

**'UNIVERSIDADE DE ITAÚNA**  
**CIÊNCIA DA COMPUTAÇÃO - 6º PERÍODO**  
**LINGUAGENS DE PROGRAMAÇÃO**

ARLEY AUGUSTO E SILVA  
IAGO ANTUNES FERREIRA

Itaúna  
01/09//2022

# Dart

## 1.História

Dart foi lançado na GOTO Conference 2011, que ocorreu nos dias 10 e 11 de outubro de 2011 em Aarhus, na Dinamarca. **Dart** é uma linguagem de script voltada à web desenvolvida pela Google. O projeto foi iniciado por Lars Bak, um programador dinamarquês famoso por seu trabalho em máquinas virtuais e contribuições para a criação dos navegadores Google Chrome e Kasper Lund, engenheiro de software.

Antes de ser lançado ao mercado, o Dart precisou passar pela análise de um comitê técnico, o TC52 Ecma International. Foi ele que certificou que a linguagem utiliza todos os padrões e que é aceita nos navegadores modernos.

Em 2013 foi lançada a primeira versão estável, denominada Dart 1.0. Em 2018, ele sofreu uma reforma, surgindo o Dart 2.0, otimizado para o desenvolvimento no Client-Side para web e mobile.

Para padronizar a linguagem de programação recém-criada, a Ecma International criou um comitê técnico conhecido como TC52. O fato de que os códigos Dart podem ser compilados em JavaScript tornou a linguagem bastante compatível com muitos navegadores modernos. Isso levou à sua inevitável padronização em julho de 2014, quando a Ecma International aprovou a primeira edição da especificação da linguagem Dart na 107ª Assembleia Geral do órgão internacional.

Em dezembro de 2014, mais uma edição da especificação da linguagem Dart foi aprovada pela Ecma International, sendo a segunda edição da especificação da linguagem Dart.

O primeiro compilador que conseguia gerar códigos JavaScript a partir do Dart era conhecido como **dartc**, mas foi descontinuado pelo Google. Seu sucessor foi o Frog, um compilador escrito em Dart, mas não conseguiu compilar os códigos completamente. O terceiro compilador foi o dart2js que também foi escrito em Dart e atualmente ainda é usado.

## 2.Objetivo

O objetivo da linguagem Dart foi inicialmente a de substituir a JavaScript como a linguagem principal embutida nos navegadores. Programas nesta linguagem podem tanto serem executados em uma máquina virtual quanto compilados para JavaScript. Porém, com a evolução da linguagem e com o passar dos anos, ela hoje pode ser considerada uma linguagem multi-paradigma, embora a linguagem apresente fortes estruturas típicas de linguagens orientadas a objetos.

## 3.Conceitos

### 3.1 Organização do Código

Na linguagem Dart, cada arquivo de origem deve importar tudo a que se refere, incluindo seus outros arquivos de origem e pacotes de terceiros. Não há namespaces da mesma maneira, e você geralmente se refere a arquivos por meio da localização do sistema de arquivos. Variáveis e funções podem ser de nível superior, não apenas classes.

### 3.2 Tipos de dados

Dart elimina essa distinção porque tudo é um objeto; todos os tipos, em última análise, derivam do tipo Object. Todas as primitivas são inicializadas implicitamente para null. Assim você não pode assumir que os valores padrão de inteiros são zero, como está acostumado em C# ou Java, e pode ser necessário adicionar verificações nulas.

Blue modern presentation

### 3.3 Tipos Numéricos

Ao contrário dos inteiros utilizados por outras linguagens de 8 a 64 bits com sabores assinados e não assinados, o tipo inteiro usado no Dart é apenas int um valor de 64 bits e BigInt para valores maiores

### 3.4 Coleções

Coleções e genéricos são muito parecidos com o que conhecemos de outras linguagens. A principal diferença é que não há arrays de tamanho fixo: apenas use o List tipo de dados onde quer que você use um array.

```
final a = [1, 2, 3];
```

```
final b = {1, 2, 3};
```

```
final c = {'a': 1, 'b': 2};
```

## 4. Funcionalidades

### Dart Native

- É utilizada para desenvolvimento mobile, desktop e de servidor, entre outros. Para realizar a produção de código, ele inclui a máquina virtual do Dart com compilação JIT (just-in-time) e compilador AOT (ahead-of-time).

