

Durante o período em que só tínhamos 2 módulos a decorrer, seguido de suspensão do curso de Julho a Setembro, inscrevi-me online na escola de programação 42 Lisboa <https://www.42lisboa.com/en/>

O que é a 42?

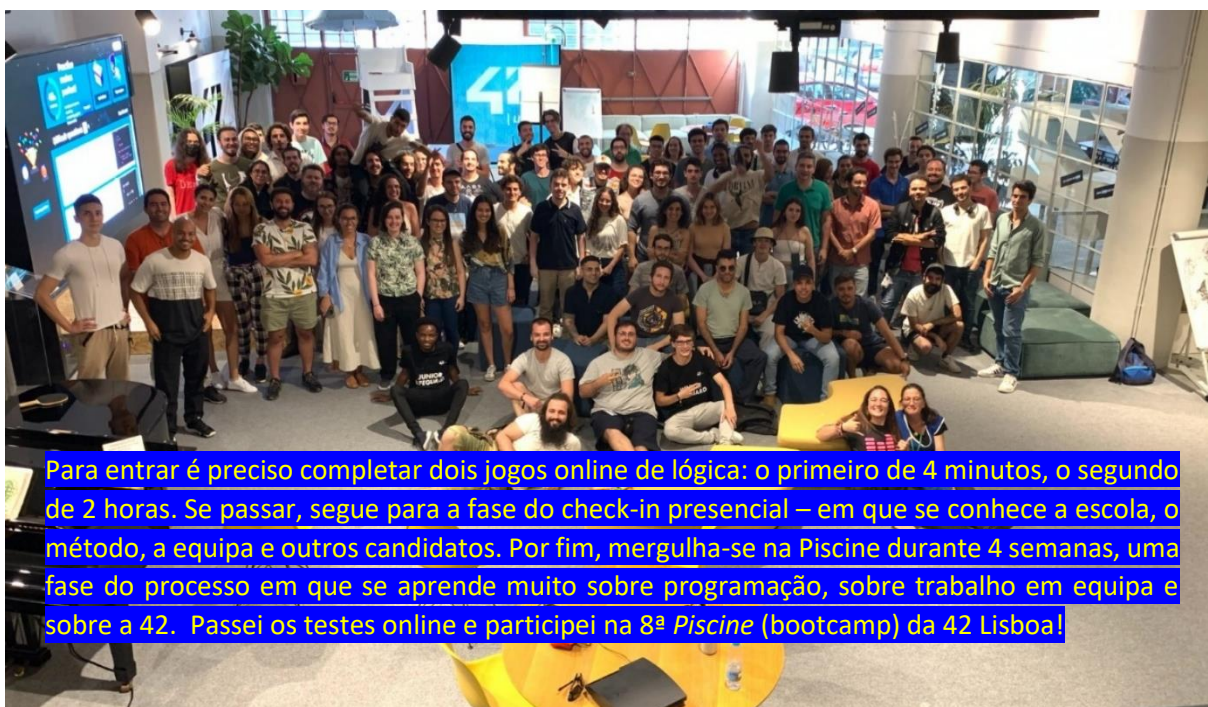
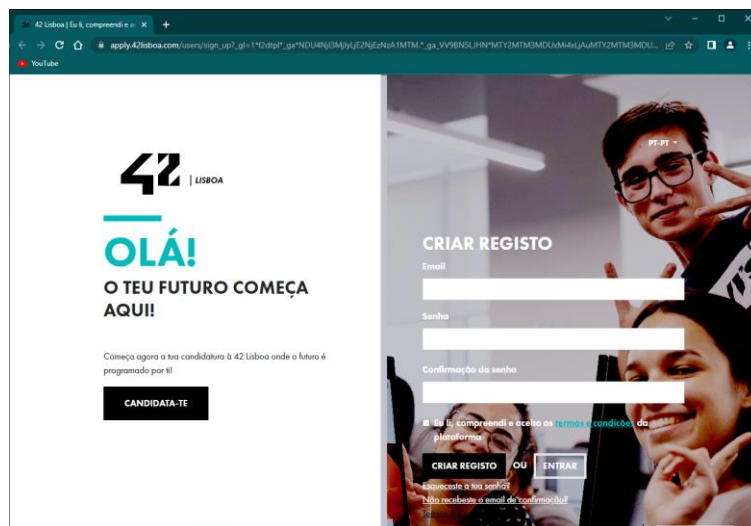
A 42 é uma escola de programação, reconhecida como uma das melhores do mundo, com um modelo pedagógico disruptivo, baseado num sistema de aprendizagem peer-to-peer, project based e gamificado.

Como é que pode não haver professores?

O programa da 42 é baseado na aprendizagem peer-to-peer, pelo que todos os alunos trabalham em conjunto, ensinam e aprendem.

Quem é que avalia os projetos?

Os projetos são avaliados por outros alunos com base em critérios de correção pré-definidos. O próprio deverá avaliar projetos dos colegas.



Para entrar é preciso completar dois jogos online de lógica: o primeiro de 4 minutos, o segundo de 2 horas. Se passar, segue para a fase do check-in presencial – em que se conhece a escola, o método, a equipa e outros candidatos. Por fim, mergulha-se na Piscine durante 4 semanas, uma fase do processo em que se aprende muito sobre programação, sobre trabalho em equipa e sobre a 42. Passei os testes online e participei na 8ª Piscine (bootcamp) da 42 Lisboa!

Foram 4 semanas mais intensas, duras, muito duras, difíceis e frustrantes, mas senti que aprendi bastante e cresci como pessoa, foi muito gratificante.

