

EMBARCADERO CONFERENCE



embarcadero®

Vantages do uso de Serverless Computing

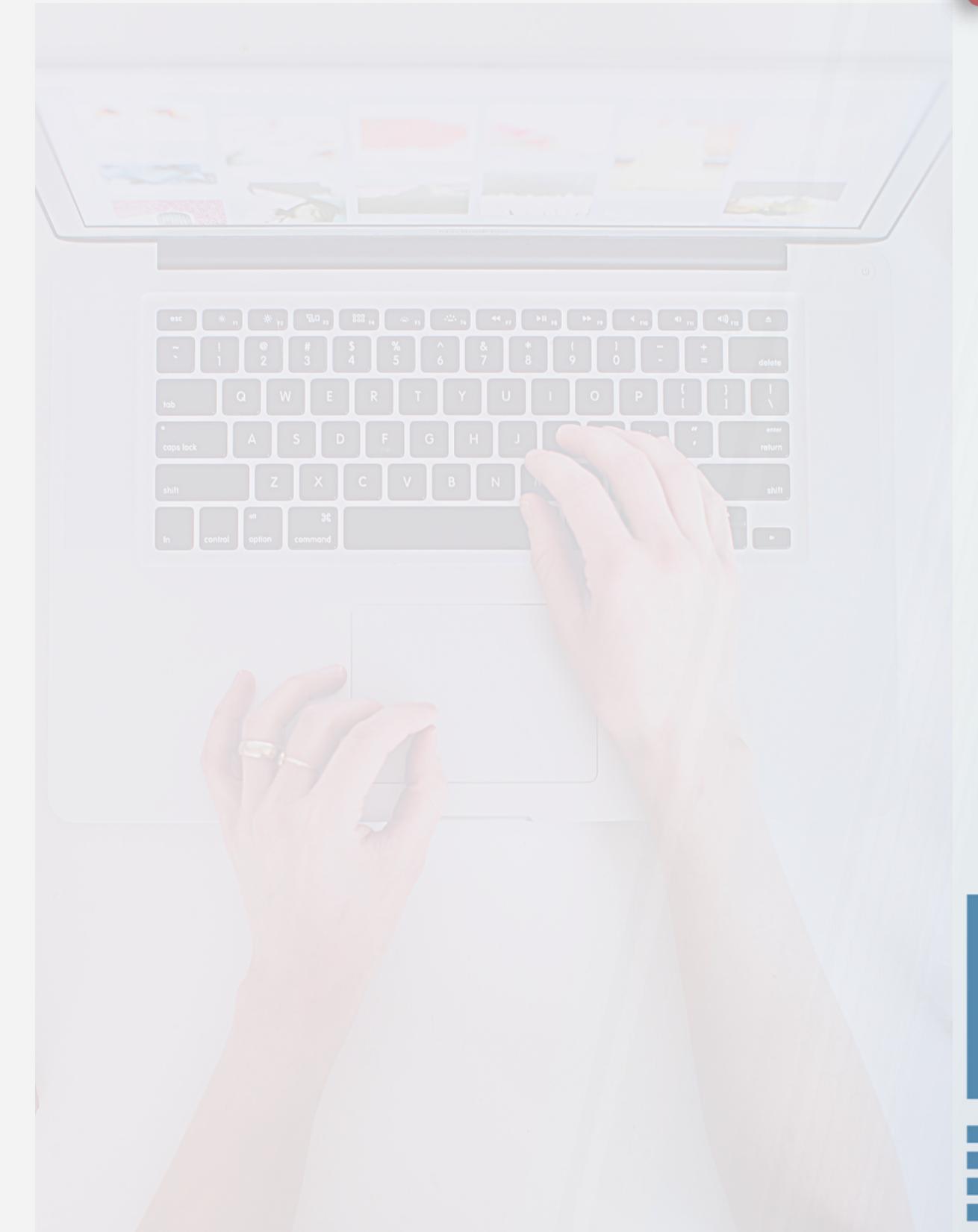


CONFERENCE
EMBARCADERO CONFERENCE 2022



Agenda

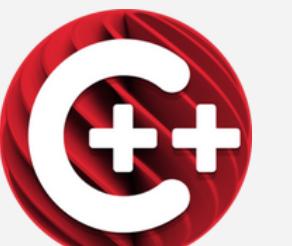
- O que é Serverless?
- Como implementar backend!?
- Como implementar frontend!?
- Vantagens de uso de Serverless
- Diferenças entre provedores





Dion Carlos Mai

- Gerente de Desenvolvimento Aquasoft
- MVP Delphi & C++Builder
- Certificado Master & Instrutor de Delphi
- Engenheiro Controle e Automação





O que é Serverless?



FAAS APLICADO

Serverless é aplicado por meio de *FaaS - Funcion as a Service*; porém, com o server "*Event Triggered*".

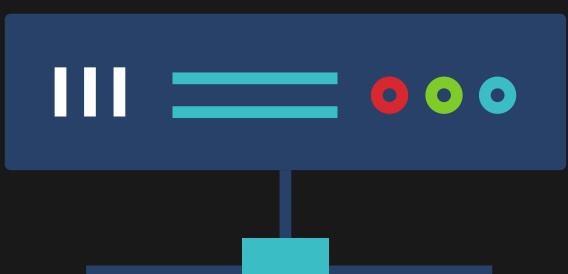
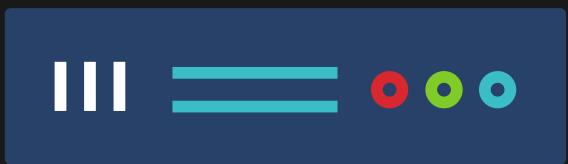
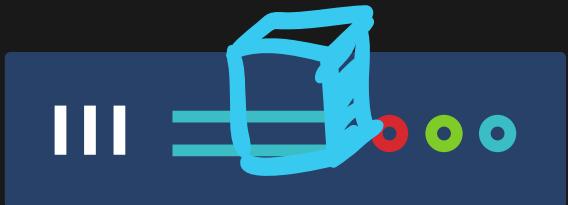


O que é Serverless?



SERVERLESS TEM SERVER?!

