Design Principles for Low Carbon and Sustainable Computing

Ismael Velasco

HELLO!

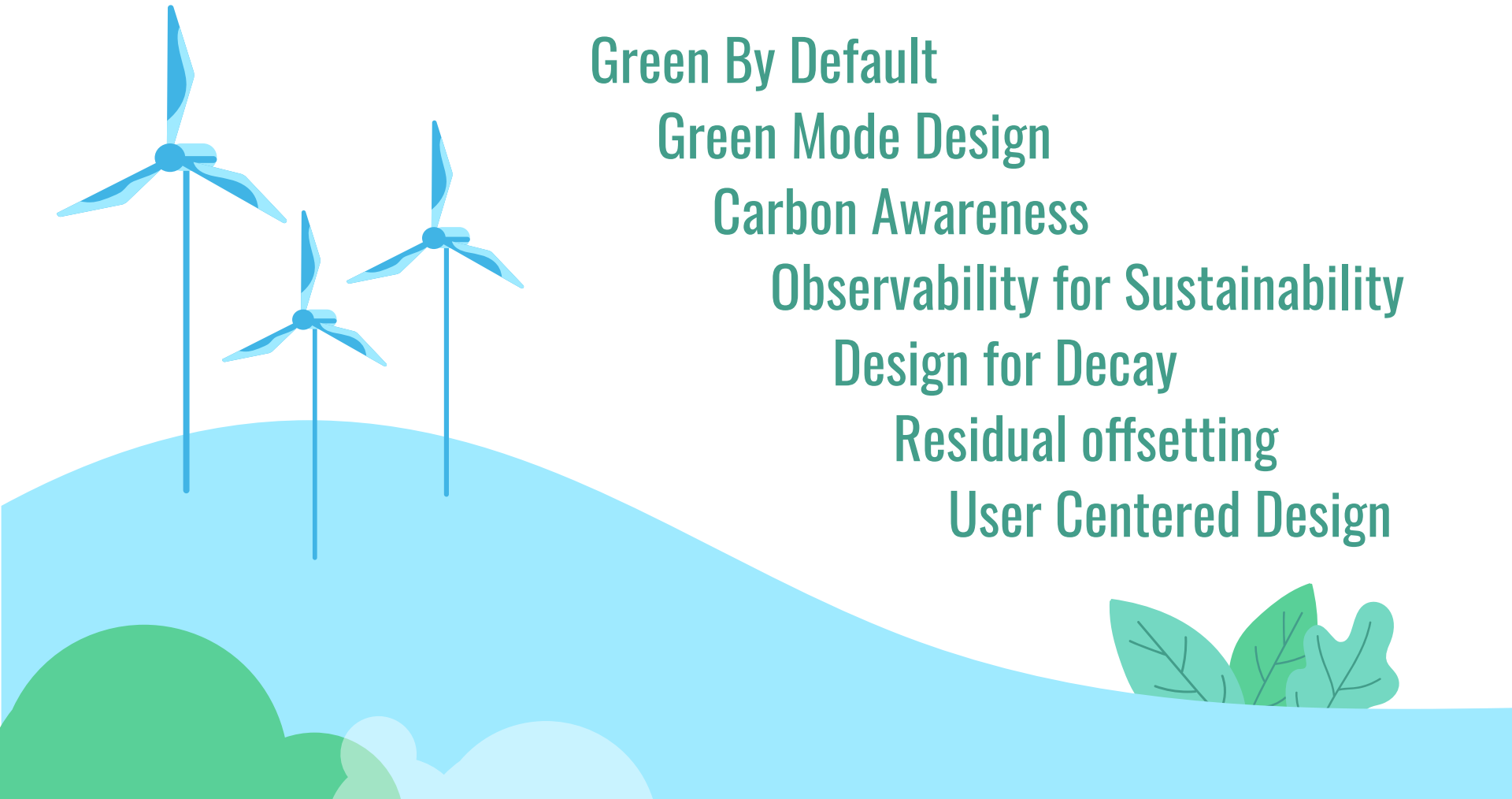
I'm Ismael Velasco, Mexican cosmopolite living in Merida, Mexico.

Software engineer, professional opera singer.

Only one 👆 is true.

