

Developer Admission Test

Career Sustainability Program

Version: v.1.08

Copyright

© 2022 Progress Rail, A Caterpillar Company

Caterpillar retains all copyrights in any text, graphic images, and software owned by Caterpillar and hereby authorizes you to electronically copy documents published herein solely for the purpose of transmitting or viewing the information. You may not mirror, modify or otherwise alter any files in this website for rebroadcast, or print the information contained therein, without written permission from Caterpillar. Except as expressly provided above, nothing contained herein shall be construed as conferring any license or right under any Caterpillar copyright, patent or trademark.

CAT, CATERPILLAR, LET’S DO THE WORK, their respective logos, "Caterpillar Corporate Yellow", the "Power Edge" and Cat "Modern Hex" trade dress as well as corporate and product identity used herein, are trademarks of Caterpillar and may not be used without permission.Cat and Caterpillar are registered trademarks of Caterpillar Inc., 100 N.E. Adams, Peoria IL 61629.

Contact

Rua Avelino Silvera Franco, 149

Bloco 1, Sala 426, 428

Campinas - SP

CEP 13105-822

Business Number: Direct: +55 19 998.500.273

Contact:

Dr. Rodrigo Gonçalves CTO - Nitro Solutions rodrigo@nitryx.com

Changelog

v1.08 2022-07-20 Minor fix

v1.07 2022-07-20 Review considering new literacies v1.0 2021-06-22 First final version v0.9 2021-06-20 Beta (complete) version for internal validation. v0.8 2021-06-18 Alpha (incomplete) version for internal validation.

Contents

[1 Instructions 3](#_Toc8109)

[2 Critical Thinking 4](#_Toc8110)

[2.1 Problem solving 4](#_Toc8111)

[2.1.1 Questions 5](#_Toc8112)

[3 System thinking 5](#_Toc8113)

[3.1 Software engineering 5](#_Toc8114)

[3.1.1 Questions: 6](#_Toc8115)

[3.1.2 Programming Task: 6](#_Toc8116)

[4 Technological Literacy 6](#_Toc8117)

[4.1 AI and Machine Learning 6](#_Toc8118)

[4.2 System simulation 6](#_Toc8119)

[4.3 Databases 6](#_Toc8120)

[4.4 Coding skills 7](#_Toc8121)

[4.4.1 Basic coding skills 7](#_Toc8122)

[4.4.2 Coding skills 7](#_Toc8123)

[4.4.3 Object oriented and Java programming 8](#_Toc8124)

[4.5 Multiprocessing and multi-threading 8](#_Toc8125)

[4.6 Virtualization, Containerization, Load Balancing 8](#_Toc8126)

[5 Data Literacy 9](#_Toc8127)

[5.1 Statistics I 9](#_Toc8128)

[5.2 Statistics II 9](#_Toc8129)

[6 Human Literacy 10](#_Toc8130)

[6.1 Theory of money 10](#_Toc8131)

[7 Cultural agility 10](#_Toc8132)

[7.1 Multi culture 10](#_Toc8133)

[8 Entrepreneurship 11](#_Toc8134)

[8.1 Innovation 11](#_Toc8135)

[9 Programming skills 11](#_Toc8136)

# Instructions

Many of our positions require applicants to take an admission test, and we use these tests to help us choose between all the excellent candidates who apply to work with us. To make your application as competitive as possible, you must submit your answer in less than 48h.

All information you need is in this exam. You are forbidden to ask for help from others, but you can search the internet or any literature. If you search the internet or the literature, you must include the references in your answers.

Important: How you respond is as important as what you answer. If you do not know something, say it. If you had to do some research and learn something to answer the question, let us know what your sources were and what you did learn from there.

You are also forbidden to share this test. This is proprietary and copyrighted material and is customized for each candidate. It can be traced back to you.

Please use UML notation whenever applicable.

# Critical Thinking

## Problem solving

A train meet is a situation in rail transit operations where a train traveling in one direction "meets" another traveling in the opposite direction. To avoid a collision, one train has to stop on a railroad siding for the other train to pass on a single-track mainline.