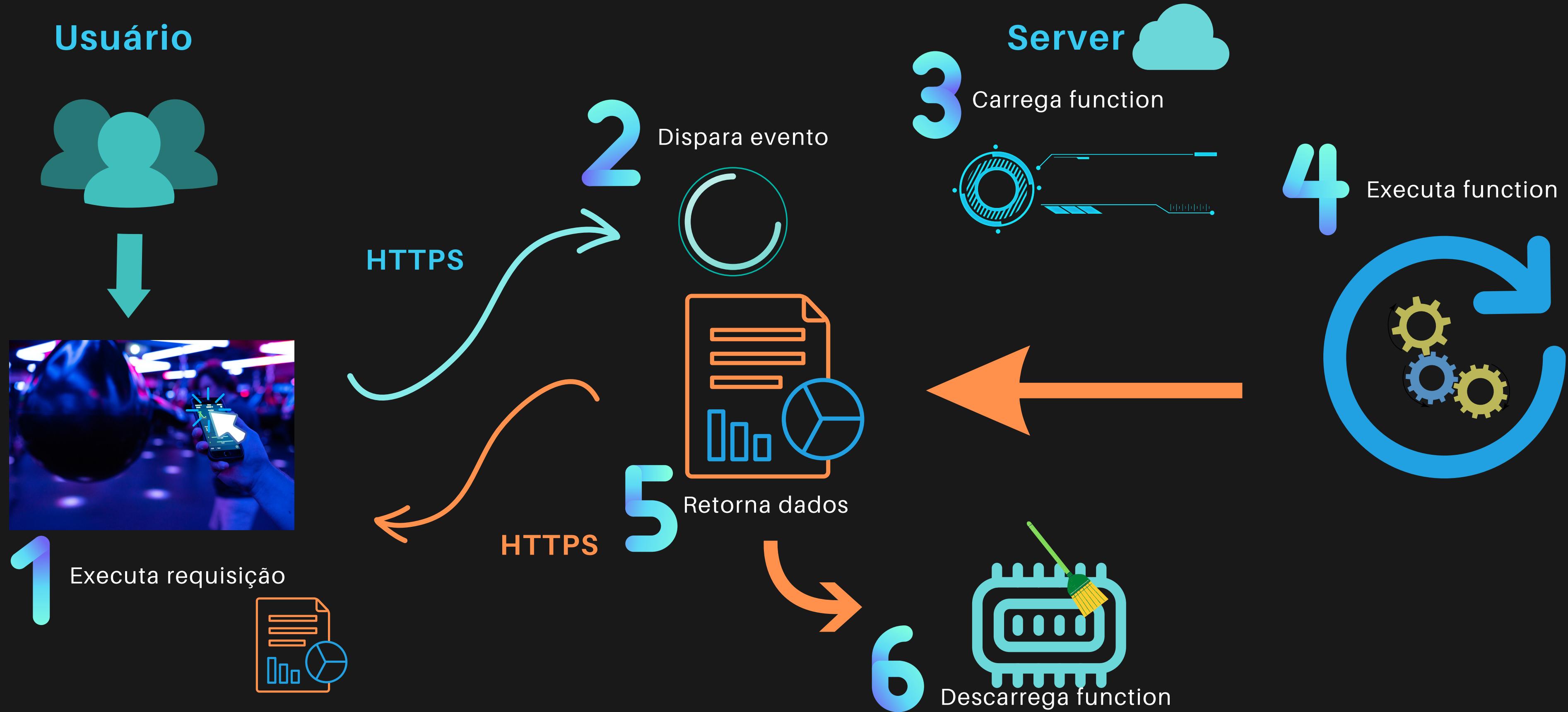
Será que erraram o nome?



Serverless - funcionamento



Entendendo o Serverless



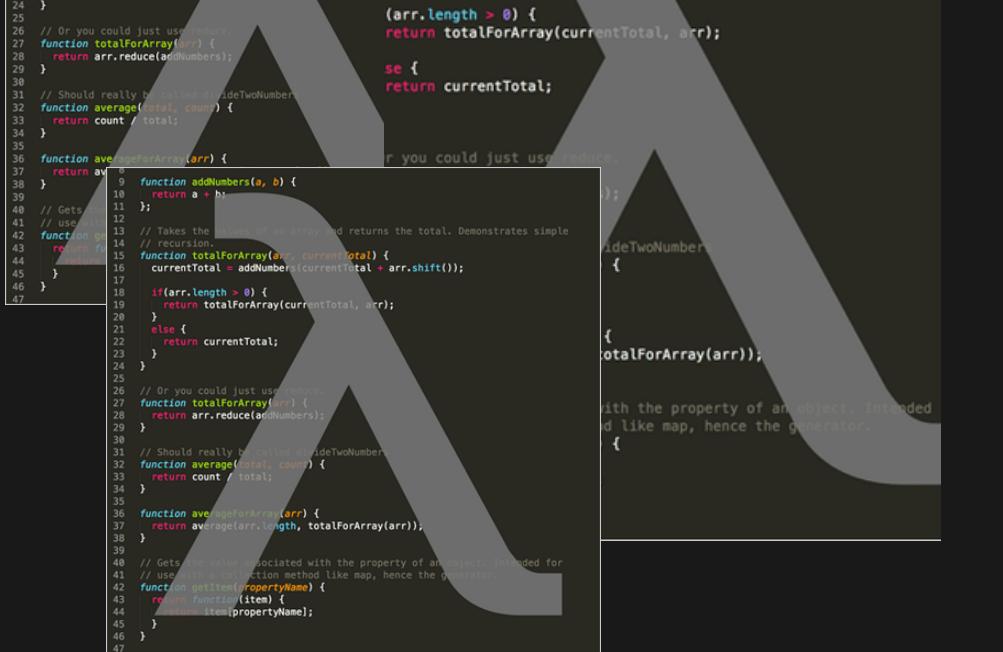


Entendendo Serverless?

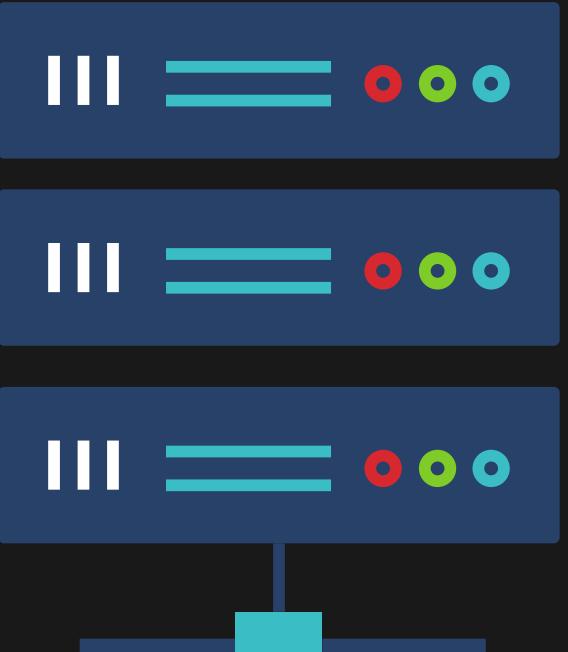


ENVIAMOS FUNÇÕES ISOLADAS PARA A CLOUD

ERP com "Infinitas" funções



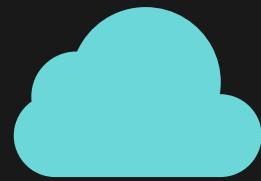
```
9 function addNumbers(a, b) {  
10    return a + b;  
11};  
12  
13 // Takes the values of an array and returns the total. Demonstrates simple recursion.  
14 function totalForArray(arr, currentTotal) {  
15    currentTotal = addNumbers(currentTotal + arr.shift());  
16  
17    if(arr.length > 0) {  
18        return totalForArray(arr, currentTotal);  
19    } else {  
20        return currentTotal;  
21    }  
22}  
23  
24 // Or you could just use reduce.  
25 function totalForArray(arr) {  
26    return arr.reduce(addNumbers);  
27}  
28  
29 // Should really be called divideTwoNumbers.  
30 function average(total, count) {  
31    return count / total;  
32}  
33  
34  
35 function avePropertyForArr(arr) {  
36    let sum = 0;  
37    let count = 0;  
38    arr.forEach(item => {  
39        sum += item.propertyName;  
40        count++;  
41    });  
42    return sum / count;  
43}  
44  
45 }  
46  
47
```



