keyHandled#"'('"
2017-3-13 17:28:52.083#change#{"from":{"line":2,"ch":6},"to":{"line":2,"ch":6},"text":["\"\""],"removed":[""],"origin":"+input"}
2017-3-13 17:28:52.083#keyHandled#"'\"'"
2017-3-13 17:30:31.907#viewportChange#0
2017-3-13 17:30:56.332#mousedown#{"isTrusted":true}
2017-3-13 17:30:56.349#focus#
```

DASHBOARDS IN ONLINE JUDGE SYSTEMS

- High rate of failures and dropouts in programming disciplines, notably for STEM courses, is a serious and recurring problem.
- One way to deal with this problem is to provide information and recommendations that help teachers to mitigate student difficulties.
  - Which students are having difficulty in the subject?
  - What are the best exercises for each student?
  - What aspects does the student need to improve to do well in the discipline?
  - Which students are 'cheating' on assignment lists?



DASHBOARDS IN ONLINE JUDGE SYSTEMS

- High rate of failures and dropouts in program disciplines, notably for STEM courses, is a se recurring problem.
- One way to deal with this problem is to provid information and recommendations that help te to mitigate student difficulties.
  - Which students are having difficulty in the subject?
  - What are the best exercises for each student?
  - What aspects does the student need to improve to do well in the discipline?
  - Which students are 'cheating' on assignment lists?

Would it be possible to answer all these questions through the **usage of logs** of online judges?

PROGRAMMING PROFILES

To answer the questions, we generate a **programming profile** for each student. from attributes extracted from the online judge's logs

Ratio between pasted characters and typed characters

Code changes between submissions

Number of events during development

Time between the start of coding and the deadline

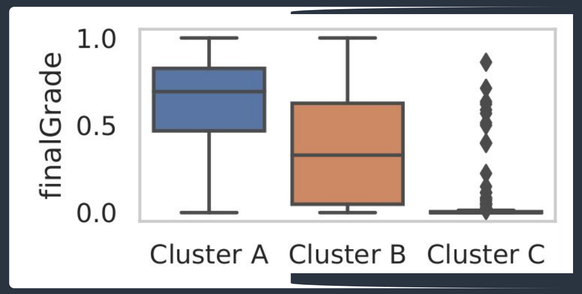
Average number of submissions per exercise