A train meet is depicted in Figure 1

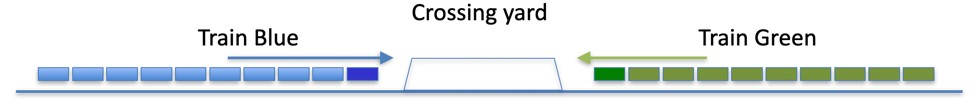


Figure 1: Train Blue meets Train Green

Usually, one train stops on the railroad siding while the other pass on the mainline, as depicted on Figure 2

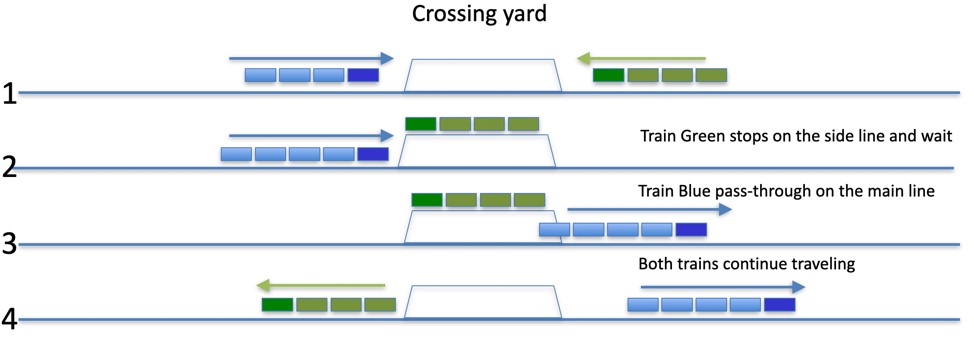


Figure 2: Solution when at least one train fits on the sideline of the crossing yard

Unfortunately, as shown in Figure 1, the Crossing Yard is too small. It can only hold 4 vehicles, either locomotives or railcars (both locos and railcars have the same size in this case).

Fortunately, as shown in Figure 3, it is possible to break the trains into smaller ones and leave railcars parked wherever you want. For example, we can move Train Blue to the other side of the Yard (step 1), invert its direction, and move backward using the sideline (step 2); break the train leaving 4 railcars on the sideline (step 3); move the train to its original position (step 4).

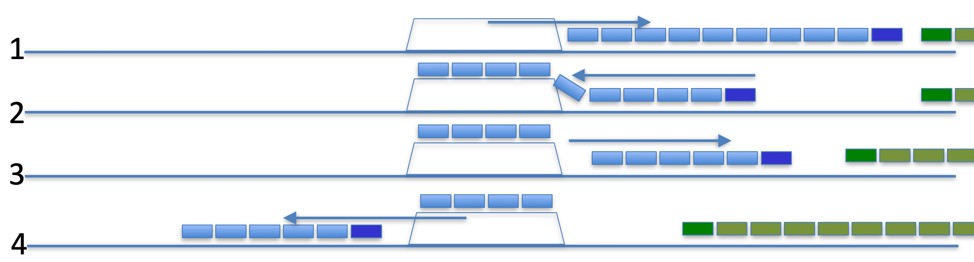


Figure 3: Breaking trains into smaller trains

### Questions

1. Is it possible to solve the meet’n’pass problem depicted in Figure 1? If yes, please draw your solution using PowerPoint or any other tool you want. You can even draw using paper and pencil and then scan or take a picture of your solution.

NÃO ENXERGUEI A SOLUÇÃO, UMA VEZ QUE CADA TREM TEM 10 vagões. se o pátio comporta somente 4 veículos, ficariam 6 veículos de fora, o que impossibilitaria a passagem.

como solução, deveria ter ao menos 10 vagas para que um dos trens entrassem totalmente no pátio.

1. Considering that the Yard can hold only four vehicles, is there any limit to the train length?

O TREM QUE IRÁ ENTRAR NO pátio DEVE TER no máximo 4 VEÍCULOS.

# System thinking

## Software engineering

The Taggart Transcontinental Railroad (TTR) wants to ensure safety (no collisions) and increase the performance of the operations. The Yard Manager will designate a Train Crossing Coordinator to help the Train Drivers perform the train movements for the train crossing problem described in Section 2.1.

A computer system called TCS (Train Crossing Support) will be created to help the Train Crossing Coordinator calculate the optimal movements and minimize the train crossing time. The user will use TCS to simulate all train movements and ensure they are sound and safe. After that, the TCS will generate a list of tasks to be dispatched to train drivers. The Train Crossing Coordinator will dispatch them individually, monitoring their progress in locus.

The Yard Manager must monthly report some key performance indicators of this operation to its boss. Their goal is to reduce the total time spam of this kind of operation by 10% in one year.

Taggart Transcontinental Railroad hired you to develop the TCS system.

