

ePortfolios that help you get organised to showcase your technical skills in future job applications

Dr Eduardo Oliveira
[eduardo.oliveira@unimelb.edu.au]





Agenda

- Who am I?
- ePortfolio
- 7-steps to organize your code repository
- Bridging academia and industry
- Looking ahead



Who am I?



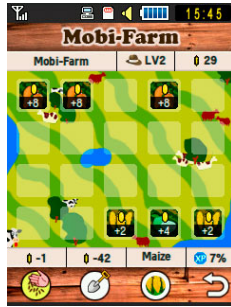
Recife, Brazil

Computer Science
(Bachelor/Masters/PhD)

Senior System Engineer
Technical consultant
14 years: Motorola/Samsung/Gemalto/Fiat

Senior Lecturer
7 years: Universities in Recife, Brazil





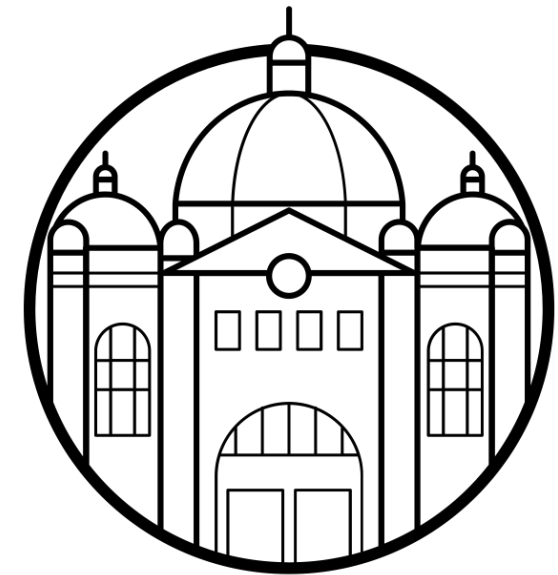


Computer Science

(post doctorate)



**Melbourne,
Australia**



Lecturer@CIS

Researcher@CSHE/MGSE



ePortfolio

A portfolio is a compilation of materials that supports your skills, qualifications, education, training, and experiences. An eportfolio is generally a static website featuring work samples and it is closely related to online resume or CV.

Using an online portfolio allows you to show real examples of your work to back up your CV. Many candidates might only say they have experienced something but uploading your portfolio gives you the chance to showcase your work, thereby making your profile stand out.



7-steps to organize your code repository

1. Keep your code repository professional
2. Keep your repo's README.md updated
3. Commit often
4. Write good commit messages
5. Branch before you build
6. Structure your folders
7. Deploy your project



7-steps to organize your code repository

1. Keep your code repository professional

Treat this as a professional space

Add a photo to your profile (no avatars)





7-steps to organize your code repository

2. Keep your repo's README.md updated

What is that repository about?

Highlight the contents of your project (summary)

Setup and Configuration details

Is there a link to a live demo for that project?

License details

The text of a license is usually stored in the LICENSE (or LICENSE.txt, LICENSE.md) file in the root of the project.



7-steps to organize your code repository

2. Keep your repo's README.md updated

Changelog

Keep track of the new changes performed after every release

At the end of each sprint/project, generate a RELEASE TAG from your repository (master branch).

Example: **<SUBJECT_CODE>_2020_<version>**

This tag includes ALL the branches that were developed and completed during that Sprint;

This is the tag that should be DEPLOYED and presented to the client (while development continues in the repository);

Add a release notes to your README (list the branches that were included in that version).



7-steps to organize your code repository

2. Keep your repo's README.md updated

Changelog

1.0.2

Updated dependencies and gulpfile [branch]

1.0.1

Updated dependencies and gulpfile [branch]

Added no-image optional class for projects without images (see above for usage) [branch]

1.0.0

Added support for optional "Show More Projects" that hides some projects by default if included [branch]

Added optional sections to display certifications, languages, etc [branch]



