

Introdução à Programação de Computadores para Biologia

Apresentação da disciplina

Aula 01

<https://ttdorres.github.io/introprog2021/>

Apresentação da disciplina

Horário:

Quinta-feira – 19:00 às 21:00 horas

Professora: Tatiana Teixeira Torres, e-mail: tttorres@ib.usp.br

Página da disciplina: <https://tttorres.github.io/introprog2021/>

Apresentação da disciplina

Objetivos:

Apresentar conceitos básicos de desenvolvimento de algoritmos e sua representação em linguagem de alto nível. Capacitar o aluno no uso da linguagem Perl. Promover o desenvolvimento e implementação de programas e sua aplicação na resolução de problemas biológicos.

Programa:

Aula	Data	Tema
1	19/08	Apresentação da disciplina
2	26/08	Introdução à Programação / Bash
3	02/09	Bash (continuação) / Algoritmos
4	09/09	Introdução ao Perl / Variáveis e tipos de dados
5	16/09	Arrays e Hashes
6	23/09	Estruturas de controle: condicionais
7	30/09	Estruturas de controle: loops I
8	07/10	Estruturas de controle: loops II
9	14/10	Manipulação de arquivos I
10	21/10	Manipulação de arquivos II
×	28/10	Dia do servidor público. Não haverá aula
11	04/11	Manipulação de strings e expressões regulares
12	11/11	Subrotinas
13	18/11	Módulos I: introdução e CPAN
14	25/11	Módulos II: BioPerl
15	02/12	Revisão e encerramento da disciplina

Apresentação da disciplina

Forma de avaliação:

Exercícios semanais: a cada novo tema de aula, será disponibilizado um exercício que deverá ser entregue no prazo de sete dias. A entrega dos exercícios será realizada via moodle (<https://edisciplinas.usp.br/>). O atraso na entrega implica na redução da nota pela metade. Serão 10 exercícios ao longo do semestre, com pesos diferentes conforme a complexidade do exercício.

Apresentação da disciplina

Forma de avaliação:

```
#!/usr/bin/perl

my ($E, $M, $H, $nota_final); #$E, easy; $M, medium; $H, hard
($E, $M, $H) = @ARGV;

$nota_final = ((2*$E)+(3*$M)+(5*$H))/10;

if ($nota_final >= 5) {
    print "Aluno aprovado\n";
} else {
    print "Aluno reprovado\n";
}

#conversao de nota para conceitos, apenas para a Pos-graduacao
if ($nota_final < 5) {
    print "R, Reprovado, sem direito a credito\n";
} elsif ($nota_final <= 7.0) {
    print "C, Regular, com direito a credito\n";
} elsif ($nota_final <= 8.5) {
    print "B, Bom, com direito a credito\n";
} else {
    print "A, Excelente, com direito a credito\n";
}
exit;
```

Apresentação da disciplina

Material de estudo:

- Curtis Jamison, D. "Perl Programming for Biologists", John Wiley & Sons, Inc., 2003.
- Wall, L.; Christiansen, T.; Orwant, J. "Programming Perl", O'Reilly Media, 2000.
- Tisdall, J. "Beginning Perl for Bioinformatics", O'Reilly Media, 2001.
- Setubal J. C.; Meidanis J. "Introduction to Computational Molecular Biology", Brooks/Cole Pub Co, 1997.
- Tutoriais e livros - <http://www.perl.org/>; <http://learn.perl.org/>



Flexible & Powerful

That's why we love Perl 5

► [Get started](#)



DOWNLOAD PERL



Perl 5 is a highly capable, feature-rich programming language with over 24 years of development. [More about why we love Perl...](#)



[Learning Perl 5](#)

With free online books, over 24,900 extension modules, and a large developer community, there are many ways to learn Perl 5.



[The Perl Community](#)

Perl has an active world wide community with over 300 local groups, mailing lists and support/discussion websites.