Proportion of submissions with errors

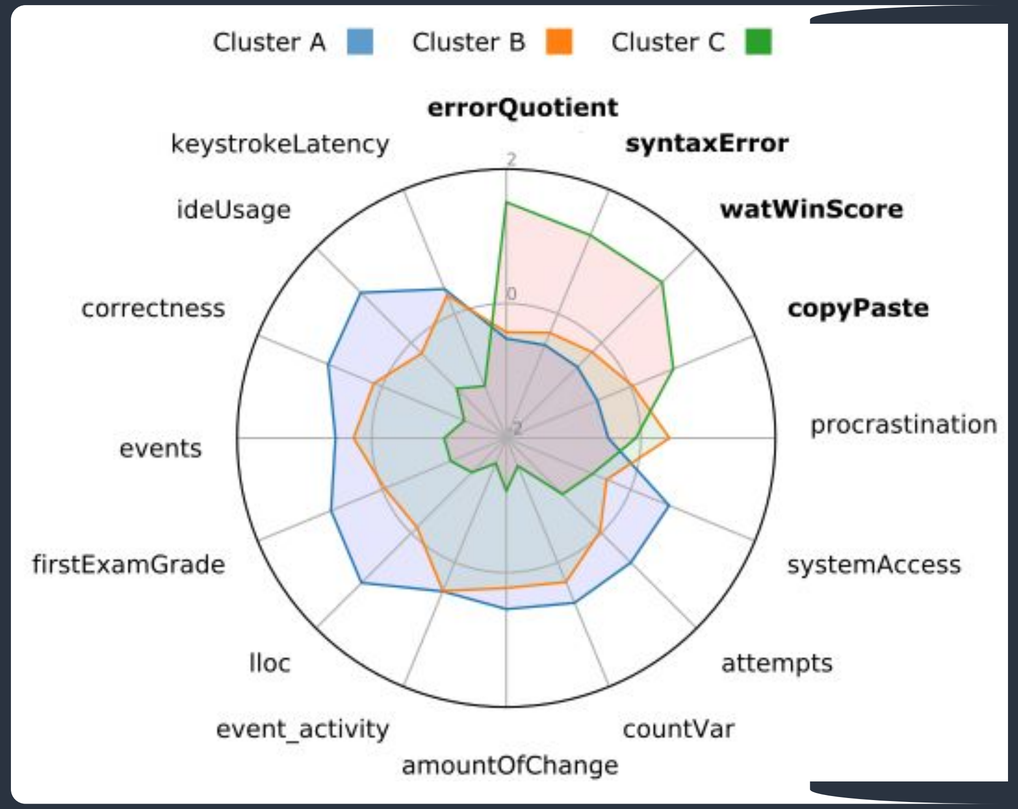
WHO ARE THE STUDENTS WITH DIFFICULTY?

EFFECTIVE AND INEFFECTIVE ATTITUDES

- Using clustering techniques (K-means) on programming profiles, students are grouped into 3 large groups:
  - Effective, average and non-effective students



- The graph on the side shows that the student's grade is directly related to the student's behavior in the online judge



Pereira, Filipe D.; OLIVEIRA, ELAINE HT ;OLIVEIRA, DAVID BF; CRISTEA, ALEXANDRA I. ; CARVALHO, LEANDRO SG ; FONSECA, SAMUEL C.; TODA, ARMANDO; ISOTANI, SEIJI. **Using learning analytics in the Amazon: understanding students? behavior in introductory programming** . British Journal of Educational Technology, vol. 51, p. 955-972, 2020.

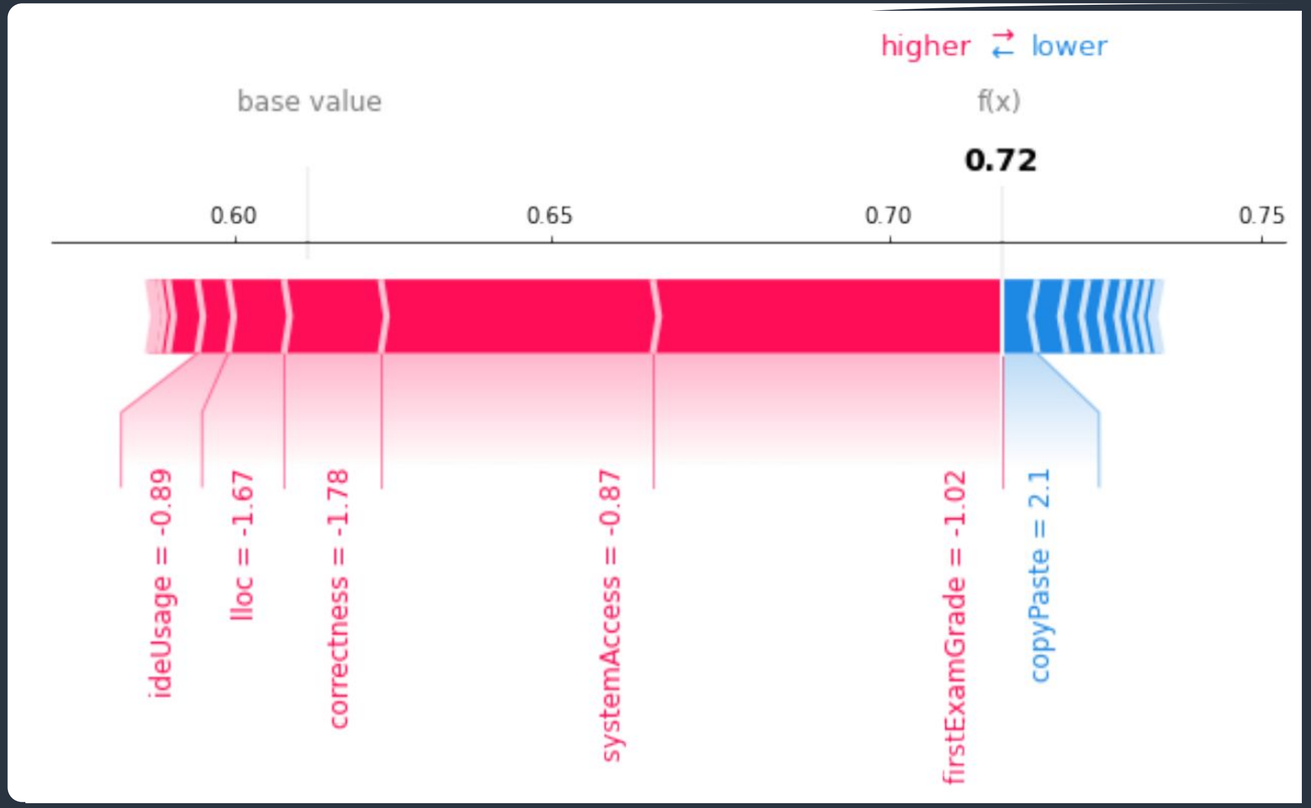
PERFORMANCE PREDICTION

- To deal with the high failure and dropout rates, it is essential to identify struggling students in advance.
- Using Deep Learning around programming profiles, it was possible to predict student performance with an accuracy of 82.5% in the first two weeks of class.
- With advance performance prediction, teachers can:
  - Provide extra assignments for students who do well in the class
  - Provide personalized support for struggling students

PEREIRA, FILIPE DWAN ; FONSECA, SAMUEL C. ; OLIVEIRA, ELAINE HT ; OLIVEIRA, DAVID BF ; CRISTEA, ALEXANDRA I. ; CARVALHO, LEANDRO SG. **Deep learning for early performance prediction of introductory programming students: a comparative and explanatory study** . BRAZILIAN JOURNAL OF COMPUTING IN EDUCATION, v. 28, p. 723-748, 2020.

EXPLAINING EFFECTIVE AND INEFFECTIVE BEHAVIORS

- In addition to predicting performance, prediction methods such as SHAP, based on games theory, are able to explain the results of their predictions.
- Early prediction empowered by its explanation may lead to better interventions.