### Questions:

1. What are the actors involved in this operation?

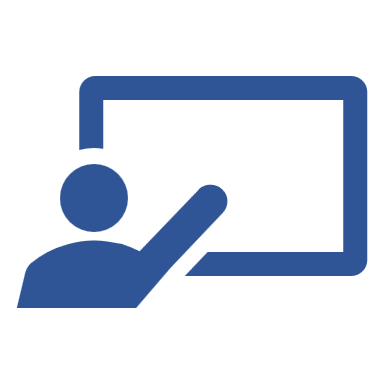
crossing coordinator, yard manager e train driver

1. Write two user stories

o crossing coordinator quer simular os movimentos do trem para gerar uma lista de tarefas para os maquinistas

o yard manager quer gerar um relatório com o os indicadores de desempenho para o chefe

1. Depict the high-level data model



YARD MANAGER

TRAIN DRIVER

CROSSING COORDINATOR

COLABORADOR

PATIO\_FERROVIARIO

TREM

OPERACAO\_TRANSITO

VIAGEM

1. Depict the process of a train crossing
2. What architecture do you suggest? (Client-Server? Web? Mobile?) and Why?

SUGIRO DUAS PLANTAFORMAS, A MOBILE E A WEB. A WEB TERÁ AS APIs PARA FAZER A INTERFACE COM O MOBILE E A APLICAÇÃO WEB.

a APLICAÇÃO WEB DEVE TER AS TELAS PARA UTILIZAÇÃO PELO CROSSING COORDINATOR E YARD MANAGER, O QUE SERIA UMA TELA MAIS AMIGÁVEL PARA VISUALIZAR OS DADOS E RELATÓRIO.

A PLATAFORMA MOBILE SERIA UTILIZADA PELO TRAIN DRIVER, QUE RECEBERIA O PASSO A PASSO PARA MANOBRAR O TREM NO PÁTIO, PODENDO APONTAR O INÍCIO O TÉRMINO (DATA/HORA) DA OPERAÇÃO DE TRÂNSITO. NO MOBILE AINDA TEMOS A POSSIBILIDADE DE MONITORAR O TRAJETO, ATRAVÉS DO GPS, EXIBINDO O CAMINHO PERCORRIDO.

1. Consider that TCS will be implemented using Java family technologies. What frameworks would you use? Explain your decisions.

eu utilizaria o spring boot, o qual já fornece uma estrutura na qual ficaria fácil de entregar as apis rest. para acessar a base o hibernate com jpa, map

1. How long would it take to implement TCS? Explain your estimate.

creio que a implementação duraria cerca de 30 dias, levando em consideração que os requisitos deveriam ser MAIS BEM detalhados, devemos mapear as apis, criar o frontend utilizando angular e ainda temos que criar a aplicação mobile. e em todas as plataformas, ainda devemos implementar as classes de teste.

* 3 dias: requisitos
* 8 dias: apis
* 8 dias: mobile
* 8 dias: web
* 3 dias implantação

### Programming Task:

Implement a program, in ANY programming language, to calculate the train movements for a given input. The input is the details of the trains arriving in both directions of the yard and the yard length. The output is the list of train movements to perform the train crossing.

# Technological Literacy

## AI and Machine Learning

Question:

1. How could AI or Machine Learning could be used in the TCS system described in Section 3.1?

tenho pouco conhecimento de ia, porém podemos a partir da informação dos trens como: quantidade de veiculos, velocidade, geoloacalização, ensinar a máquina a reconhecer os padrões e antes que os maquinistas cheguem ao destino, indicar como será a manobra, baseado nas manobras mais semelhantes ao dado momento.

## System simulation

How could a discrete event simulator be used by Taggart Transcontinental Railroad to improve its train crossing operations?

## Databases

Question:

1. Would you use a SQL or a Non-SQL database for the TCS system? Explain your answer.

creio que o banco de dados non-sql como o mongodb, seria uma alternativa melhor. primeiro porque poderíamos já guardar as informações no formato json, o que facilitaria a elaboração das apis, segundo porque o mongo permite uma escalabilidade horizontal podendo distribuir em diversas máquinas o processamento.

## Coding skills

### Basic coding skills

Implement a function to print a diamond in ASCII art, as shown below:



Figure 4: Examples for length=3, 5 and 7

The length of the diamond is a parameter of the function. The output is a String with the diamond.

You can choose ANY programming language. Minimize the program size: The best program is the one with the least number of lines of code.

Please provide both the source code and the proof that it works.

### Coding skills

Figure 5 shows a small function coded in a programming language that you most certainly do not use on a daily basis.

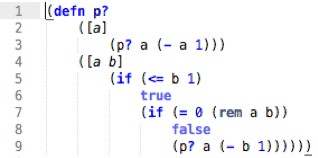


Figure 5: Code example

1. What is the result for (p? 5)
2. Translate the program to either Java or Groovy

### Object oriented and Java programming

1. What are “Tell Don’t Ask” and “Fail Fast”?

“tell don’t ask”: é o conceito básico para orientação a objetos, ajudando a utilizar a questão do encapsulamento. teremos as propriedades e os métodos que devem ser devinicos comforme a realidade, em um objeto do tipo veículo, podemos ter uma ação de “acelerar”, em UM OBJETO CONTA, PODEMOS TER UMA AÇÃO “SACAR”. NESSE OBJETO TEREMOS TODAS AS VALIDAÇÕES ENCAPSULADAS TAMÉM, POR EXEMPLO, O CARRO TEM UM LIMETE DE VELOCIDADE E PARA SACAR DINHEIRE, A CONTA TEM QUE TER SALDO.

