

diogo castro

20 Greenvale • Drumbeg, United Kingdom • 07552 081 034

E-MAIL dc@diogocastro.com • **BLOG** diogocastro.com/blog • **WEB** diogocastro.com • **GITHUB** github.com/dcastro

PROFILE Proficient **C#** developer, **Haskell & PureScript** junkie.

Proactive with good teamwork skills.

Functional programming aficionado.

Engaged participant on **StackOverflow**.

EDUCATION **Master's in Informatics and Computing Engineering**
Faculty of Engineering of the University of Porto, Portugal
September 2007 – February 2013

Main Topics: Software Engineering, Project Management, Agile Development, Mobile Computing, Algorithms and Data Structures, Databases, Web Development, Artificial Intelligence, Distributed Systems.

TRAININGS **Advanced C# Course**
Instil Software, May 2015
Focused on concurrency, functional programming in C#, LINQ.

JavaScript Training
Instil Software, February 2016
Functional programming in JS, prototype inheritance, AngularJS.

CERTIFICATES **Functional Programming Principles in Scala**
Martin Odersky, École Polytechnique Fédérale de Lausanne, June 2013

Microsoft Certified Professional - Programming in C#
Microsoft, March 2014

SKILLS **Object-oriented programming** • **Functional programming** • **Scala** • **Haskell** • **PureScript** • **C#** • **TypeScript** • **Software design patterns** • **Agile** • **Dependency injection** • **TDD** • **Property-based testing** • **git** • **ElasticSearch** • **CI/CD**

EXPERIENCE

Fraunhofer

Researcher, August 2012 – March 2013.

Developed the navigation module for an Android application, using both turn-by-turn and landmark-based approaches. Studied and compared the efficiency of these approaches in navigating older adults with mild dementia. Developed complex heuristics to evaluate landmarks data retrieved from OpenStreetMap, and used the device's sensors (e.g., gyroscope, accelerometer) to locate and navigate the user.

NantHealth

Software Engineer, May 2013 – July 2014

Senior Software Engineer, July 2014 - Present

Responsible for designing, unit testing, implementing and deploying a variety of applications, such as:

- RESTful APIs, using ASP.NET Web API, ActiveMQ and Couchbase, Castle Windsor, Bootstrap, Coffeescript, LESS, MS SQL Server;
- Front-end single page applications, using AngularJS, JavaScript/TypeScript;
- Internal libraries (e.g. for standardized logging and messaging) written in C#;
- An internal framework that acts as a concurrent general-purpose service host, handling multiple cross-cutting concerns;
- Routing of messages between applications using Java and Apache Camel.

Experience with CI/CD tools, e.g. Chef, Rundeck and TeamCity.

I've also built some small internal tools to help streamline my coworkers' and my day-to-day activities, using Scala and Haskell.

I help organize a weekly Brown Bag Session at lunch hour, during which people talk about topics that interest them. Sometimes I bring a *kata* (i.e. an exercise) for people to solve in a language of their choice, and share their solutions at the end.

OPEN SOURCE **Contributed to AutoFixture**

<https://github.com/AutoFixture/AutoFixture>

AutoFixture is an open source library for .NET designed to minimize the 'Arrange' phase of your unit tests in order to maximise maintainability. Its primary goal is to allow developers to focus on what is being tested rather than how to setup the test scenario, by making it easier to create object graphs containing test data.

Author of csi-init

<https://github.com/dcastro/csi-init>

Csi-init is a simple command line tool written in Haskell, which allows you to launch Roslyn's C# REPL (csi) preloaded with all the assemblies found in one or more directories.

Author of JSend WebApi & Client

<https://github.com/dcastro/JSendWebApi>

JSend.WebApi is an extension of ASP.NET Web API for designing APIs using the JSend protocol (<https://labs.omniti.com/labs/jsend>).
JSend.Client is a library for consuming JSend APIs.

Author of Sequences

<https://github.com/dcastro/Sequences>

Sequences is a port of Scala's *Stream[+A]* to C#. A *Sequence<T>* is an immutable lazy list whose elements are only evaluated when they are needed. It is composed by a head (the first element) and a lazily-evaluated tail (the remaining elements).

Author of DequeNET

<https://github.com/dcastro/DequeNET>

A concurrent lock-free deque (double-ended queue) implementation in C# - push/pop/peek operations in constant time $O(1)$ - and a regular deque implemented as a ring buffer.

Author of MementoContainer

<https://github.com/dcastro/MementoContainer>

An alternative approach to the Memento design pattern. Through reflection, the container takes a snapshot of your objects' state, so that you can easily rollback when recovering from errors or implementing an "undo" mechanism.

PROJECTS

MetroTasks

<https://www.microsoft.com/en-us/store/apps/metrotasks/9wzdncrdfxdk>

Metro Tasks is a productivity application for Windows 8 / Windows RT that helps you keep track of your to-do's. The app features synchronization with Google Tasks, type-to-search, tile and lock screen updates.