DDD(Domain Driven Design)
Um modelo que pode auxiliar



Serverless - implementando o backend



PASSOS PARA CONSTRUÇÃO

- Selecionar provedor: Azure, AWS, GoogleCloud, IBM Cloud, AlibabaCloud
- Criar conta no provedor
- Selecionar tecnologia do backend
 - Usar linguagem nativa
 - Usar containers, miniVMs para executar
- Implementar função
- Criar rotina de deploy
- Publicar função
- Demais configurações
 - Limite de execuções
 - Regiões de acesso
 - Segurança
 - Logs



Serverless - Exemplo Azure



MANUAL ONLINE

Microsoft tem um manual online atualizado:

<https://learn.microsoft.com/en-us/azure/azure-functions/create-first-function-vs-code-other?tabs=go%2Cwindows>



The screenshot shows a Microsoft Learn page for Azure Functions. At the top, there's a navigation bar with links for Microsoft Learn, Documentation, Training, Certifications, Q&A, Code Samples, Shows, and Events. Below that is a secondary navigation bar for Azure with links for Product documentation, Architecture, Learn Azure, Develop, and Resources. On the left, there's a sidebar with a 'Filter by title' search bar and a list of documentation sections: Functions Documentation, Overview, Quickstarts (with 'Create your first function' and 'Visual Studio Code' options), and a 'Learn' section. The main content area has a breadcrumb trail 'Learn / Azure / Functions /' and a set of edit and share icons. The main title of the page is 'Quickstart: Create a Go or Rust function in Azure using Visual Studio Code'.



Serverless - Exemplo Azure



1 - CRIAR CONTA

- 30 dias de crédito para testes
- Alguns recursos *always free* com base em consumo
- Informar cartão de Crédito

<https://portal.azure.com/>

The screenshot shows the Microsoft Azure portal interface. At the top, there's a browser-like header with the URL <https://portal.azure.com/#home>. Below it is a blue navigation bar with the Microsoft Azure logo and a search bar that says "Search resources, services, and docs (G+/)". The main content area is titled "Azure services" and features several icons with corresponding links: a plus sign for "Create a resource", a lightning bolt for "Function App", a key for "Subscriptions", a rocket ship for "Quickstart Center", a computer monitor for "Virtual machines", and a network icon for "App Services".



Serverless - Exemplo Azure



1 - CRIAR CONTA - DETALHES

- 1 milhão de execs mês gratuitas
- Mesmo para 5 milhões de exec, preço de R\$ 5,00
- Máquina mínima de config custa R\$ 800,00

<https://azure.microsoft.com/en-us/pricing/calculator/>



Serverless - Exemplo Azure



1 - CRIAR CONTA - DETALHES

The screenshot shows the Azure Pricing calculator interface for a serverless plan. The main summary table indicates:

Requests	= R\$4.29
5,000,000 Execution count	
Upfront cost	R\$0.00
Monthly cost	R\$4.29

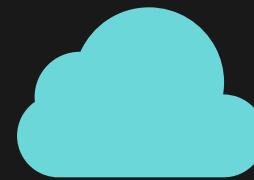
Below the summary, there are sections for Support and Licensing. Under Support, it says "Included". Under Licensing Program, it says "Microsoft Customer Agreement (MCA)". There is also a checkbox for "SHOW DEV/TEST PRICING".

At the bottom, there are sections for "Estimated upfront cost" (R\$0.00) and "Estimated monthly cost" (R\$4.29). A large blue arrow points from the monthly cost section towards a currency conversion table.

At the very bottom, there are buttons for "Export", "Save", and "Share". On the right side, there is a "CURRENCY" dropdown set to "Brazil – Real (R\$) BRL" with a note "1 USD = 5.36 BRL".



Serverless - Exemplo Azure



1 - CRIAR CONTA - DETALHES

Virtual Machines 1 D2 v3 (2 vCPUs, 8 GB RAM) x 730 Hours (Pay as you go) Upfront: R\$0.00 Monthly: R\$983.25

REGION: Brazil South OPERATING SYSTEM: Windows TYPE: (OS Only) TIER: Standard

CATEGORY: All INSTANCE SERIES: All INSTANCE: D2 v3: 2 vCPUs, 8 GB RAM, 50 GB Temporary storage, R\$1.347/hour

Virtual machines 1 730 Hours

Savings Options

Explore pricing models to help optimize your Azure costs. Learn more

Compute (D2 v3)