### Dart Web

- Tem como foco o desenvolvimento web e junto *dartdevc*, que compila o código para o desenvolvimento, e o *dart2js*, que é indicado para fazer o build para produção. Neste caso, o resultado final é um Javascript, que será facilmente interpretado por qualquer navegador visto que essa linguagem é universal.

## 5. Detalhes

- Segue o paradigma orientado a objetos
- Todos os objetos herdam da classe `Object`
- Fortemente tipada, mas não é necessário colocar um tipo, pois Dart consegue inferir os tipos

## 6. Vantagens

- **Velocidade**

Uma das principais vantagens da linguagem Dart está ligada à sua velocidade. Ela permite a criação de aplicações de forma muito rápida devido aos seus recursos JIT e AOT.

- **Segurança**

A linguagem Dart tem ainda mais uma vantagem: é mais segura em termos de possíveis erros de desenvolvimento. Por ser uma linguagem compilada, muitos dos erros de código podem ser encontrados ainda durante esse processo.

## 7. Desvantagens

- **Pouco tempo de Mercado**

Algumas desvantagens da Dart estão ligadas ao fato de que ela é relativamente recente no mercado, e sua ampla utilização é ainda mais recente. Desse modo, aprender a utilizá-la pode ser uma tarefa complexa se você não possui conhecimentos prévios de POO.

- **Sem diversidade de Frameworks**

Nesse sentido a linguagem Dart ainda tem mais uma desvantagem: apesar de possuir um framework popular e interessante, muitas linguagens possuem uma diversidade de frameworks com os quais podem ser utilizadas.

# C SHARP

## 1.História

O desenvolvimento do C# (pronuncia-se: C Sharp), de acordo com o website Wikipédia (2013), teve início no final do século XX (1999). A convite da Microsoft, Anders Hejlsberg formou uma equipe de programadores para desenvolver uma nova linguagem de programação. A primeira linguagem criada pela equipe de Hejlsberg recebe o nome de Cool. Mais tarde, na virada de século, a linguagem Cool foi rebatizada para C#.

O C# surgiu com a finalidade de possibilitar a criação de soluções executáveis sobre a plataforma .NET Framework, com o intuito de flexibilizar o desenvolvimento de aplicativos. Dessa maneira, o desenvolvedor não cria soluções para um dispositivo eletrônico específico, e sim para a plataforma .NET Framework.

Para a codificação de sistemas baseados em .NET Framework, a Microsoft lançou o Visual Studio. Esta ferramenta é uma IDE (Integrated Development Environment) que conta com um editor de código, compilador, com modelos de projetos de exemplos, designers e assistente de códigos.

MSDN (2012), diz que os algoritmos escritos em C# são executados no .NET Framework, um componente para Windows que inclui um sistema de execução virtual da aplicação por meio do Common Language Runtime (CLR) e um conjunto de bibliotecas de classes, além de contar com uma solução comercial da Microsoft, que faz parte da infraestrutura de linguagem comum (CLI). O CLI é tido como um padrão internacional para a criação e execução de ambientes de desenvolvimento onde as linguagens e as bibliotecas trabalham em sincronia.

Ainda de acordo com MSDN (2012), a compilação em C# é processada em uma Intermediate Language (IL) que está em conformidade com a especificação CLI. O código IL e seus recursos, tais como bitmaps e strings, são armazenados no disco rígido em um arquivo executável chamado de Assembly. No momento da execução de um programa em C#, o Assembly é carregado para o CLR. Uma vez aprovado os requisitos de segurança, o CLR executa a compilação Just in Time (JIT) para converter o código IL em linguagem de baixo nível.

## 2.Características

A criação do C# foi baseada nas linguagens de programação C, C++ e Java. Deste modo, programadores habituados com uma dessas linguagens são capazes de desenvolver soluções sem maiores dificuldades, pois ela reúne recursos de cada uma destas linguagens base e adiciona novos recursos exclusivos.

