

# MATTIA PISPISA

[mattia.pispisa@gmail.com](mailto:mattia.pispisa@gmail.com)

I've always been passionate about mobile development. I started out with iOS, then moved on to a hybrid approach with Flutter. I've been building apps professionally with Flutter since its first release in 2018. After my love for mobile development, I also developed a strong interest in Web Front-End, and I'm now specialized in React as well. Whenever I approach something new, I like to begin with some theoretical study 😎, do an analysis, and then put everything into practice 🧑.

 [www.linkedin.com/in/mattia-pispisa-252363189](https://www.linkedin.com/in/mattia-pispisa-252363189)

 [github.com/MattiaPispisa](https://github.com/MattiaPispisa)

 [dev.to/mattia](https://dev.to/mattia)

 [g.dev/mattia\\_pispisa](https://g.dev/mattia_pispisa)

 [bsky.app/profile/mattia-pispisa.bsky.social](https://bsky.app/profile/mattia-pispisa.bsky.social)

 [twitter.com/MattiaPispisa](https://twitter.com/MattiaPispisa)

 [www.instagram.com/mattiapispisa](https://www.instagram.com/mattiapispisa)

 [t.me/MattiaPispisa](https://t.me/MattiaPispisa)

# PROJECTS

## DART TIME ↗

A comprehensive Dart library that extends and enhances Dart's built-in time functionality. This library provides powerful extensions to DateTime and Duration, plus specialized classes for advanced time operations.

[#library](#)[#dart](#)

## CRDT ↗

A project for managing conflict free replicated data. It is divided into several libraries:

- [A library that implements the HLC \(Hybrid Logical Clock\) protocol for managing replicated data.](#)
- [A library that implements the CRDT \(Conflict-free Replicated Data Type\) protocol for managing replicated data.](#)
- [A library for synchronizing crdt documents in real time between different users.](#)
- [A library that supports the persistence of crdt data using hive.](#)

[#library](#)[#dart](#)

## EN .ENV READER ↗

A Visual Studio Code extension for managing .env files with a user-friendly interface.

[#library](#)[#typescript](#)

## EN YUP DECORATOR ↗

A TypeScript library for creating yup schema decorators, simplifying schema validations.

[#library](#)[#typescript](#)

## DATA MIGRATION ↗

A library for applying a series of migrations to update old versions of data structures to the most recent one.

[#library](#)[#dart](#)

## EN LOGGER ↗

EnLogger allows you to write log messages according to your needs without restricting you to writing messages to the debug console or other systems. It maintains a list of EnLoggerHandlers internally. You can implement your own EnLoggerHandler based on your specific requirements. Each time you want to log a message with EnLogger, each EnLoggerHandler will be invoked to perform the write operation.

[#library](#)[#dart](#)

## FILE UPLOADER ↗

A file upload functionality that is implementation-agnostic.

This library provides the capability to:

- upload a complete file;
- upload a file in chunks;
- upload a file in chunks with the ability to pause and resume the upload from where it left off (restorable chunks file upload).

Plugins

- [http\\_file\\_uploader](#): This plugin allows you to implement file uploads using the http library;
- [dio\\_file\\_uploader](#): This plugin allows you to implement file uploads using the dio library.

UI package

#library

#dart

## POSIX TIMEZONES ↗

Posix timezones package. This package is designed to simplify working with timezones, allowing developers to read and write timezones in posix standard.

#library

#dart

## GOOD ENUMS ↗

Code generator to make enums easier to use.

#library

#dart

## HOW TO TRAIN YOUR USEREDUCER ↗

A talk given at the ReactJS meetup in Verona in 2023.

Code "how-to-train-your-use-reducer"

Splitting the view from the controller is foundational to build great applications. We tend to rely on external libraries to manage their state. But React itself has a great off-the-shelf state management tool: the useReducer. In this talk we're going to get to know more about it, and especially learn how to enhance it tailoring it to our needs. You'll discover what you have been missing.

#talk

#react

## EFFICIENT TREE MANIPULATIONS ↗

An exercise on "algorithms and data structures" completed during my bachelor's degree in computer science.

#exercise

#c

#university

# POST

## GO ROUTER NAVIGATION OBSERVER ↗

How to implement a navigation observer with Go Router.

#dart

#go\_router

May 17, 2025

## EFFICIENT LOGGING IN APPLICATIONS ↗

An approach to logging messages of your applications with also my own package solution in dart/flutter.

#dart

#logging

September 7, 2024

## EXPAND YOUR FUTURE, EXTENSIONS IN DART ↗

You can extend functions, and if these functions are asynchronous, you can create something truly remarkable. Let's dive into some practical and simple demonstrative examples.

#flutter

#dart

#async

April 12, 2024

## FLUTTER, NOT YET READY FOR WEB ↗

An analysis of flutter current limitations on web.

#flutter

#dart

#javascript

December 12, 2023

## REACT HOOKS AND FIBER DEEP DIVING ↗

Starting from a code example that had a particular bug will be explored the source code of react focusing especially on "hooks" and "fiber" nodes.

#react

#javascript

October 28, 2023

## FLUTTER MONOREPO & DEPENDENCY INJECTION ↗

Monorepo refers to the practice of software development in which all components of an application or system are maintained within a single repository. In the case of a flutter application an excellent monorepo manager is Melos, here a tutorial on how to create and manage it.