[Documentation](#)

Core documentation, FAQs and translations.



[Contribute to Perl](#)

Perl is being [actively developed](#). There are many ways to [get involved](#).



[CPAN](#)

The Comprehensive Perl Archive Network (CPAN) has over 18,000 open source modules available for download.



[Events and Conferences](#)

Conferences, social and technical events around the world offer lots of networking and learning opportunities.



[Perl 6](#)

Perl 6 is a sister language, part of the Perl family. Perl 6 is not production ready yet, you can get involved with its [development](#) though.



[The Perl Foundation](#)

The Perl Foundation is dedicated to the advancement of the Perl programming language through open discussion, collaboration, design, and code.

Current Perl version

[5.16.0 - download now](#)

Find out more

[Learn](#)

[Documentation](#)

[Community](#)

[Events](#)

Tip

Database interfaces

[DBIx::Class](#) provides an Object Relational Mapper (ORM) to databases (e.g. Oracle, SQL Server, MySQL, Postgress, Access etc)

Sponsor



Local Search

Apresentação da disciplina

Requisitos computadores pessoais:

SISTEMA OPERACIONAL UNIX

MacOS (Intel), Linux

PERL

EDITOR DE TEXTO

Apresentação da disciplina

Requisitos computadores pessoais:

WINDOWS 10: duas opções

1. "Dual boot" Windows e Linux
2. Subsistema do Windows para o Linux

Apresentação da disciplina

Requisitos computadores pessoais:

PERL

perl -v

no mínimo 5.8.3

<http://www.perl.org/>

Faremos a instalação em uma das aulas

Apresentação da disciplina

Requisitos computadores pessoais:

EDITOR DE TEXTOS

Qualquer um, mas...

EDITOR DE TEXTOS

Editor	no.	%
Vim (or vi or gvim)	1097	34%
Emacs (or xemacs)	430	13%
Ultra Edit	224	7%
Eclipse EPIC	210	6%
Other answer...	143	4%
Notepad++	142	4%
Komodo IDE	128	4%
Komodo Edit	105	3%
TextMate	105	3%
Padre	101	3%
Kate	56	2%
Gedit	55	2%
TextPad	49	2%
nano	40	1%
SciTE	38	1%
Geany	36	1%

Apresentação da disciplina

EDITOR DE TEXTOS vs IDE

IDE, do inglês Integrated Development Environment ou Ambiente Integrado de Desenvolvimento