Em relação a linguagem C, o C# tem foco na compilação de soluções de alto nível, já a linguagem C é voltada para o desenvolvimento de baixo nível. Outras comparações, de acordo com MSDN (2012), o C#, simplifica de modo significativo a complexidade do C++ e introduz novos elementos não disponíveis no Java, tais como: tipos primitivos com valores nulos, delegações, expressões lambda e acesso direto à memória.

- Unsafe mode: ponteiros e fórmulas aritméticas utilizados sem verificação de integridade só podem ser utilizados na modalidade de modo inseguro;
  - Overflow: acessos a objetos são realizados através de referências seguras, dessa forma não podem ser invalidadas e normalmente as operações aritméticas são checadas contra sobrecarga;
  - Garbage collector: os objetos não são liberados da memória explicitamente, e sim através de um processo que elimina objetos quando não há referências;
  - Disposable: Destrutores não existem, mas utilizada junto com a construção using block, torna possível que recursos alocados por um objeto sejam liberados;
  - C# é mais seguro com tipos: as conversões implícitas por default são conversões seguras, tais como ampliação de inteiros e conversões de um tipo derivado para um tipo base; não existem conversões implícitas entre inteiros e variáveis lógicas ou enumerações; não existem ponteiros nulos; todas as conversões implícitas definida pelo usuário deve ser marcada explicitamente;
  - Namespace: membros de enumeração são colocados em seu próprio espaço de nome;
  - Reflexão: recursos de reflexão complexos estão disponíveis.
- 
- Suporta conceitos típicos de orientação a objetos, como encapsulamento, herança e polimorfismo;
  - Todas as variáveis, métodos e o ponto de execução de uma aplicação, são encapsuladas em definições de classes;
  - Uma classe derivada pode herdar apenas uma classe pai, e herdar quantas interfaces forem necessárias;

- Os métodos da classe derivada que substituem os métodos virtuais de uma classe pai exigem a utilização da palavra reservada: `override` para evitar a redefinição acidental de um método;
- Uma `struct` tem a mesma funcionalidade de classe, com a diferença, de ser mais simplificada; implementam interfaces, mas não suporta herança;
- Suporte à implementação de métodos e tipos genéricos para garantir maior segurança de tipo e de desempenho à aplicação;
- Os iteradores permitem a implementação de coleções de classes para definir comportamentos de iteração personalizados;
- As expressões LINQ (Consultas Integradas à Linguagem) são responsáveis em realizar consultas com base na integração dos recursos de consultas da própria linguagem.

### **3.Possível Desenvolver**

Por ser multiplataforma, o C# permite o desenvolvimento web, mobile e desktop, indo desde projetos simples até a criação de games completos e sistemas mais complexos, aplicativos cliente-servidor, webXML, componentes distribuídos ou aplicativos de banco de dados.

Esta linguagem é muito utilizada pelas maiores empresas, como é o caso da Microsoft, Amazon, StackOverflow, Dell, Intel, Honda, entre outras gigantes, principalmente devido à maturidade da linguagem e sua estabilidade.

### **4.Funcionalidades**

Com ele, é possível criar, por exemplo:

- Aplicativos cliente-servidor;
- Serviços web XML;
- Componentes distribuídos;
- Aplicativos de banco de dados.

Alguns recursos do C-Sharp auxiliam na construção de aplicativos duráveis e robustos, como:

1. Coleta de lixo: Recupera a memória ocupada por arquivos não-utilizados e inacessíveis de forma automática;
2. Tratamento de exceção: Fornece uma abordagem extensiva e estruturada, a fim de detectar e recuperar erros;



3. Design tipado: Impossibilita a realização da leitura de variáveis não-inicializadas e a indexação de matrizes além dos seus limites. Evita, ainda, que sejam executadas conversões de tipos não-verificadas.

Ele conta com outros recursos muito úteis para os desenvolvedores, incluindo:

- Tipo de valor anulável;
- Enumerações;
- Delegates;
- Expressões lambda;
- Acesso direto à memória.