Desde a trabalhar com comandos Shell a criar códigos em C, tudo era feito no terminal (UNIX).

Programas em C não podíamos usar as funções das bibliotecas conhecidas, funções como o printf ou strcpy, não eram permitidas. Alguns exemplos de trabalhos:

```
drwxr-xr-x 5 alexa alexa 4096 Jul 31 18:03 venv
drwxr-xr-x 6 alexa alexa 4096 Jul 31 20:24 .vim
-rw-r--r-- 1 alexa alexa 10542 Aug 22 17:03 .viminfo
-rwxrwxrwx 1 alexa alexa 1912 Jul 31 20:23 .vimrc
-rw-r--r-- 1 alexa alexa 40142 Jul 31 17:50 .zcompdump
-rw-r--r-- 1 alexa alexa 50565 Aug 21 16:35 .zcompdump-LAPTOP-S619035R-5.8.1
-rw-r--r-- 1 alexa alexa 8615 Aug 24 20:54 .zsh_history
-rw-r--r-- 1 alexa alexa 3932 Jul 31 18:35 .zshrc
+ ~ vim teste.c
+ ~ Shared
+ Shared ls
a.out hello.c hello.c:Zone.Identifier Piscine
+ Shared Piscine
+ Piscine ls
C00 C02 C04 C06 C08 C10 C12 Shell00 stdheader.vim
C01 C03 C05 C07 C09 C11 C13 Shell01 stdheader.vim:Zone.Identifier
+ Piscine C00
+ C00 ls
c00.pdf ex01_ft_print_alphabet ex04_ft_is_negative ex07_ft_putnbr
c00.pdf:Zone.Identifier ex02_ft_print_reverse_alphabet ex05_ft_print_comb ex08_ft_print_combn
ex00_ft_putchar ex03_ft_print_numbers ex06_ft_print_comb2
+ C00 ex01_ft_print_alphabet
+ ex01_ft_print_alphabet ls
ft_print_alphabet.c ft_print_alphabet.c:Zone.Identifier
+ ex01_ft_print_alphabet vim ft_print_alphabet.c
+ ex01_ft_print_alphabet ..
+ C00 ..
+ Piscine ls
C00 C01 C02 C03 C04 C05 C06 C07 C08 C09 C10 C11 C12 C13 Shell00 Shell01 stdheader.vim stdheader.vim:Zone.Identifier
+ Piscine C05
+ C05 ls
c05.pdf ex00_ft_iterative_factorial ex02_ft_iterative_power ex04_ft_fibonacci ex06_ft_is_prime 'ex08_The Ten Queens'
c05.pdf:Zone.Identifier ex01_ft_recursive_factorial ex03_ft_recursive_power ex05_ft_sqrt ex07_ft_find_next_prime julhoAgostoIEFP.png
+ C05 ..
+ Piscine C04
+ C04 ls
c04.pdf c04.pdf:Zone.Identifier ex00_ft_strlen ex01_ft_putstr ex02_ft_putnbr ex03_ft_atoi ex04_ft_putnbr_base ex05_ft_atoi_base
```

Unix shell.

```
1 /* ***** */
2 /* */
3 /* */
4 /* ft_putchar.c */
5 /* */
6 /* By: apinto-a <apinto-a@student.42lisboa.com> */
7 /* */
8 /* Created: 2022/07/02 18:27:48 by apinto-a */
9 /* Updated: 2022/07/02 18:42:41 by apinto-a */
10 /* */
11 /* ***** */
12
13 #include <unistd.h>
14
15 void ft_putchar(char c)
16 {
17     write(1, &c, 1);
18 }
```

Função write (print).

```

12
13 #include <unistd.h>
14
15 void    ft_print_alphabet(void)
16 {
17     char    c;
18
19     c = 'a';
20     while (c <= 'z')
21     {
22         write (1, &c, 1);
23         c++;
24     }
25 }

```

Função para imprimir alfabeto.

```

1  /* ***** */
2  /*
3  /*
4  /*    ft_strlen.c
5  /*
6  /*    By: apinto-a <apinto-a@student.42lisboa.com>
7  /*
8  /*    Created: 2022/07/18 12:43:01 by apinto-a
9  /*    Updated: 2022/07/19 11:48:15 by apinto-a
10 /*
11 /* ***** */
12
13 int    ft_strlen(char *str)
14 {
15     int    count;
16
17     count = 0;
18     while (str[count] != '\0')
19     {
20         count++;
21     }
22     return (count);
23 }

```

Função para ler a string no seu comprimento total.

```

12
13 char    *ft_strcat(char *dest, char    *src)
14 {
15     int    i;
16     int    j;
17
18     i = 0;
19     j = 0;
20     while (dest[i] != '\0')
21         i++;
22     while (src[j] != '\0')
23     {
24         dest[i] = src[j];
25         i++;
26         j++;
27     }
28     dest[i] = '\0';
29     return (dest);
30 }
12
13 int    ft_atoi(char *str)
14 {
15     int    res;
16     int    sign;
17     int    count;
18
19     res = 0;
20     sign = 1;
21     count = 0;
22     while (str[count] == 32 || (str[count] >= 9 && str[count] <= 13))
23         count++;
24     while (str[count] == '-' || str[count] == '+')
25     {
26         if (str[count] == '-')
27             sign *= -1;
28         count++;
29     }
30     while (str[count] >= '0' && str[count] <= '9')
31     {
32         res = res * 10 + str[count] - 48;
33         count++;
34     }
35     return (res * sign);
36 }

```

Função atoi (ASCII to integer) - converte strings em números inteiros.