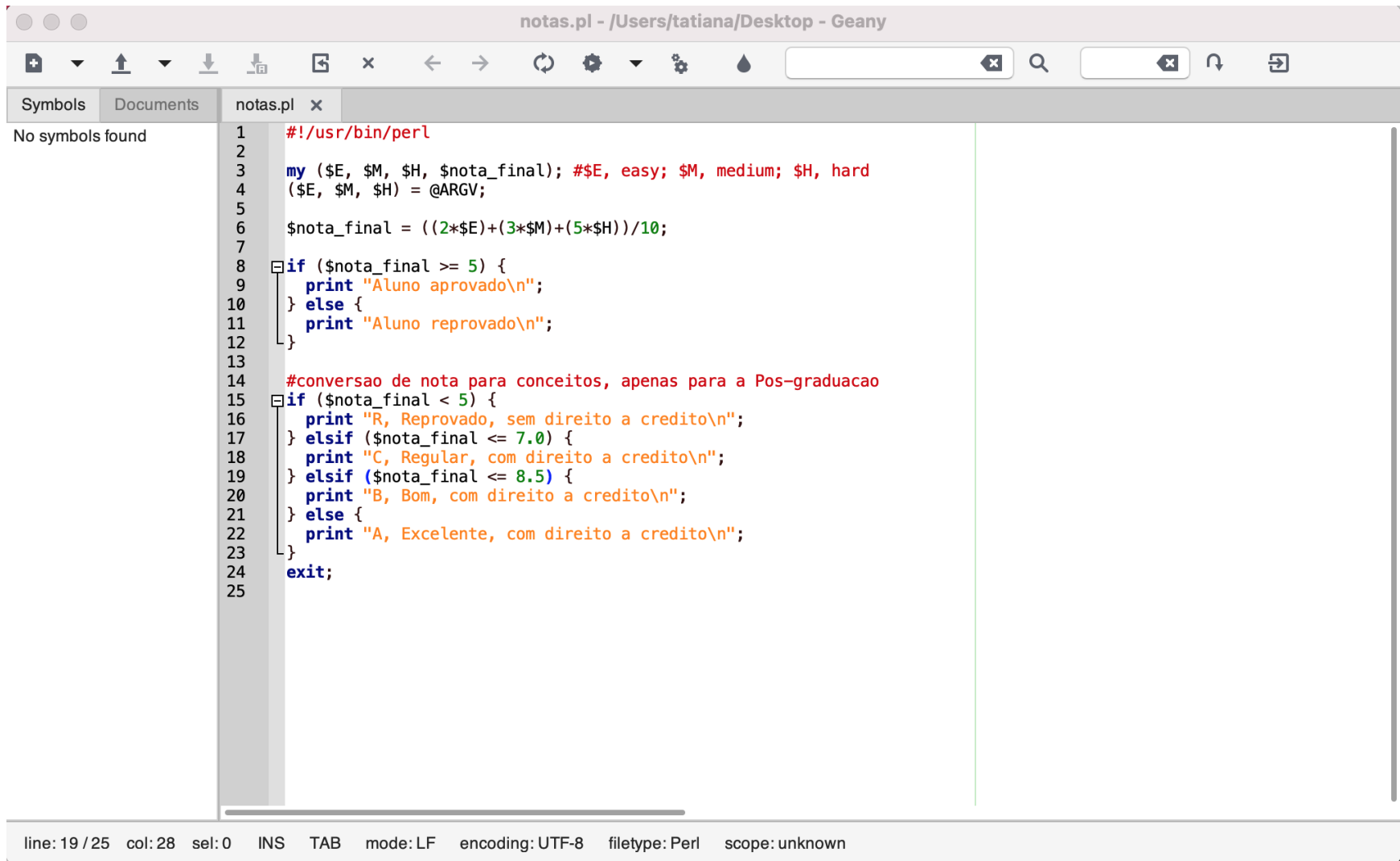
Editor de texto

Compilador (compiler)

Depurador (debugger)