## 5. Como desenvolver com a linguagem

Assim como o Java, o C-Sharp utiliza o conceito de máquina virtual. Trata-se do CLR (Common Language Runtime), que é uma espécie de computador virtual que gerencia a execução de programas que utilizam a plataforma “.net”.

Como o CLR é um ambiente de execução de todas as linguagens da plataforma, ele executa o código em uma linguagem intermediária, conhecido como CIL (Common Intermediate Language). Desta forma, oferece mais segurança e desempenho para o processo.

Veja um exemplo para entender melhor esse funcionamento:

- Um código gerado pelo C-Sharp é transmitido por um compilador de linguagem que o transforma em uma linguagem intermediária;
- Em seguida, o CIL captura esse código e, utilizando o compilador JIT (Just in Time), o converte em código de máquina;
- Somente após isso é que o código é efetivamente executado.

Apesar de, aparentemente, transparecer ser um processo grande, burocrático e demorado, o desempenho da execução do código escrito em C-Sharp é excelente, além de seguro e eficiente.

A ideia mais utilizada para desenvolvimento em C-Sharp é o [Visual Studio](#). Para executar uma aplicação em C-Sharp, é necessária uma máquina virtual de linguagem, além das bibliotecas do “.net”.

Mas fique tranquilo que tudo isso já vem junto com a instalação do Visual Studio.

Ele inclui, por exemplo:

- Compiladores;
- Intelligence;
- Designers gráficos.

É possível utilizar o Visual Studio em versões gratuitas, que são muito úteis para quem está começando e aprendendo sobre suas funcionalidades, ou nas suas edições pagas e mais completas – ideais para profissionais e empresas.

## 6. Utilização

Diversas empresas utilizam o C-Sharp para colocar em prática os seus projetos.

Confira alguns exemplos:

- Stackoverflow;
- Dell;
- Siemens;
- Microsoft;
- Intel;
- Asus;
- Honda;
- Infraero;
- Outback.

Além de ser interessante para quem é apaixonado pelo ecossistema da Microsoft, Windows e afins, essa linguagem tem atraído cada vez mais empresas envolvidas em desenvolvimento de games.

Inclusive, ele já é a linguagem-base utilizada por uma das maiores plataformas de criação de jogos, o Unity.

## 7.Vantagens do C#

- Integração com Windows;
- eficiente no gerenciamento do sistema;
- custo menor e mais seguro em relação a outras linguagens;
- sistema de cache viável e simples;
- grande comunidade online;
- código compilado em linguagem padrão, independentemente do Sistema Operacional e arquitetura de destino.

## **8.Desvantagens do C#**

- linguagem menos flexível;
- execução lenta;
- programas e componentes antigos precisam ser reescritos;
- necessidade de compilar a cada vez que o código for alterado.

## **9.Conclusão**

C# é uma linguagem simples, moderna, com características para o desenvolvimento de software que será aplicável ainda durante muitos anos. A linguagem é de fácil aprendizado, com um mercado competitivo, tornando-a uma boa sugestão para quem está em dúvida sobre qual linguagem de programação escolher.

## Referências

- «Dart 2.17: Productivity and integration». *medium.com* (em inglês). 11 de maio de 2022.  
  
Consultado em 11 de maio de 2022
- CISL GOVERNO FEDERAL. Palestra Técnica do CISL: Linguagem de Programação  
  
OpenSource DART.
- DART. A tour of the Dart language. Disponível em:  
  
<https://dart.dev/guides/language/language-tour>. Acesso em: 4 ago. 2019.
- BURROWS, Chris (2010). New C# Features in the .NET Framework 4.
- kunal-chowdhury (2012). Evolution of C# (1.0 - 5.0) - What are the New  
  
Features Introduced in C#
- LIMA, E. (2002). C# e .NET - Guia do Desenvolvedor. Editora Campus, p. 19.
- MSDN (2005). O que há de novo na linguagem e no compilador C# 2.0
- MSDN (2007). Overview of C# 3.0.
- MSDN (2012). Introdução à linguagem C# e o .NET Framework
- OFICINA DA NET. C# (C Sharp) o que é está linguagem?