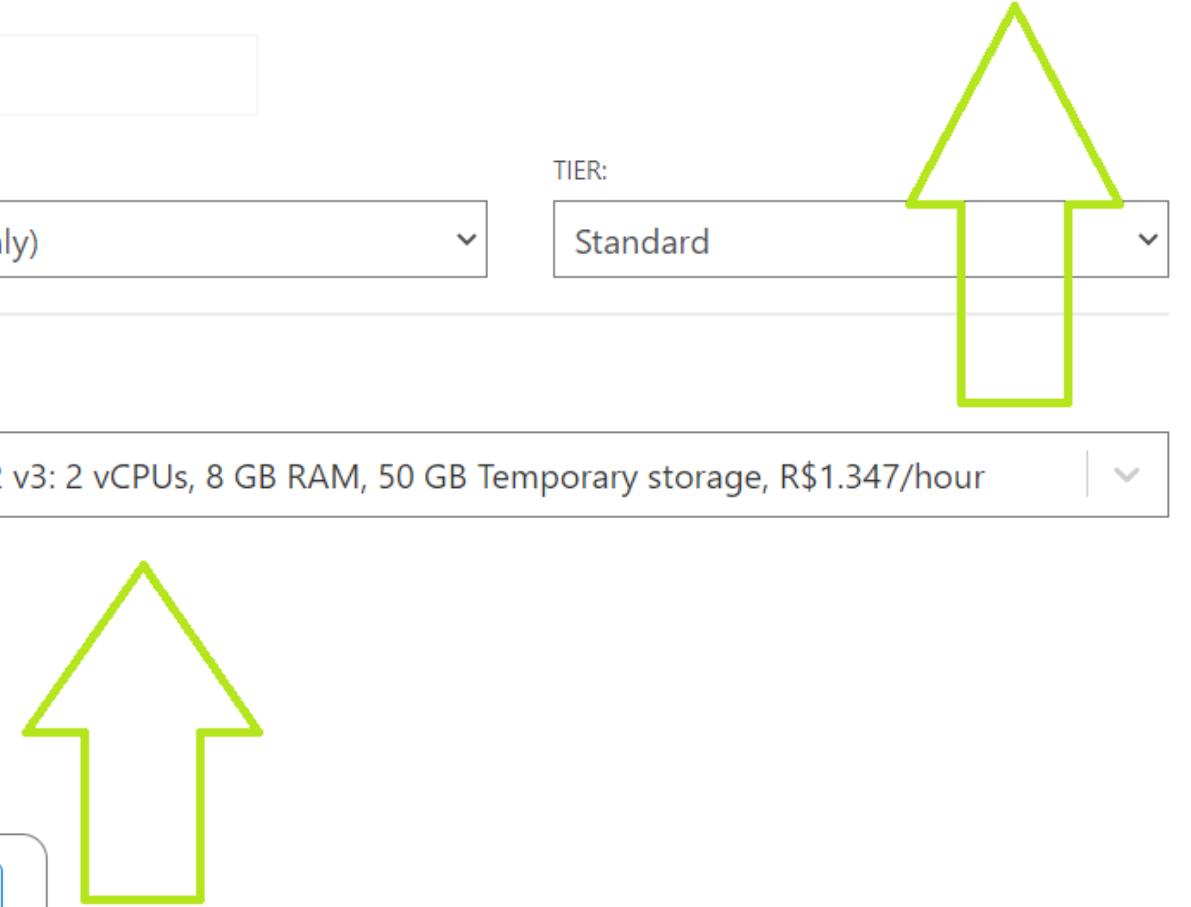
Pay as you go

1 year savings plan (~13% discount)

OS (Windows)

License included

Azure Hybrid Benefit



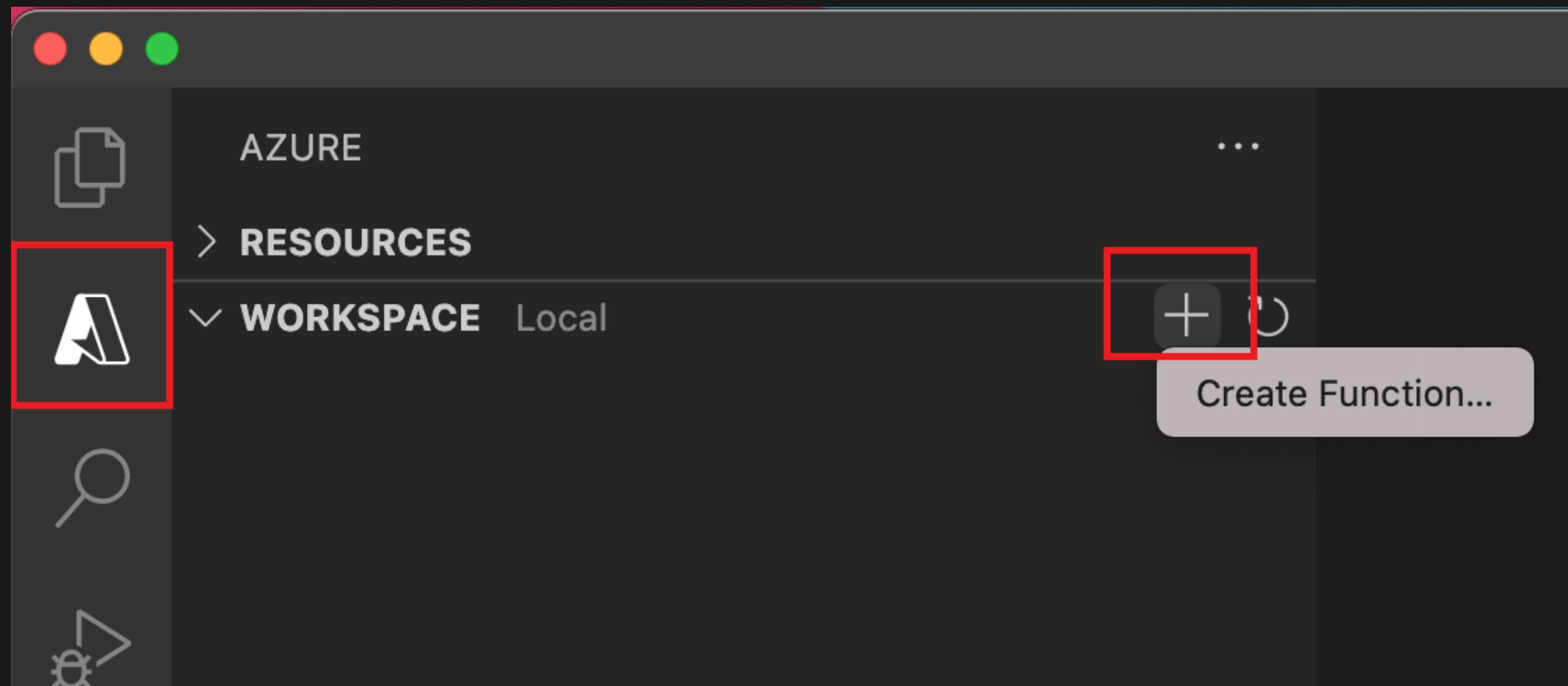


Serverless - Exemplo Azure



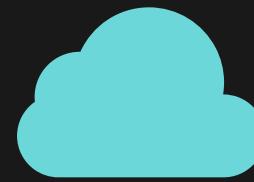
2 - CONFIGURAR AMBIENTE DEV

- Instalar Visual Studio Code/Visual Studio
- Instalar Azure Functions Core Tool
- Iniciar criação da função localmente





Serverless - Exemplo Azure



3 - DEFINIÇÃO TIPO FUNÇÃO

Funções podem ser implementadas em linguagens nativas ou publicadas no formato de executáveis ou containers.

Diferenças principais:

- Pequeno delay no momento da execução
- Possível custo computacional mais elevado devido a carga de recursos adicionais para execução da função

Função para execução genérica

Prompt	Selection
Select a language for your function project	Choose Custom Handler.

Função linguagem nativa

Prompt	Selection
Select a language	Choose C#.
Select a .NET runtime	Select .NET 6.