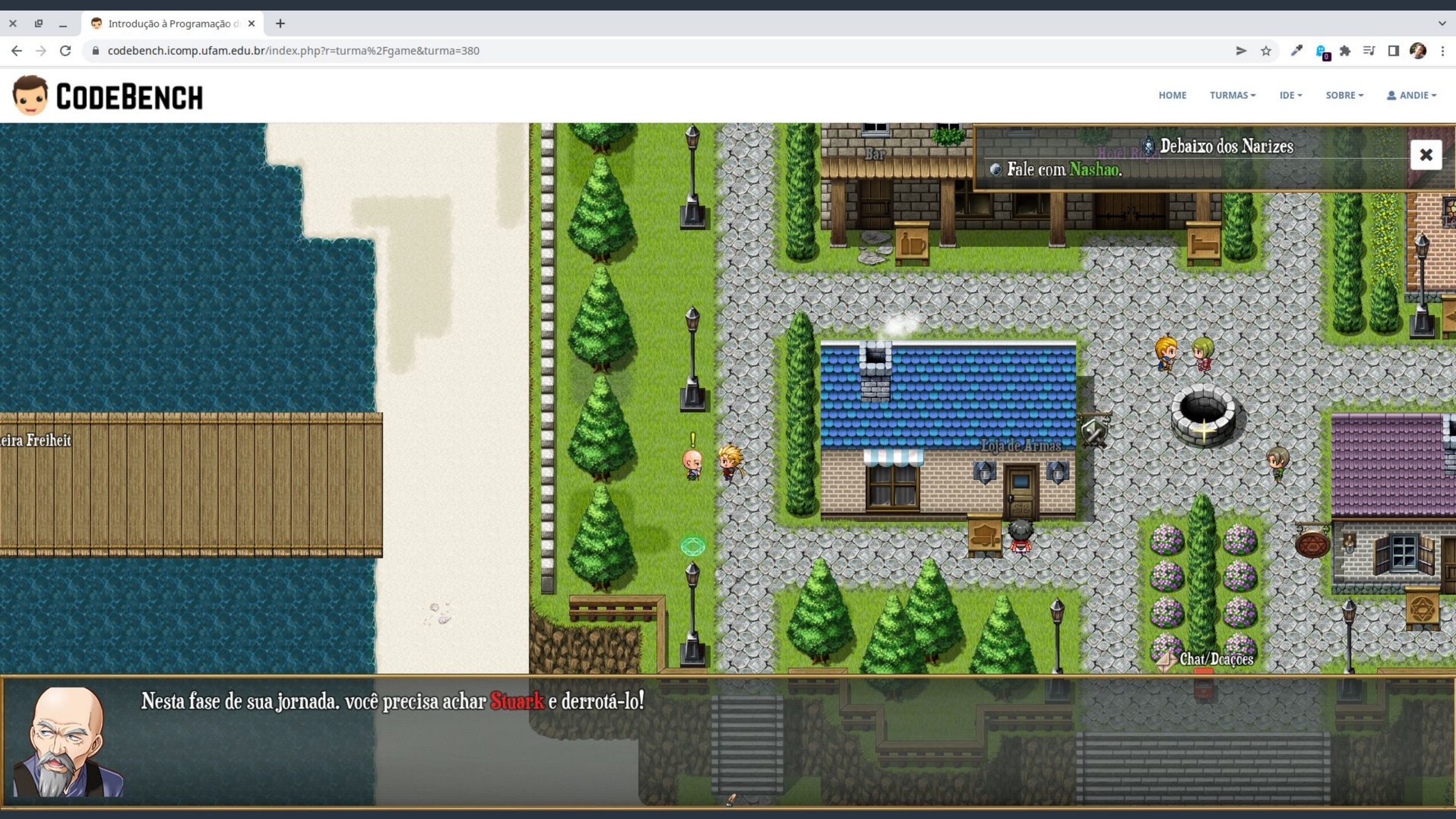


PEREIRA, FILIPE DWAN ; FONSECA, SAMUEL C. ; OLIVEIRA, ELAINE HT ; CRISTEA, ALEXANDRA I. ; BELLHAUSER, HENRIK ; RODRIGUES, LUIZ ; OLIVEIRA, DAVID BF ; ISOTANI, SEIJI ; CARVALHO, LEANDRO SG **Explaining Individual and Collective Programming Students? Behavior by Interpreting a Black-Box Predictive Model.** IEEE Access v. 9, p. 117097-117119, 2021.

STIMULATING EFFECTIVE BEHAVIORS

- We already know what behaviors lead to good performance in programming disciplines
- In this way, would it be possible to use customized gamification to encourage each student to behave in a more effective way?
- Gamification in CodeBench is based on RPG games, in which students need to learn programming to break free from a virtual world where they are trapped.

PESSOA, MARCELA ; MELO, RAFAELA ; HAYDAR, GABRIEL ; OLIVEIRA, DAVID BF DE ; CARVALHO, LEANDRO SG ; OLIVEIRA, ELAINE HT DE ; CONTE, TAYANA ; PEREIRA, FILIPE DWAN ; RODRIGUES, LUIZ ; ISOTANI, SEIJI. **An analysis of player types on a gamification platform incorporated into an online judge system** . In: Brazilian Symposium on Informatics in Education, 2021, Brazil.


Introdução à Programação
codebench.icomp.ufam.edu.br/index.php?r=turma%2Fgame&turma=380
CODEBENCH
HOME
TURMAS
IDE
SOBRE
ANDIE
Debaixo dos Narizes
Fale com Nashao.
Bar
Loja de Armas
Chat/Doações
eira Freiheit
Nesta fase de sua jornada. você precisa achar Stuark e derrotá-lo!

"Eu sou uma região de memória referenciada por um identificador... apesar de poder receber vários valores distintos. eu só posso assumir um único valor por vez! O que eu sou?"

Sob o Domínio de **Marduk**

Abrir Jogo

Enredos da gamificação: Geral | Capítulo 1 | Capítulo 2 | Capítulo 3 | Capítulo 4 | Capítulo 5 | Capítulo 6 | Final