7-steps to organize your code repository

3. Commit often

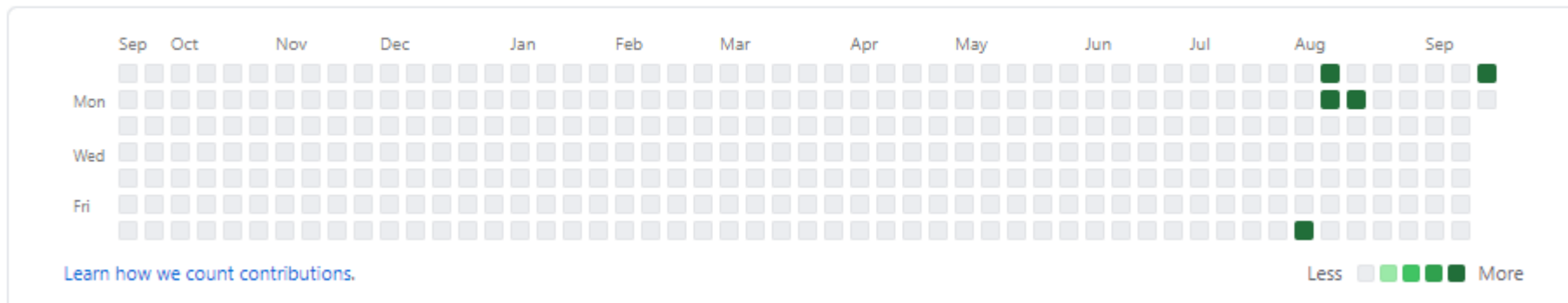
The smaller the changes, the easier it is to integrate the code later. Also, if everyone in the team is committing regularly, everyone is more likely to have the most up to date code

Each commit should be a single conceptual unit of change. Even if two changes happen in the same file, if they relate to two different bugs for example, they should go in different commits.

Conversely, if changes in two separate files are related to the same bug, they should go together in the same commit

7 contributions in the last year

Contribution settings ▾





7-steps to organize your code repository

4. Write good commit messages

Use the imperative mood in the subject line

A properly formed Git commit subject line should always be able to complete the following sentence:

If applied, this commit will *your subject line here* (this should be your commit message.
You do not need to write 'If applied...')

For example:

If applied, this commit will *refactor subsystem X for readability*

If applied, this commit will *update getting started documentation*

If applied, this commit will *remove deprecated methods*



7-steps to organize your code repository

5. Branch before you build

The naming convention simply adds prefixes to branch names, so that branches of the same type get the same prefix

feature/username/task_name

bugfix/username/task_name

test/username/task_name

This lets you search for branches in many git commands, like this:

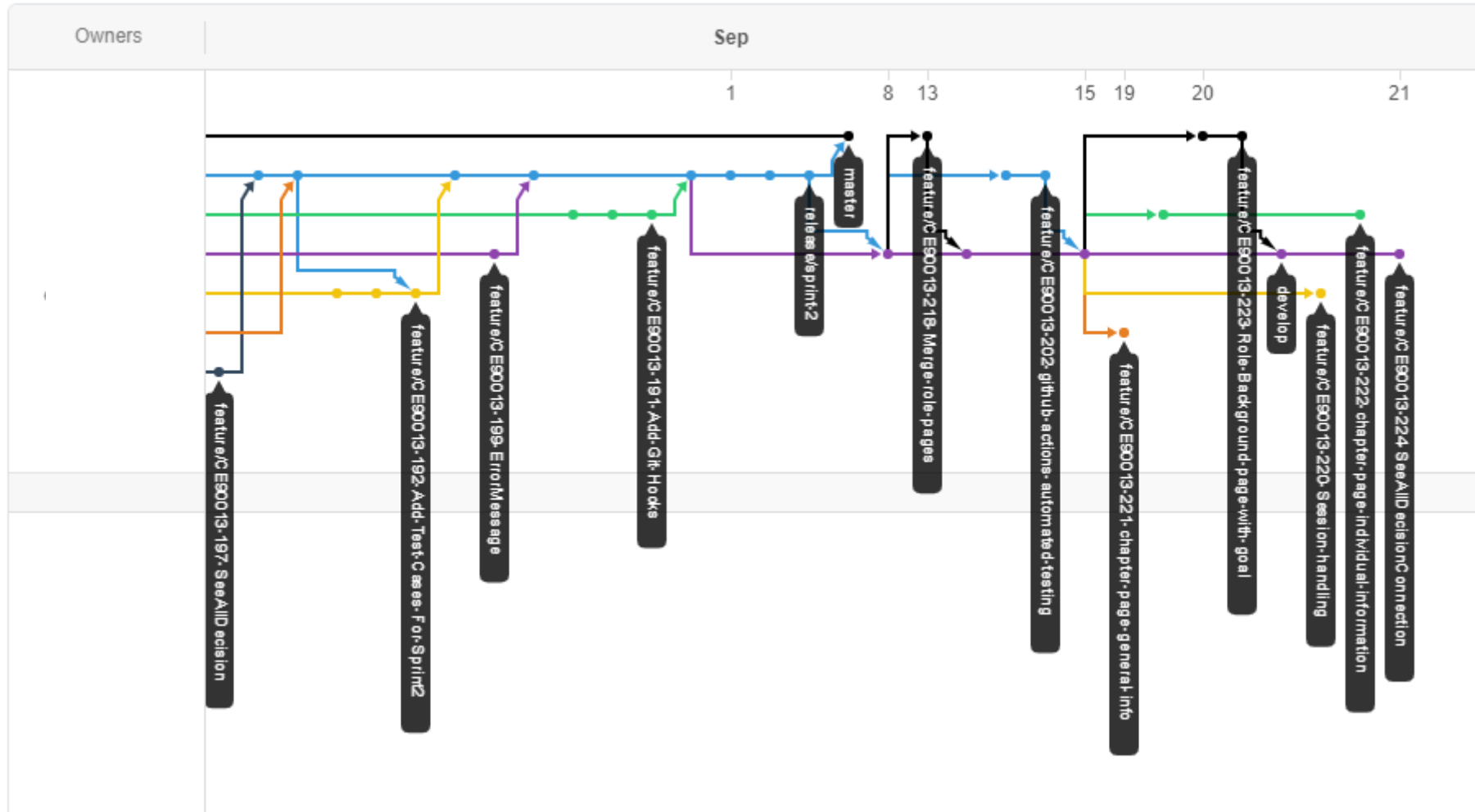
```
git branch --list "feature/*"
```

```
git log --graph --oneline --decorate --branches="feature/*"
```

```
git --branches="feature/*"
```

7-steps to organize your code repository

5. Branch before you build



7-steps to organize your code repository

5. Branch before you build

Active branches					
feature/CE90013-224-SeeAllDecisionConnection	Updated 1 hour ago by	✓	1 10	#39	Closed 🗑
feature/CE90013-222-chapter-page-individual-information	Updated 16 hours ago by		1 8	#37	Open 🗑
feature/CE90013-220-Session-handling	Updated 17 hours ago by		1 7		New pull request 🗑
develop	Updated 19 hours ago by	✓	1 9		New pull request 🗑
feature/CE90013-223-Role-Background-page-with-goal	Updated yesterday by	✓	1 8	#36	Merged 🗑
feature/CE90013-221-chapter-page-general-info	Updated 2 days ago by		1 7	#38	Open 🗑
feature/CE90013-202-github-actions-automated-testing	Updated 8 days ago by	✓	1 3	#35	Merged 🗑
feature/CE90013-218-Merge-role-pages	Updated 8 days ago by		1 2	#34	Merged 🗑
release/sprint-2	Updated 20 days ago by		1 0	#32	Merged 🗑
feature/CE90013-191-Add-Git-Hooks	Updated 22 days ago by		51 0	#19	Merged 🗑
feature/CE90013-199-ErrorMessage	Updated 22 days ago by		42 0	#31	Merged 🗑
feature/CE90013-192-Add-Test-Cases-For-Sprint2	Updated 22 days ago by		16 0	#29	Merged 🗑
feature/CE90013-197-SeeAllDecision	Updated 22 days ago by		27 0	#27	Merged 🗑

7-steps to organize your code repository

6. Structure your folders

```
├── build                # Compiled files (alternatively `dist`)  
├── docs                 # Documentation files  
├── src                  # Source files (alternatively `lib` or `app`)  
├── test                 # Automated tests  
├── tools                # Tools and utilities  
├── LICENSE  
└── README.md
```

Use short lowercase names at least for the top-level files and folders except LICENSE, README.md.