Serverless - Exemplo Azure



4 - IMPLEMENTAR/CONFIGURAR

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORADOR**: Shows the project structure with files like host.json, function.json, .vscode, and .gitignore.
- EDITORES ABERTOS**: Shows two open files: host.json and function.json under the HttpExample folder.
- function.json Content (highlighted in yellow box)**:

```
1  {
2   "bindings": [
3     {
4       "authLevel": "anonymous",
5       "type": "httpTrigger",
6       "direction": "in",
7       "name": "req",
8       "methods": [
9         "post"
10      ]
11    },
12    {
13      "type": "http",
14      "direction": "out",
15      "name": "res"
16    }
17  ]
18 }
```



Serverless - Exemplo Azure

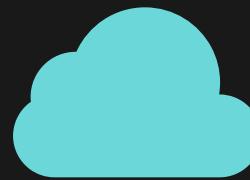


4 - IMPLEMENTAR/CONFIGURAR - EXE

```
{ host.json } host.json U X
{
  "version": "2.0",
  "logging": {
    "applicationInsights": {
      "samplingSettings": {
        "isEnabled": true,
        "excludedTypes": "Request"
      }
    }
  },
  "extensionBundle": {
    "id": "Microsoft.Azure.Functions.ExtensionBundle",
    "version": "[3.*, 4.0.0)"
  },
  "customHandler": {
    "enableForwardingHttpRequest": true,
    "description": {
      "defaultExecutablePath": "functionDelphiAzure.exe",
      "workingDirectory": "",
      "arguments": []
    }
  }
}
```



Serverless - Exemplo Azure



4 - IMPLEMENTAR/CONFIGURAR - EXE

```
procedure RunServer(APort: Integer);
var
  LServer: TIIdHTTPWebBrokerBridge;
  LResponse: string;
begin
  LServer := TIIdHTTPWebBrokerBridge.Create(nil);
  try
    var portEnvVarStr: string := GetEnvironmentVariable('FUNCTIONS_CUSTOMHANDLER_PORT');
    var portEnvVar: Integer;
    if TryStrToInt(portEnvVarStr, portEnvVar) then
      begin
        LServer.DefaultPort := portEnvVar;
      end else begin
        LServer.DefaultPort := APort;
      end;

    StartServer(LServer);
    while True do
      begin
        // ...
      end;

    finally
      StopServer(LServer);
      LServer.Free;
    end;
  end;
```



Serverless - Exemplo Azure



4 - IMPLEMENTAR/CONFIGURAR - NATIVO

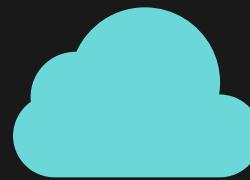
The screenshot shows the Microsoft Visual Studio interface with the following components:

- Code Editor:** Displays the `FunctionNovidadesERP.cs` file. The code implements an Azure Function named `FunctionNovidadesERP` that handles HTTP requests. It includes logic to validate an app key and return a JSON response.
- Solution Explorer:** Shows the project structure for "NovidadesERP". It includes files like `host.json` and `local.settings.json`, along with `FunctionNovidadesERP.cs`.
- Properties Window:** Shows the properties for the selected file.
- Task List:** Shows a message: "Não foi encontrado nenhum problema" (No problems found).
- Status Bar:** Shows the current line (Ln: 21), column (Car: 6), and encoding (SPC, CRLF).

```
18 public static class FunctionNovidadesERP
19 {
20     private static readonly Dictionary<string, string> atualizacoes = new Dictionary<string, string>
21     {
22         { "Relatórios", "Adicionados múltiplos filtros" },
23         { "Dashboards", "Consolidação dos totalizadores filtrados" },
24         { "Boletos", "Funcionalidade de pagamento por PIX" }
25     };
26
27     1 referência
28     public static bool validaLicenca(string key)
29     {
30         return key.IndexOf("96cd") > 0;
31     }
32
33     [FunctionName("NovidadesERP")]
34     0 referências
35     public static async Task<IActionResult> Run(
36         [HttpTrigger(AuthorizationLevel.Function, "post", Route = null)] HttpRequest req,
37         ILogger log)
38     {
39         log.LogInformation("Request - HTTP");
40
41         string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
42         dynamic data = JsonConvert.DeserializeObject(requestBody);
43         string appKey = req.Headers["app-key"];
44         appKey ??= data?.appKey;
45
46         string responseMessage = string.IsNullOrEmpty(appKey)
47             ? "Sistema/Licença não localizados"
48             : "1 nova atualização disponível";
49
50         var objResp = new
51             sistema = "Controle de Updates by Azure",
52             licencaAtiva = validaLicenca(appKey),
53             status = responseMessage,
54             atualizacoes = atualizacoes
55         };
56         return new JsonResult(objResp);
57     }
58 }
```



Serverless - Exemplo Azure



5 - EXECUTAR LOCAL

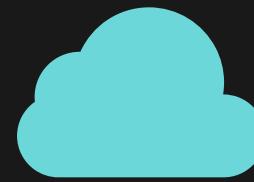
```
PS C:\Users\...> func start --verbose
    %%%%%%
    %%%%
    @  %%%%  @
    @@ %%%%  @@
    @@@ %%%%%%% @@@
    @@ %%%%%%% @@@
    @@ %%%  @@
    @@ %%  @@
    @@ %  @@
    %

Azure Functions Core Tools
Core Tools Version:      4.0.4785 Commit hash: N/A (64-bit)
Function Runtime Version: 4.10.4.19213

[2022-10-24T17:55:20.255Z] Building host: startup suppressed: 'False', configuration suppressed: 'False', startup operation id
30f089'
[2022-10-24T17:55:20.263Z] Reading host configuration file 'C:\Users\...'
[2022-10-24T17:55:20.266Z] Host configuration file read:
```



Serverless - Exemplo Azure



5 - EXECUTAR LOCAL

Functions:

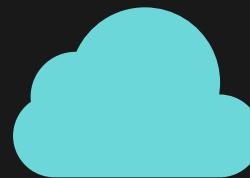
```
HttpExample: [POST] http://localhost:7071/api/HttpExample
```

```
[2022-10-24T17:55:22.345Z] Worker process started and initialized.
```

```
[2022-10-24T17:55:26.554Z] Host lock lease acquired by instance ID '000000000000000000000000442FC398'.
```



Serverless - Exemplo Azure



6 - CONSUMIR - CLIENT

The screenshot shows the Postman application interface. At the top, the URL `http://localhost:7071/api/HttpExample` is entered, and the method is set to `POST`. Below the URL, the `Headers (8)` tab is selected, showing a table with one row: `app-key` (Key) and a redacted value (Value). The `Body` tab is also visible. In the bottom section, the `Body` tab is selected, displaying a JSON response with the following content:

```
1  {
2      "atualizacoes": {
3          "boletos": "Funcionalidade de pagamento por PIX",
4          "dashboards": "Consolidação dos totalizadores filtrados",
5          "relatórios": "Adicionados múltiplos filtros"
6      },
7      "licençaAtiva": true,
8      "sistema": "Controle de Updates by Azure + Delphi",
9      "status": "1 nova atualização disponível"
10 }
```