Você e um grupo de amigos foram presos em um mundo virtual controlado por um ser extremamente poderoso chamado Marduk. Esse reino virtual, uma espécie de Matrix, é chamado de Midgard e foi desenvolvido através de uma tecnologia computacional muito sofisticada. A única forma de se libertar desse reino virtual é através de um portal mágico protegido por um monstro, a terrível Quimera, uma besta virtual muito impiedosa e poderosa. Como o mundo virtual foi desenvolvido através de tecnologia computacional e de linguagens de programação disponíveis no mundo real, a única forma de vencer a Quimera e se proteger dos demais perigos do mundo virtual é através do conhecimento de programação. Estudar programação faz você aumentar seu poder, possibilitando que você se transforme em um ser realmente poderoso, conhecido no mundo virtual como **CodeMeistre**.

Guerreiros e Guerreiras — Grupo Ouro 1

Maurício Viana Carvalho

Nível **18** Experiência **21129** x**23712**

Vida 640/640

Energia 108/108

Sandro Gomes Pinheiro

Nível **11** Experiência **5591** x**1948**

Vida 500/500

Energia 80/80

Pedro José Martins

Nível **10** Experiência **3985** x**1199**

Vida 480/480

Definições de termos

CARTAS DE RECOMPENSA (CARTAS DE THORIEL)

São as cartas sorteadas para um aluno quando um exercício é resolvido corretamente. As recompensas são: moedas, pontos de experiência, abertura de novos locais exploráveis e progressão nos capítulos. Também conhecidas como **cartas de Thoriel**.

EXPERIÊNCIA

Pontos de experiência (EXP) podem ser sorteados ao fim da resolução de exercícios, encontrados explorando os mapas, e realizando missões.

NÍVEIS

Indica o nível de evolução do aluno. Quanto maior o nível, mais poderoso ele se torna. O nível é definido pela

DETECTING PLAGIARISM USING LEARNING ANALYTICS

- In the initial programming classes, the codes developed by the students tend to be simple and small, making it difficult to use conventional plagiarism detection techniques.