Automated tests are usually placed into the **test** or, less commonly, into the **spec** or **tests** folder.

```
├── ...  
├── test                # Test files  
│   ├── benchmarks    # Load and stress tests  
│   ├── integration    # End-to-end, integration tests  
│   └── unit           # Unit tests  
└── ...
```

7-steps to organize your code repository

6. Structure your folders

Often it is beneficial to include some reference data into the project, such as Rich Text Format (RTF) documentation, which is usually stored into the `docs` or, less commonly, into the `doc` folder.

```
.
├── ...
├── docs
│   ├── TOC.md
│   ├── faq.md
│   ├── misc.md
│   ├── usage.md
│   └── ...
└── ...
```

Documentation files (alternatively ``doc``)
Table of contents
Frequently asked questions
Miscellaneous information
Getting started guide
etc.



7-steps to organize your code repository

7. Deploy your project

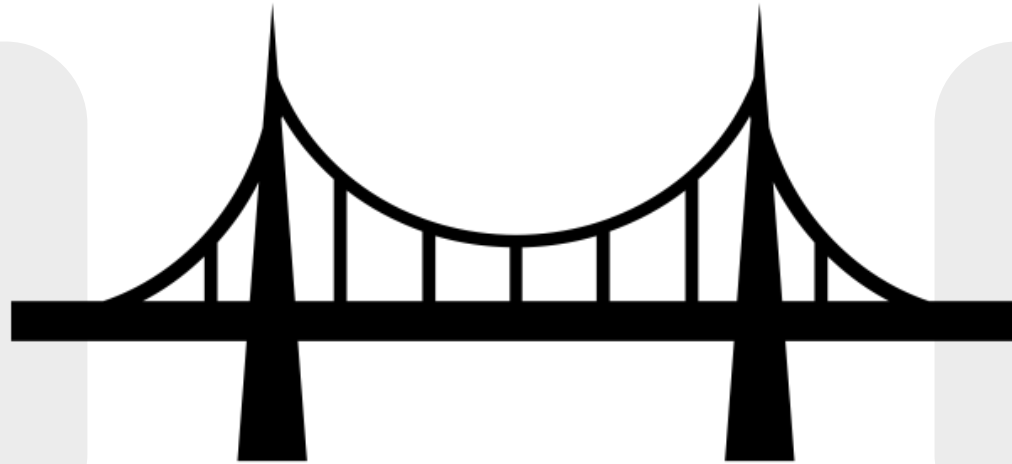
Have your project ready to show! One of the worst possible things is to show up to an interview and offer to show your super cool website and then have to do a bunch of debugging or configuration in order to launch it (true story).

Deploy your project and have it publicly available via URL (keep the URL simple and easy to type for non-tech people).



Bridging academia and industry

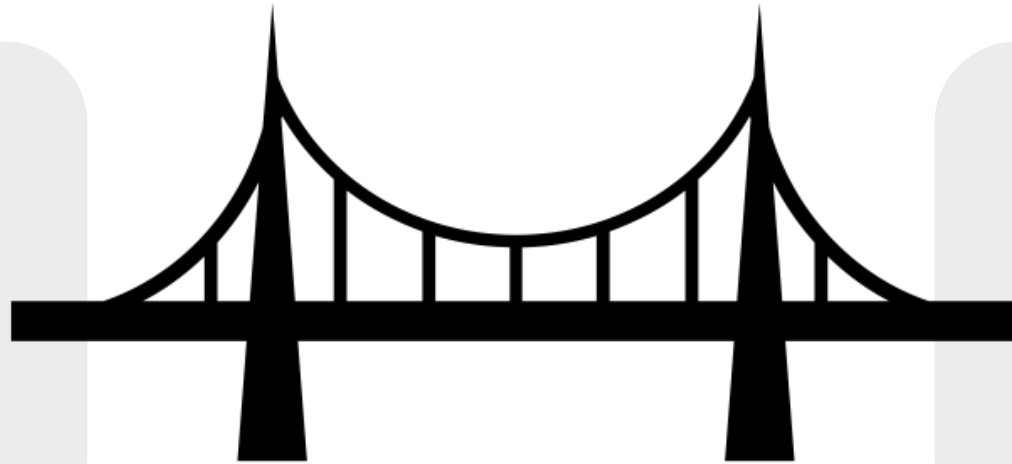
CIS



INDUSTRY



Bridging academia and industry



INDUSTRY



Bridging academia and industry

Insert Web Page

This app allows you to insert secure web pages starting with `https://` into the slide deck. Non-secure web pages are not supported for security reasons.