Gerador de código

Geany

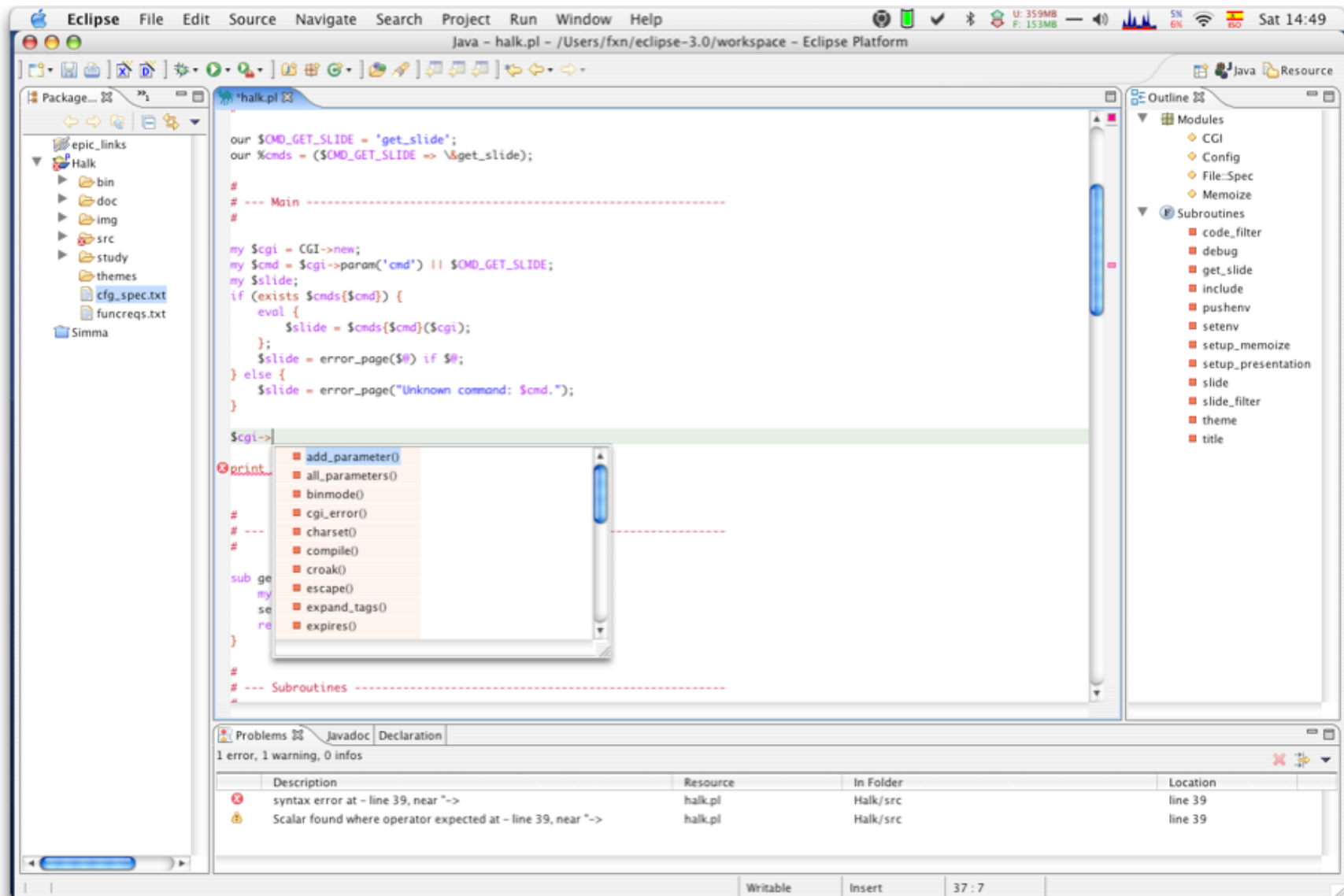


The screenshot shows the Geany IDE with a file named `notas.pl` open. The file is located at `/Users/tatiana/Desktop`. The code is a Perl script that calculates a final grade based on three inputs: `$E` (easy), `$M` (medium), and `$H` (hard). The calculation is `$nota_final = ((2*$E)+(3*$M)+(5*$H))/10;`. The script then uses a series of `if` and `elsif` statements to print a message based on the final grade. The messages are: "Aluno aprovado\n" for `$nota_final >= 5`, "R, Reprovado, sem direito a credito\n" for `$nota_final < 5` and `$nota_final <= 7.0`, "C, Regular, com direito a credito\n" for `$nota_final <= 8.5`, "B, Bom, com direito a credito\n" for `$nota_final <= 8.5`, and "A, Excelente, com direito a credito\n" for `$nota_final <= 8.5`. The script ends with `exit;`.