- In this way, instead of using traditional techniques, we chose to analyze the students' behavior during the creation of codes in the online judge's IDE
- Starting from the premise that the behavior of those who are pasting is different from those who are actually solving the exercises, we arrived at the following results:

	Precisão	Revocação	Medida-F
Plágio	0.806	0.844	0.824
Não-plágio	0.846	0.809	0.827
Média	0.827	0.826	0.826

OLIVEIRA, DAVID BF DE; LAVAREDA FILHO, RONEM M. ; OLIVEIRA, ELAINE HT ; CARVALHO, LEANDRO SG ; PEREIRA, FILIPE DWAN ; COLONNA, JUAN G. ; MENEZES, ADRIA. A Plagiarism Detection Method for Online Judge Systems Based on Student Behavior. In: Brazilian Symposium on Informatics in Education, 2021, Brazil. P. 836.

Detecção de Plágio

admin.codebench.icomp.ufam.edu.br/index.php?r=trabalho%2Fplagio&id=2124

Busca... Contexto de Busca 0 David Fernandes

Informações Gerais | Editar Trabalho | Editar Exercícios do Trabalho | Analisar Casos de Teste | **Detecção de Plágio** | Estatísticas do Trabalho

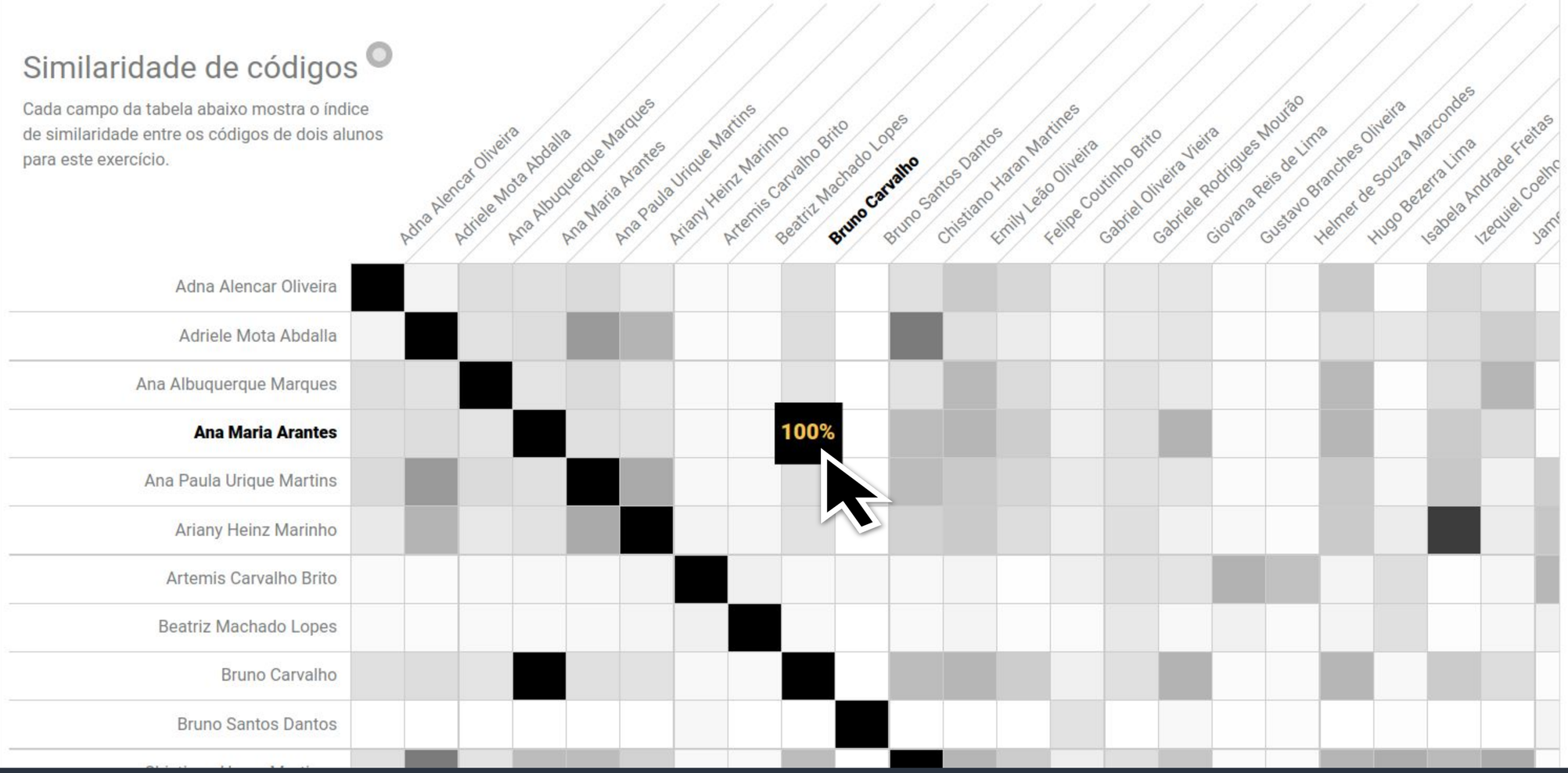
- Bloco de Exercícios 1
- Bloco de Exercícios 2
- Bloco de Exercícios 3
- Bloco de Exercícios 4
- Bloco de Exercícios 5
- Bloco de Exercícios 6
- Bloco de Exercícios 7
- Bloco de Exercícios 8
- Bloco de Exercícios 9
- Bloco de Exercícios 10
- Bloco de Exercícios 11
- Bloco de Exercícios 12

vetores | Moleza!

Detalhes do Exercício

Similaridade de códigos

Cada campo da tabela abaixo mostra o índice de similaridade entre os códigos de dois alunos para este exercício.



Bloco de Exercícios 1
Bloco de Exercícios 2
Bloco de Exercícios 3
Bloco de Exercícios 4
Bloco de Exercícios 5
Bloco de Exercícios 6
Bloco de Exercícios 7
Bloco de Exercícios 8
Bloco de Exercícios 9
Bloco de Exercícios 10
Bloco de Exercícios 11
Bloco de Exercícios 12

Detalhes do Exercício

Bloco de Exercícios 1 — Similaridade entre os códigos: **100%** Avaliar Plágio

Bruno Carvalho Opções