Please enter the URL below.

`https://`

Note: Many popular websites allow secure access. Please click on the preview button to ensure the web page is accessible.





Bridging academia and industry

Insert Web Page

This app allows you to insert secure web pages starting with `https://` into the slide deck. Non-secure web pages are not supported for security reasons.

Please enter the URL below.

`https://`

Note: Many popular websites allow secure access. Please click on the preview button to ensure the web page is accessible.





Bridging academia and industry

Insert Web Page

This app allows you to insert secure web pages starting with https:// into the slide deck. Non-secure web pages are not supported for security reasons.

Please enter the URL below.

https://

Note: Many popular websites allow secure access. Please click on the preview button to ensure the web page is accessible.





Bridging academia and industry

Insert Web Page

This app allows you to insert secure web pages starting with `https://` into the slide deck. Non-secure web pages are not supported for security reasons.

Please enter the URL below.

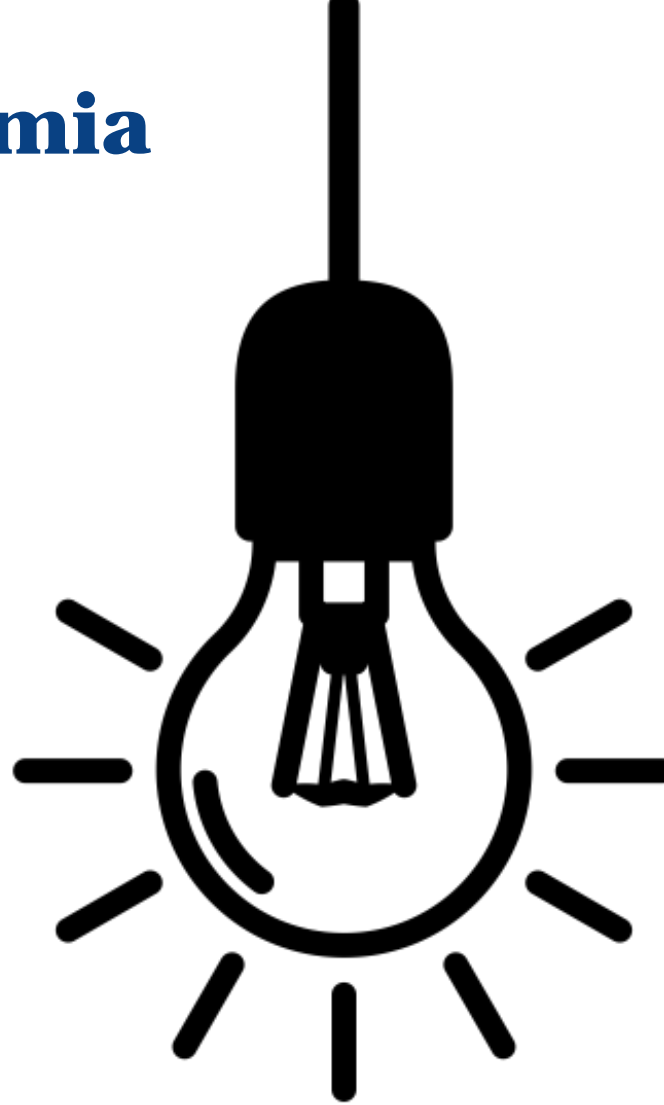
`https://`

Note: Many popular websites allow secure access. Please click on the preview button to ensure the web page is accessible.