Serverless - Exemplo Azure



6 - CONSUMIR - CLIENT

Object Inspector Data.TestCloudFunctions X

RESTClientAzure TRESTClient

Properties Events

Accept	
AcceptCharset	
AcceptEncoding	
AllowCookies	<input checked="" type="checkbox"/> True
Authenticator	
AutoCreateParams	<input checked="" type="checkbox"/> True
BaseUrl	http://localhost:7031/api/Novidade
> BindSource	RESTClientAzure.BindSource
ConnectTimeout	30000
ContentType	
FallbackCharsetEncoder	utf-8
HandleRedirects	<input checked="" type="checkbox"/> True
> LiveBindings Designer	LiveBindings Designer
Name	RESTClientAzure
Params	(TRESTRequestParameterList)
ProxyPassword	
ProxyPort	0
ProxyServer	
ProxyUsername	
RaiseExceptionOn500	<input checked="" type="checkbox"/> True
ReadTimeout	30000
> RedirectsWithGET	[Post301,Post302,Post303,Put303,Delete303]
> SecureProtocols	[]

Diagram:

```
graph TD; RESTClientAzure[RESTClientAzure] --> RESTResponseAzure[RESTResponseAzure]; RESTClientAzure --> TRequestAzure[TRequestAzure]; RESTClientAzure --> RESTClientAzure[RESTClientAzure]
```



Serverless - Exemplo Azure

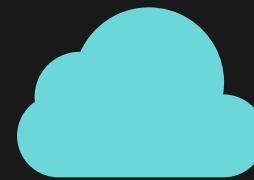


6 - CONSUMIR - CLIENT

```
• procedure TdmdTestCloudFunctions.GetAzureVersionUpdates;
• begin
5   RESTRequestAzure.Execute;
•   var jsonContent: string := RESTResponseAzure.Content;
•
•   if jsonContent <> '' then
•     begin
0       end else begin
•         dmdNotifications.Send('Sem resposta!', 'Não foi possível verificar novidades da versão');
•       end;
•     end;
•   end;
```

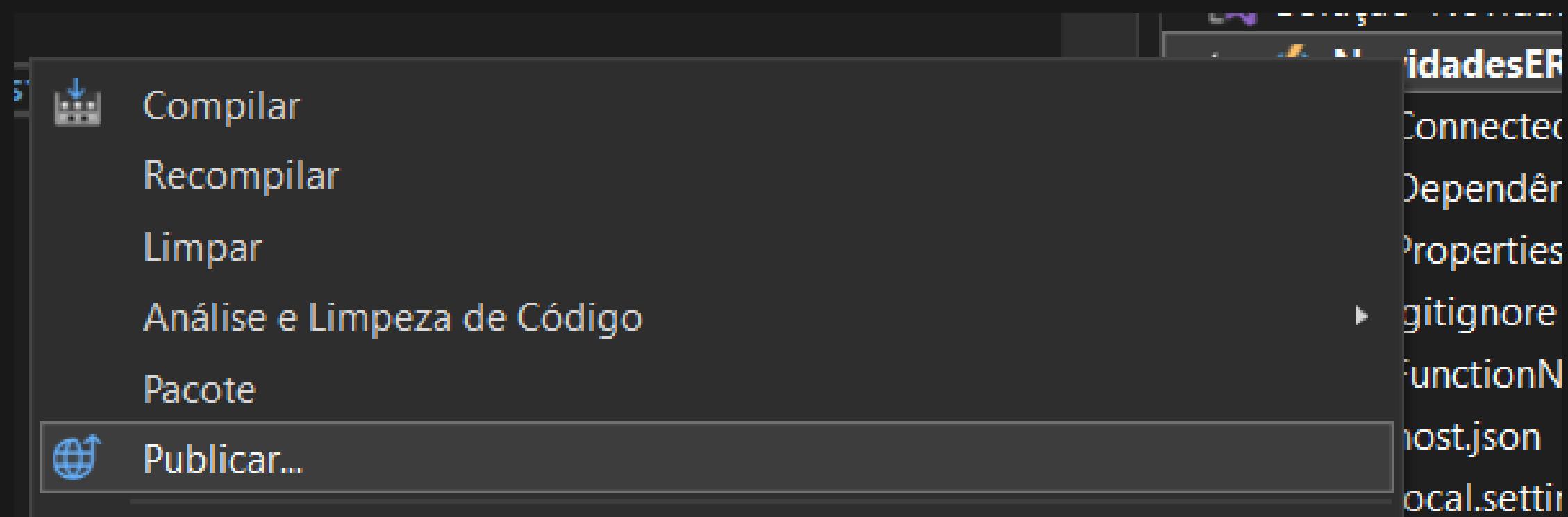


Serverless - Exemplo Azure



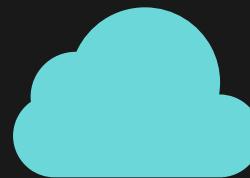
7 - DEPLOY

- Publica a função
- Disponibiliza para acesso imediato a todos os clients
- Feita via ferramentas da MS





Serverless - Exemplo Azure



7 - DEPLOY

The screenshot shows the Azure portal interface for deploying an Azure Function app. At the top, there's a navigation bar with a cloud icon, the title 'AzureOnTheFly - Zip Deploy.pubxml', and a dropdown for 'Aplicativo de Funções do Azure (Windows)'. Below this is a large green success message box containing a checkmark and the text 'Publicação com êxito em 05/10/2022 às 17:12:14.' followed by a link 'Abrir o site'. To the right of this message is a large blue arrow pointing right, with a rounded rectangle around the 'Publicar' button. Below the message box are sections for 'Configurações' (with 'Configuração' and 'Tempo de Execução de Destino' options), 'Hospedagem' (with 'Assinatura', 'Grupo de recursos' set to 'NovidadesERP20221006115124ResourceGroup', 'Nome do recurso' set to 'AzureOnTheFly', 'Nome de usuário' (redacted), and 'Senha' (redacted)), and a 'Site' URL at the bottom: <https://azureonthefly.azurewebsites.net>.