```
1  from numpy import*
2  v = array(eval(input("Qual o valor:")))
3  p = 200
4  i = 0
5  while(i<size(v)):
6      if(v[i]==1):
7          p = p*4
8      if(v[i]==2):
9          p = p*2
10     if(v[i]==4):
11         p = p/2
12     i = i + 1
13 print(round(p,2))
```

Ana Maria Arantes Opções

```
1  from numpy import*
2  v = array(eval(input("Qual o valor:")))
3  p = 200
4  i = 0
5  while(i<size(v)):
6      if(v[i]==1):
7          p = p*4
8      if(v[i]==2):
9          p = p*2
10     if(v[i]==4):
11         p = p/2
12     i = i + 1
13 print(round(p,2))
```

Gabriele Rodrigues Mourão | Giovana Reis de Lima | Gustavo Branches Oliveira | Helmer de Souza Marcondes | Hugo Bezerra Lima | Isabela Andrade Freitas | Izequiel Coelho

Artemis Carvalho Brito
Beatriz Machado Lopes
Bruno Carvalho
Bruno Santos Dantos

OTHER RESEARCH INVOLVING LEARNING ANALYTICS

- **Classification of difficulty of programming exercises**: by analyzing the behavior of students when proposed to an exercise, it is possible to infer the difficulty of it.
- **Detecting plagiarism**: by analyzing the behavior of students in the system, we're trying to identify the characteristics of who is 'cheating'.
- **Automatic detection of students feeling confused**: by analyzing the behavior of students in the system, it is possible to identify students who are feeling confused in a given exercise.
- **Group formation**: by analyzing the students' learning paths, we'll analyse the best collaborative techniques and group arrangements.

LEARNING ANALYTICS IN ONLINE JUDGE SYSTEMS

NEW CODEBENCH

- In 2023, version 3.0 of the CodeBench system will be launched, which is more secure, scalable and modern
- The new system will be released to institutions that want to conduct research involving online judges and learning analytics
- Some features of the new system:
  - Support for multiple languages
  - Support for multiple institutions
  - Support for creating study materials and interactive learning objects
  - Classroom (similar to Google Classroom)

01 02 03 04 05 06 07 08 09 10 11 12

ENUNCIADO

Escreva um programa que imprima o fatorial de um valor informado na entrada. Em matemática, o fatorial de um número natural n, representado por n!, é o produto de todos os inteiros positivos menores ou iguais a n.

DICAS

É importante não se esquecer dos casos bases, isto é, dos fatoriais de 0 e de 1.

CASOS DE TESTE DE EXEMPLO

Caso de Teste 1	EXECUTAR >_
Entrada	16
Saída	0 1 1 2 3 5 8 13 21 34

main.c | area.c | rectangle.c