Bridging academia and industry





Bridging academia and industry



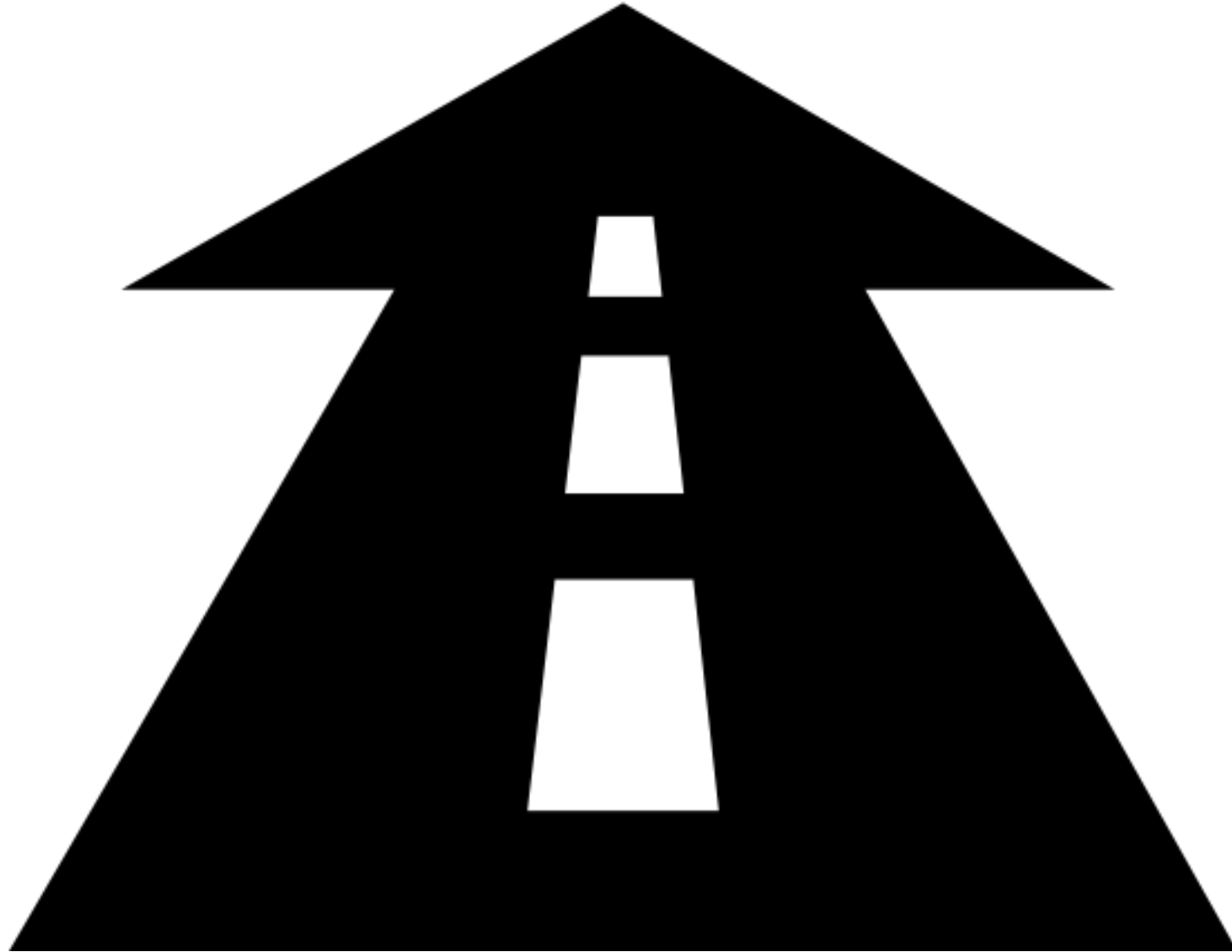
Workshop: Mastering LinkedIn Wednesday 23 September, 11am

Now is a great time to develop your career goals and action plan. Join us and learn to leverage LinkedIn as a professional networking and research tool to align your professional brand to your industry area.

This interactive workshop is open to all students. Please [book via Careers Online](#).



Looking ahead





Thank you very, very much!

ePortfolios that help you get organised to showcase your technical skills in future job applications

Dr Eduardo Oliveira
[eduardo.oliveira@unimelb.edu.au]

CIS Industry Awareness Series