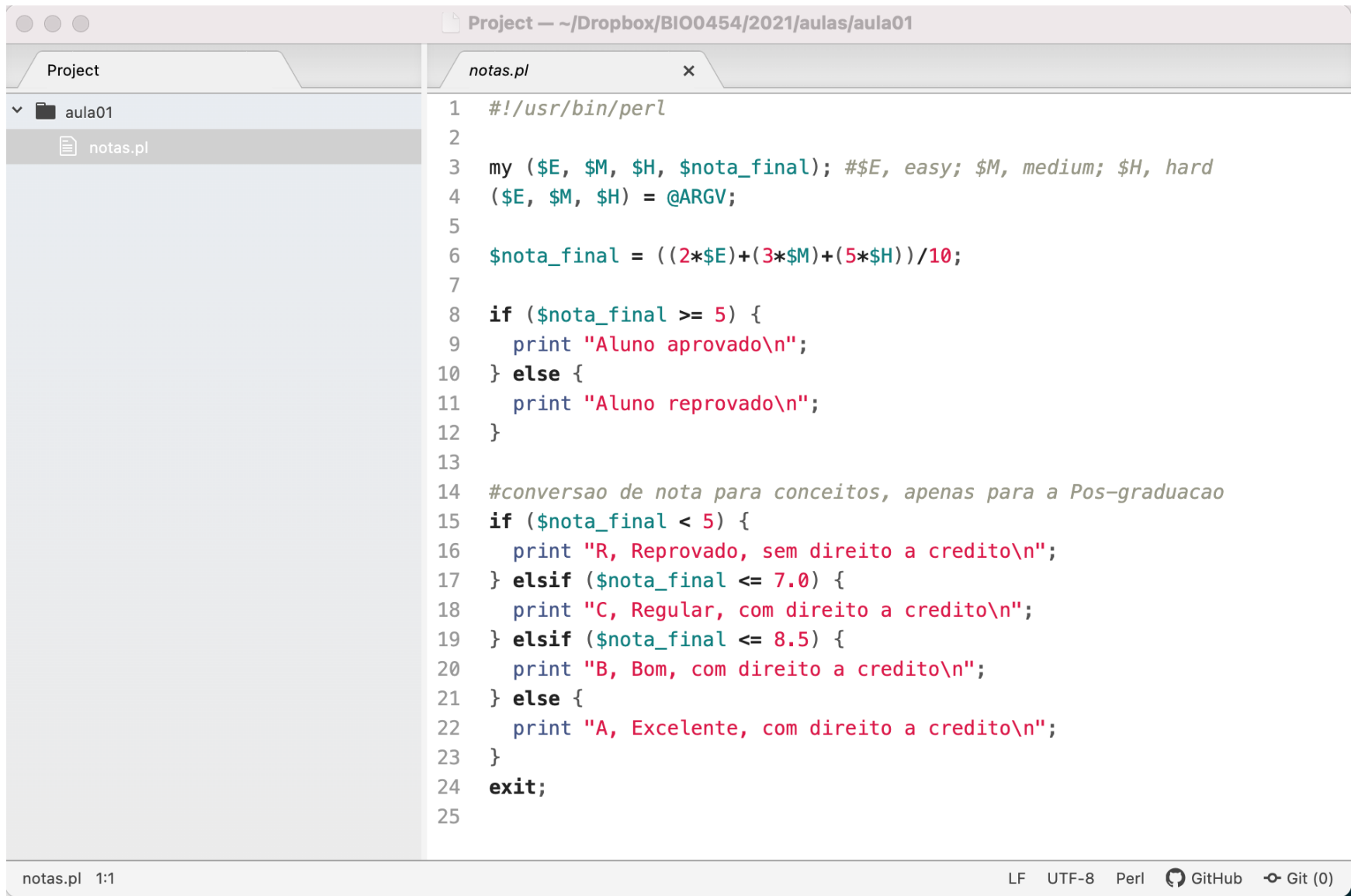
```
1  #!/usr/bin/perl
2
3  my ($E, $M, $H, $nota_final); #E, easy; $M, medium; $H, hard
4  ($E, $M, $H) = @ARGV;
5
6  $nota_final = ((2*$E)+(3*$M)+(5*$H))/10;
7
8  if ($nota_final >= 5) {
9      print "Aluno aprovado\n";
10 } else {
11     print "Aluno reprovado\n";
12 }
13
14 #conversao de nota para conceitos, apenas para a Pos-graduacao
15 if ($nota_final < 5) {
16     print "R, Reprovado, sem direito a credito\n";
17 } elsif ($nota_final <= 7.0) {
18     print "C, Regular, com direito a credito\n";
19 } elsif ($nota_final <= 8.5) {
20     print "B, Bom, com direito a credito\n";
21 } else {
22     print "A, Excelente, com direito a credito\n";
23 }
24 exit;
25
```

line: 19 / 25 col: 28 sel: 0 INS TAB mode: LF encoding: UTF-8 filetype: Perl scope: unknown