```c
#include <stdio.h>
#include <math.h>
int main()
{
  int x1, y1, x2, y2, x, y, distance;

  // take first point's coordinates
  printf("Enter coordinates of first point: ");
  scanf("%d %d", &x1, &y1);

  // take second point's coordinates
  printf("Enter coordinates of second point: ");
  scanf("%d %d", &x2, &y2);

  x = (x2 - x1);
  y = (y2 - y1);

  distance = sqrt(x * x + y * y);

  // display result
  printf("Distance = %d", distance);

  return 0;
}
```

Francis

michael

CONSOLE | SHELL

Console

00:33:05

CodeMeistre

localhost:8081/ide

Universidade Federal do Amazonas

PT-BR

Lonie I
Student

01 02 03 04 05 06 07 08 09 10 11 12

ENUNCIADO

Escreva um programa que imprima o fatorial de um valor informado na entrada. Em matemática, o fatorial de um número natural n, representado por n!, é o produto de todos os inteiros positivos menores ou iguais a n.

DICAS

É importante não se esquecer dos casos bases, isto é, dos fatoriais de 0 e de 1.

CASOS DE TESTE DE EXEMPLO

Caso de Teste 1	EXECUTAR >_
Entrada	16
Saída	0 1 1 2 3 5 8 13 21 34

main.c | area.c | rectangle.c

```c
#include <stdio.h>
#include <math.h>
int main()
{
  int x1, y1, x2, y2, x, y, distance;

  // take first point's coordinates
  printf("Enter coordinates of first point: ");
  scanf("%d %d", &x1, &y1);

  // take second point's coordinates
  printf("Enter coordinates of second point: ");
  scanf("%d %d", &x2, &y2);

  x = (x2 - x1);
  y = (y2 - y1);

  distance = sqrt(x * x + y * y);

  // display result
  printf("Distance = %d", distance);