Serverless - Exemplo Azure



8 - MONITORAMENTO

Home > AzureOnTheFly > AzureOnTheFly - Application Insights >

AzureOnTheFly Application Insights

Search Application Dashboard Getting started Search Logs Monitor resource group Feedback Favorites Rename Delete

Overview

Resource group (move) : NovidadesERP20221006115124ResourceGroup
Location : Brazil South
Subscription (move) : Azure subscription 1
Subscription ID : 7a1adf12-55dc-450b-835f-123d27d44a70
Tags (edit) : Click here to add tags

Instrumentation Key : [REDACTED]
Connection String : [REDACTED]
Workspace : DefaultWorkspace-7a1adf12-99de-430b-893f-123d27d44a70

Show data for last: 30 minutes 1 hour 6 hours 12 hours 1 day 3 days 7 days 30 days

Failed requests

Failed requests (Count) azureonthefly

0

3:20 PM 3:25 PM 3:30 PM 3:35 PM 3:40 PM UTC-03:00 Oct 24

Server response time

Server response time (Avg) azureonthefly

8.04 ms

25ms
20ms
15ms
10ms
5ms
0ms

3:20 PM 3:25 PM 3:30 PM 3:35 PM 3:40 PM UTC-03:00 Oct 24

Server requests

Server requests (Count) azureonthefly

6

3
2
1
0

3:20 PM 3:25 PM 3:30 PM 3:35 PM 3:40 PM UTC-03:00 Oct 24

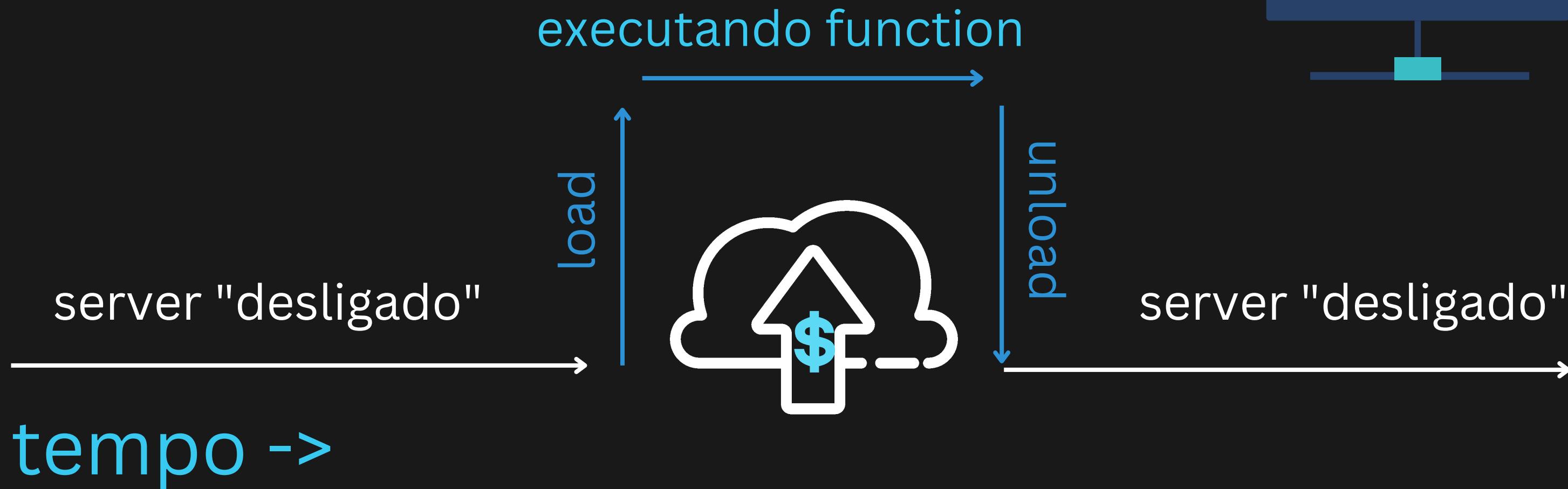
The screenshot displays the Azure Application Insights dashboard for the 'AzureOnTheFly' application. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Investigate (Application map, Smart detection, Live metrics, Transaction search, Availability, Failures, Performance, Troubleshooting guides (preview)), Monitoring, and Alerts. The main area shows essential details like Resource group, Location, Subscription, and Tags. Three charts are present: 'Failed requests' (0), 'Server response time' (8.04 ms), and 'Server requests' (6). A large teal arrow points from the top right towards the 'Server requests' chart.



Serverless - Recapitulando...

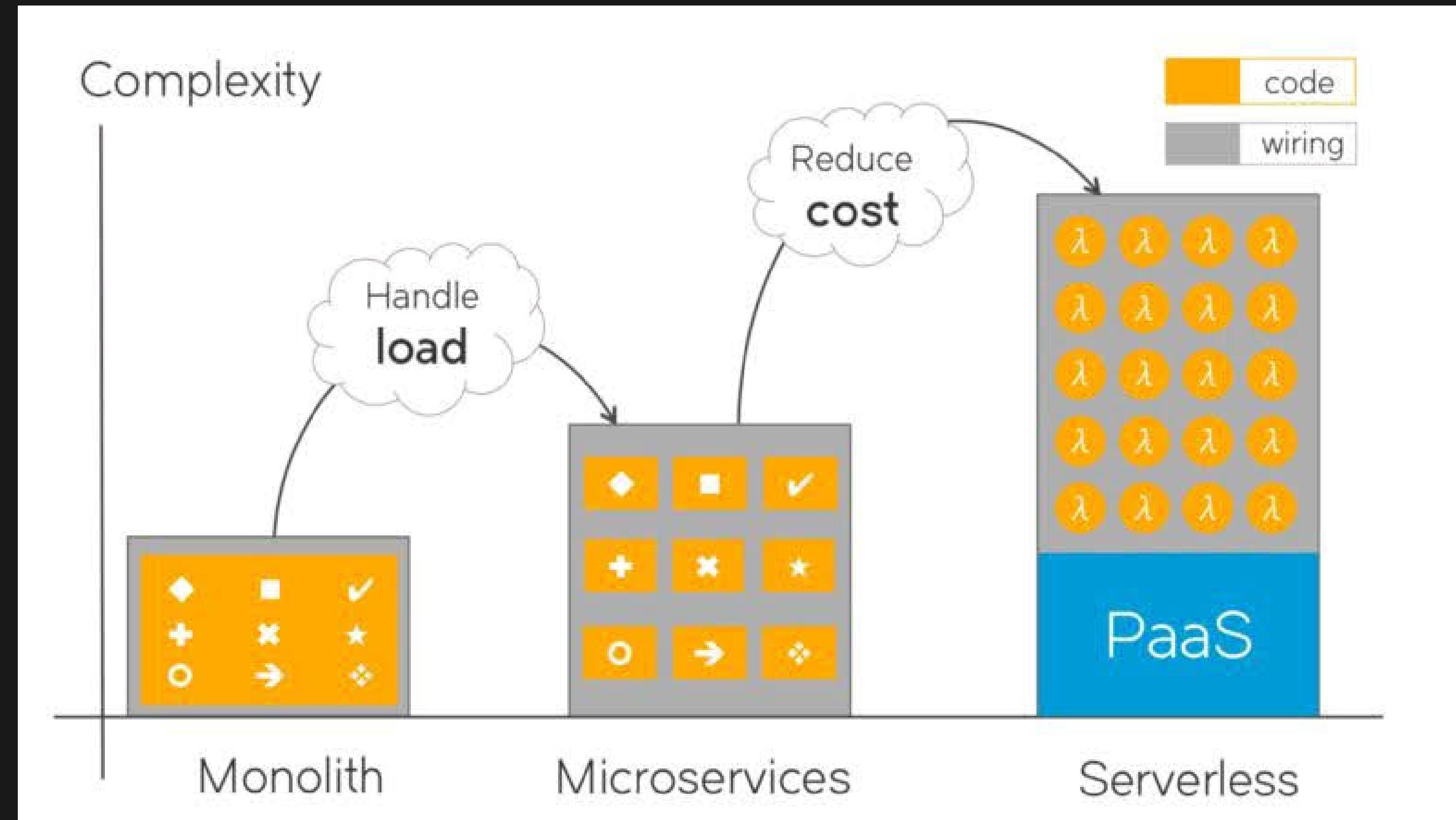


CICLO DE VIDA DO "SERVER"