Eclipse + Epic



Atom



Project — ~/Dropbox/BIO0454/2021/aulas/aula01

Project

aula01

notas.pl

```
1  #!/usr/bin/perl
2
3  my ($E, $M, $H, $nota_final); #$E, easy; $M, medium; $H, hard
4  ($E, $M, $H) = @ARGV;
5
6  $nota_final = ((2*$E)+(3*$M)+(5*$H))/10;
7
8  if ($nota_final >= 5) {
9      print "Aluno aprovado\n";
10 } else {
11     print "Aluno reprovado\n";
12 }
13
14 #conversao de nota para conceitos, apenas para a Pos-graduacao
15 if ($nota_final < 5) {
16     print "R, Reprovado, sem direito a credito\n";
17 } elsif ($nota_final <= 7.0) {
18     print "C, Regular, com direito a credito\n";
19 } elsif ($nota_final <= 8.5) {
20     print "B, Bom, com direito a credito\n";
21 } else {
22     print "A, Excelente, com direito a credito\n";
23 }
24 exit;
25
```

notas.pl 1:1

LF UTF-8 Perl GitHub Git (0)

Para as aulas

Google meet

Editor de textos
ou IDE

Página da
disciplina

Terminal

Para as aulas

The image is a collage of four screenshots from a computer screen, arranged in a 2x2 grid.

- Top Left:** A Google Meet window titled "Meet - cog-mgak-ipt". The main video feed shows a close-up of Yoda's head. Below the video is a control bar with icons for microphone, camera, chat, and other meeting functions. The address bar shows "meet.google.com/cog-mgak-ipt".
- Top Right:** A web browser window showing a course program page. The URL is "torres.github.io". The page title is "BIO0454 / BIO5793 2021". The main heading is "Programa" (Program), followed by "Introdução à Programação de Computadores para Biologia" (Introduction to Computer Programming for Biology). Below this is "2o semestre de 2021" (2nd semester of 2021). The browser's tab bar shows several open tabs including "Torres Lab", "Lab-meeting", "CCP", "SIAF", "SAGe", "Sistemas USP", "Altmetric it!", "introprog2...", "Programa", "Sistemas...", "PERL Onli...", and "Meet - co...".
- Bottom Left:** A code editor window titled "notas.pl" showing a Perl script. The script is a simple grading program that takes three arguments: \$E (easy), \$M (medium), and \$H (hard). It calculates a weighted average and prints "Aluno aprovado" (Student approved) or "Aluno reprovado" (Student failed). The code is as follows:

```
1 #!/usr/bin/perl
2
3 my ($E, $M, $H, $nota_final); #$E, easy; $M, medium; $H, hard;
4 ($E, $M, $H) = @ARGV;
5
6 $nota_final = ((2*$E)+(3*$M)+(5*$H))/10;
7
8 if ($nota_final >= 5) {
9     print "Aluno aprovado\n";
10 } else {
11     print "Aluno reprovado\n";
12 }
```
- Bottom Right:** A terminal window titled "tatiana" showing the output of the Perl script. The terminal shows the last login time, the default shell (zsh), and the output of the script: "Aluno aprovado". The terminal prompt is "tatiana\$".

Apresentação da disciplina

Apresentação dos alunos

- Nome
- Programa de Pós-graduação / curso Graduação
- Expectativas
- Exemplo de aplicação
- Experiência em programação
- Sistema operacional