#flutter

#dart

#dependency\_injection

October 10, 2023

## MASTERING DEPENDENCY INJECTION IN FLUTTER ↗

One of the 5 SOLID principles capitulates that a high-level module should depend only on abstract class not on the implementation. This leads to less coupling between modules. In this article we will focus on how to apply this pattern in flutter.

#flutter    #dart    #dependency\_injection

August 10, 2023

# SKILLS

## DART

5/5

Advanced expertise in Dart since 2018. I follow language releases closely, apply them in personal projects, blog posts, and professional work. My largest personal Dart project is a CRDT system.

code\_generation    dart\_frog    rx\_dart    melos  
hive    get\_it

## FLUTTER

5/5

Expert in Flutter application development, particularly for mobile platforms (Android/iOS). Strong focus on cross-platform consistency and performance.

bloc\_pattern    animations    platform\_channels

## JAVASCRIPT & TYPESCRIPT

5/5

I develop almost exclusively in TypeScript (not JavaScript) because I believe it leads to more maintainable code over time and within teams. I leverage TypeScript's full potential for composing complex data structures.

## PYTHON

3.5/5

Primarily used for training, fine-tuning, and serving artificial intelligence models.

pytorch    tensorflow    scikit-learn    pandas

## IOS

3.5/5

## JAVA

3.5/5

## REACT & REACT ROUTER

5/5

Expert in building web applications with React and React Router. I've examined React's source code multiple times to fully understand its inner workings. I've written blog posts about it.

## ANDROID

4.5/5

## SQL

3.5/5

## C#

3.5/5

# EXPERIENCE

## MOB & WEB ENGINEER ↗

MVLABS

I'm the team lead for both the mobile (MOB) and web (WEB) environments. In my role, I'm responsible for application analysis and design, organizing and dividing work within the team, and providing technical training. In both areas, I oversee the company's core product as well as consulting or development projects commissioned by other companies.

On the mobile side, the main stack I've chosen to use for app development includes Dart/Flutter as the primary language and framework, GetIt + Injectable for dependency injection, BLoC for application business logic, Hive/Hive\_CE for data persistence, and Melos for monorepo management.

On the web side, the main stack I've chosen includes: TypeScript as the language, React as the library, React Router as the framework, Inversify for dependency injection, Redux for global state, Reducer + Immer + Yup for business logic, Vite as the bundler, and Turborepo for monorepo management.

Feb 2021 - Present

## DESIGNER AND DEVELOPER OF A WEB CONNECTOR FOR THE PROXFLOW APPLICATION ↗

TECNOTRONICA S.A.S.

I designed and developed a connector for the ProxFlow application. The connector was split into a:

- Back-End developed in Dart using principally with dart\_frog
- A Web-App built with React. Initially, the client was also supposed to be developed in Dart/Flutter, but Dart's interoperability with JavaScript — especially for complex tasks like displaying large PDF files (sometimes up to 100 MB) — was not yet mature enough for a production-grade application.

Nov 2022 - May 2023

## IOS MOBILE DEVELOPER ↗

ARCHEIDO

Jul 2019 - Mar 2020

## **EMPLOYEE**

FILL IN THE BLANKS S.R.L.

Jun 2015 - Jul 2015

# EDUCATION

## MASTER'S DEGREE

UNIVERSITÀ DEGLI STUDI DI UDINE

Master's Degree in Computer Science, Computer Science (Interactive Systems)

Vote: **110/110** with merit to the best graduates



Oct 2019 - Oct 2021

## BACHELOR'S DEGREE

UNIVERSITÀ DEGLI STUDI DI UDINE

Bachelor's degree in Computer Science, Computer Science

Vote: **110/110**

Oct 2016 - Oct 2019

## DIPLOMA

ISIS ARTURO MALIGNANI

Diploma in telecommunications

Jan 2012 - Jan 2016

# BOOKS READ

A collection of my favorite books I've read. The list isn't complete yet, but I'll keep adding more over time.

[1984](#) - George Orwell

[A First Course in Operations Research](#) - Giuseppe Lancia

[Clean Code](#) - Robert C. Martin

[Dive Into Design Patterns](#) - Alexander Shvets

[Emotional Design](#) - Donald Norman

[Fluent React](#) - Tejas Kumar

[Flutter Complete Reference 2.0: The ultimate reference for Dart and Flutter](#) - Alberto Miola

[Human-Computer Interaction. I fondamenti dell'interazione tra persone e tecnologie](#) - Luciano Gamberini, Luca Chittaro, Fabio Paternò

[Introduction to Algorithms](#) - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

[Javascript: The Definitive Guide](#) - David Flanagan

[Learn React Hooks](#) - Daniel Bugl

[Learning React](#) - Alex Banks, Eve Porcello

[Machine Learning for Coders](#) - Andrej Karpathy

[Refactoring: Improving the Design of Existing Code](#) - Martin Fowler

[Reti di calcolatori](#) - Larry L. Peterson, Bruce S. Davie

# PASSIONS

I love designing, programming, and always finding new challenges to push myself.

Besides that, in my free time, I'm passionate about motorsports, especially Formula 1, always Ferrari 🏎 !

I often go to the gym 🏋️ and together with some friends, I manage a page dedicated to organizing mountain excursions 🏔 . Follow it too: [montanari\\_scarsi](#)

I also love Marvel, Star Wars, and Naruto 🍀 !