You can find me at ismaelvelasco.dev

I am the founder of adorafoundation.org and lead author of the State of Green Software Report for the Green Software Foundation



Green By Default

Green Mode Design

Carbon Awareness

Observability for Sustainability

Design for Decay

Residual offsetting

User Centered Design

Green By Default

Optimise for energy efficiency using best in class tools and patterns

Apply “Progressive Enhancement”: design for maximum backward compatibility and minimal default functionality first.

Modularise additional functionality, only engaging it when actually needed, rather than begin from maximum functionality first.

Green Mode Design

Green Mode Design empowers users to choose graceful degradations from functional defaults in exchange for energy, data and environmental savings.

Software equivalent of “energy saver mode” in hardware or eco-mode in cars.

- make image display or resolution quality optional
- offer sound or transcript-only versions of videos
- offer less powerful but more energy efficient algorithms for queries
- offer latency increases for diminished energy intensity

Carbon Awareness

Link functionality to energy intensity of local electricity grid using historical electricity data, APIs like Watttime or the Green Software Foundation's Carbon Aware SDK.

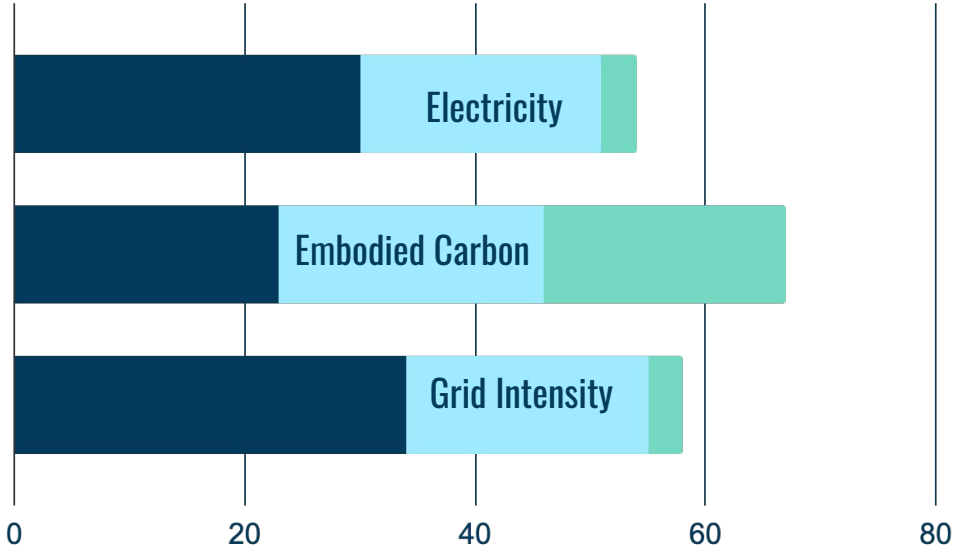
- Time shifting: queue up jobs for times when the electricity grid is running on clean energy in your locality
- Location shifting: route jobs to localities where the electricity is greenest at a given time.
- Demand shifting: increase or degrade functionality in relation to the current energy intensity of the electricity grid

Observability for Sustainability

Any digital product should allow for real time metrics both server and client side tracking, at least, electricity consumed, grid intensity, and embodied carbon.

Environmental impact tracking should be incorporated at the software design stage

- The Software Carbon Intensity specification by the Green Software Foundation is a promising approach
- Complementary or alternative metrics should also be considered where appropriate



Design for Decay

Reuse, repair and retiring should be added to software lifecycle planning. Incorporate automated protocols for managing disuse, decay and end of life.

What’s not being used should not be live. What doesn’t need to be stored, should be deleted.

This applies to a software product, its individual features and its data, and should be automated at the design stage.

Residual offsetting

After the other design principles are implemented, **and only then**, any residual CO2 from carbon intensity metrics should be offset on both producer and consumer ends.

Tooling for residual CO2 offsets should be implemented at the software design stage.

Offsetting should be science driven, holistic, and verified.

User Centred Design

Design should aim not for purity but for balance between environmental, social, technical and commercial sustainability.

Software design should be implemented with the user experience in mind and if possible with user participation.

“Users” be defined at the design stage in equitable, diverse and inclusive ways, including direct and indirect consumers.

“User experience” should include not only usability but justice and care (accessibility, privacy, equity, wellbeing)

Thanks!

Do reach out!

linkedin.com/in/IsmaelVelasco