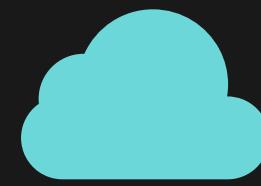


Serverless - Arquitetura





Serverless - Arquitetura



Questions / Challenges on that Journey

From Monolith



**Where to start?
What to break out?
Dependencies?**

To Services



**Works as expected?
Users happy?
Does it scale?
Does it perform?**

To Functions



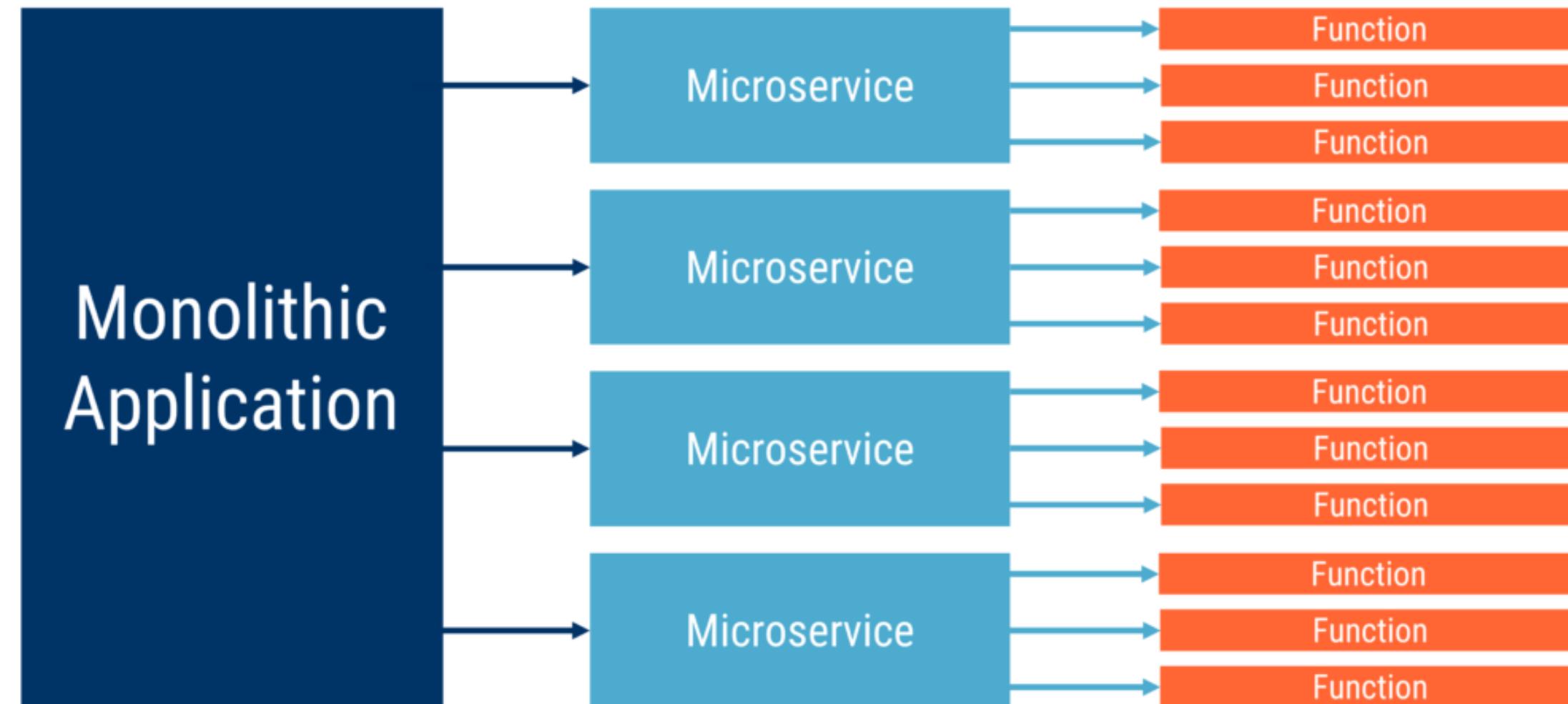
**Works as expected?
Users happy?
How to optimize?
How to automate?**



Serverless - Arquitetura

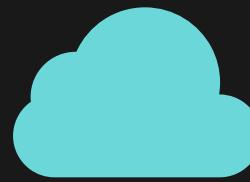


Functions provide further application granularity





Serverless - Vantagens do uso



VANTAGES

- Custo
- Velocidade de "go live"
- Ferramentas de controle/monitoramento
- Aumento da reutilização de código



Serverless - Desvantagens do uso

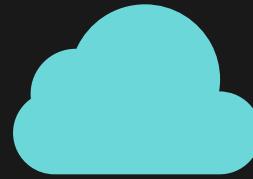


DESVANTAGES

- Aumento na complexidade
- Dependência de provedor de cloud
- Perda de acesso por ausência/atraso de pagamento
- Erros de lógica/hacks podem gerar grandes custos
- Troca de provedor de cloud pode demandar alterações de fonte



Serverless - Diferenças entre provedores

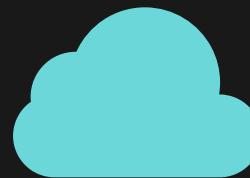


DIFERENÇAS

- Custo
- Linguagens nativas
- Ambiente de execução das funções
- Exigência de bibliotecas para acesso
- Quantidade de acessos gratuitos mensais
- Ferramentas adicionais



Serverless - Features relacionadas



OUTRAS FEATURES

- Mensageria
- Logs
- "Functions Apps"/"Logical Apps"
- Application Insight
- Monitors



Serverless Links



GUIAS

Guia Embarcadero

- <https://www.youtube.com/watch?v=NihVuXeW5Qo>

Custom Handlers

- <https://learn.microsoft.com/en-us/azure/azure-functions/create-first-function-vs-code-other?tabs=go%2Cwindows>

Funções nativas

- <https://learn.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-function-visual-studio?source=recommendations&tabs=in-process>



Serverless Links



TUDO DISPONÍVEL NO GITHUB





OBRIGADO

Bora usar
Serverless!