  return 0;
}
```

Francis

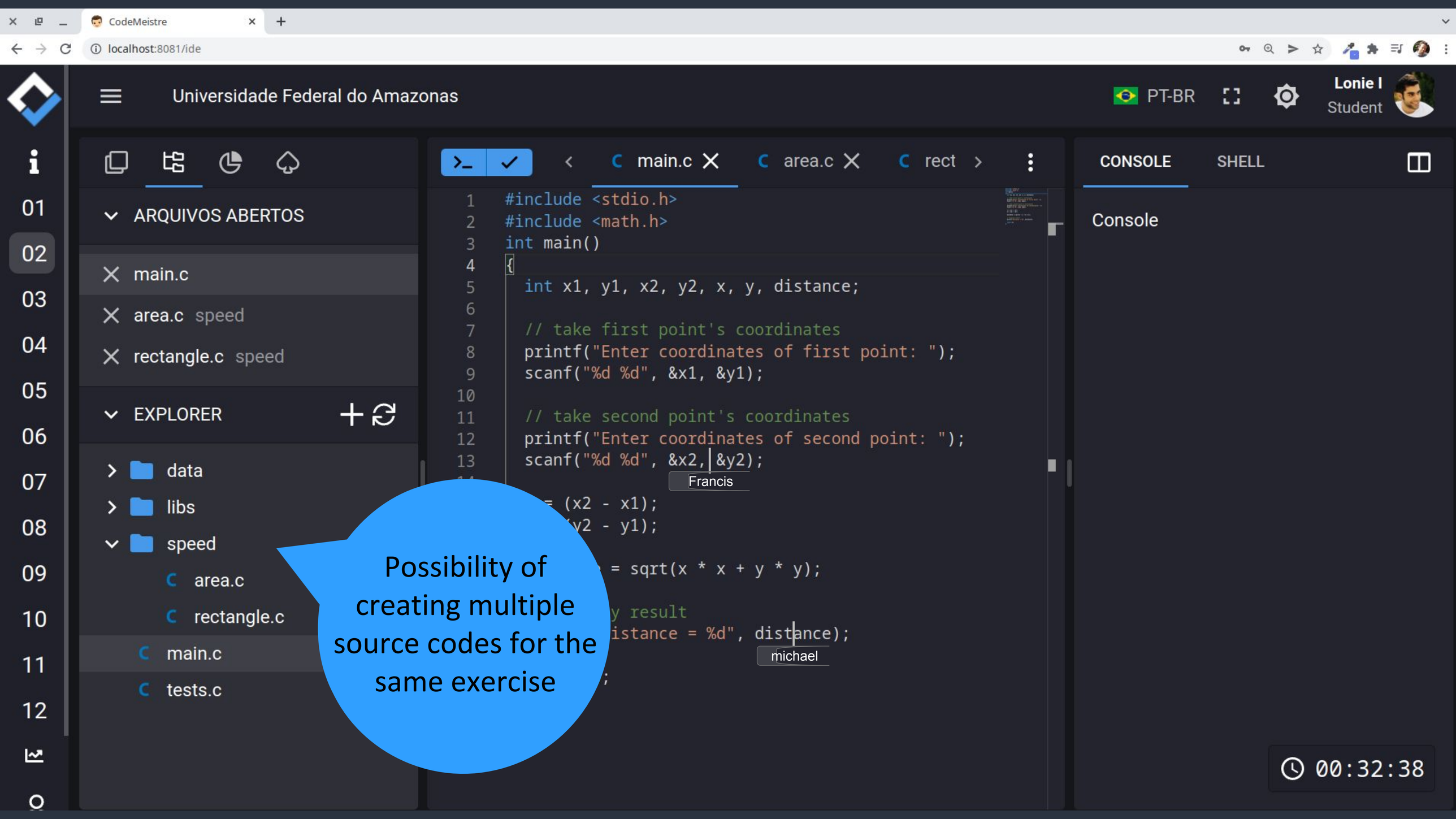
michael

CONSOLE | SHELL

Console

Possibility of creating group work, where students will develop the code collaboratively

00:33:05


CodeMeistre
localhost:8081/ide
Universidade Federal do Amazonas
PT-BR
Lonie I
Student
ARQUIVOS ABERTOS
main.c
area.c speed
rectangle.c speed
EXPLORER
data
libs
speed
area.c
rectangle.c
main.c
tests.c
main.c
area.c
rect
CONSOLE
SHELL
Console
#include <stdio.h>
#include <math.h>
int main()
{
int x1, y1, x2, y2, x, y, distance;
// take first point's coordinates
printf("Enter coordinates of first point: ");
scanf("%d %d", &x1, &y1);
// take second point's coordinates
printf("Enter coordinates of second point: ");
scanf("%d %d", &x2, &y2);
= (x2 - x1);
y2 - y1);
= sqrt(x * x + y * y);
y result
istance = %d", distance);
;
Francis
michael
Possibility of creating multiple source codes for the same exercise
00:32:38

UFAM

ICOMP
Instituto de Computação

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

(I) Brazilian-German Symposium on Learning Analytics
Mainz, May 12th 2023

Would you like to collaborate? :)

Elaine Harada Teixeira de Oliveira

Research Group on Computing Education

Institute of Computing

Federal University of Amazonas (UFAM)

Manaus, Amazonas, Brazil

CAPES

Alexander von HUMBOLDT STIFTUNG

SUPER
PROJETO PARA EDUCAÇÃO E PESQUISA