“fail fast”: consiste em falhar o mais rápido possível para detectar os possíveis erros e diminuir o “prejuízo” DE TER QUE COLHER OS MALS FRUTOS DO PROBLEMA TARDIAMENTE. SUA APLICAÇÃO PRÁTICA É CODIFICAR DE MANEIRA A PREVER TODAS AS EXCEÇÕES ANTES DE PROSSEGUIR, DESSA FORMA PENSANDO NA CONTA BANCÁRIA, JÁ VALIDO SE A PESSOAL TEM O SALDO PARA REALIZAR O SAQUE, CASO NÃO TENHA, JÁ RETORNO UMA EXCESSÃO, AVISANDO AO SISTEMA QUE ELE NÃO PODE PROSSEGUIR.

1. What is the “Anemic domain model.” Is it a Pattern or an Anti-pattern? Why?

NO MODELO ANÊMICO, SÃO DEFINIDAS BASICAMENTE AS PROPRIEDADES DO OBJETO E AS AÇÕES LIGADAS AO OBJETO SÃO REALIZADAS EXTERNAMENTE. MARTIN FOWLER DEFINE COMO ANTI-PADRÃO, MAS QUANDO UTILIZAMOS OS DATA TRANSFER OBJECTS (DTO), ELE É BASTANTE IMPORTANTE, UMA VEZ QUE O OBJETO NÃO PRECISA TER REGRAS, SOMENTE TER SUAS PROPRIEDADES TRANSFERIDAS PARA SEREM TRANSMITIDAS

1. What is the “KISS” principle?

KEEP AND SIMPLE, STUPID: CONSISTE EM TER O CÓDIGO O MAIS SIMPLES POSSÍVEL, DIVIDINDO O CÓDIGO EM PROBLEMAS MAIS SIMPLES, COM O INTUITO DE QUE QUANDO O PROGRAMADOR VER O CÓDIGO, JÁ SABE O SEU OBJETIVO.

1. Services come in 3 flavors: Domain Services, Application Services, and Infrastructure Services. Briefly explain each one of them.

DOMAIN SERVICES: CARREGAM AS REGRAS DE NEGÓCIO DO DOMÍNIO E SERVEM PARA MANIPULAR AS ENTIDADES DE DOMÍNIO

APPLICATION SERVICES: ESTES POR SUA VEZ RECEBEM AS INFORMAÇÕES E CHAMA OS MÉTODOS DOS SERVIÇOS DE DOMÍNIO.

INFRASTRUCTURE SERVICES: ENVIA E RECEBE MENSAGENS, DANDO APOIO AS OUTRAS CAMADAS. NELA TEMOS OS REPOSIÓRIOS E PERSISTÊNCIA DE DADOS.

1. What is your personal experience with Command-query separation (CQS)?

a idéia é que um médoto pode ter somente uma funcão, ou ele altera o estado do objeto (command) ou retorna a informação do objeto, sem alterar sue estado.

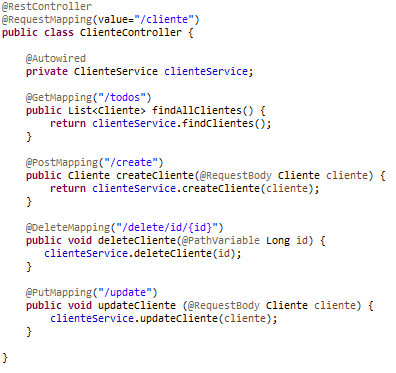
1. What is an endpoint?

endpoint é uma url do sistema para fornecer acesso a alguma funcionalidade para aplicações externas.

1. What is “Convention over configuration,” and how it is applied in the Java Spring framework?

o coc visa simplificar a implementação diminuindo o uso de configurações pelo desenvolvedor, mas com a flexibilidade de permitir que ele possa personalizar essas configurações. esse padrão é adotado por diveros framewors, como no asp.net mvc, no zend framework, no ruby on rails. na figura abaixo temos um código que demonstra isso, o nome da classe já diz que ela é um controller, e as anotações (iniciadas com @) permitem a personalização pelo desenvolvedor.

@getmapping(“/todos”), por exemplo, define que quando o usuário chamar a rota /cliente/todos, será executado o método findallclientes.



What are the Spring annotations @RestController, @Controller, @Service, @Repository, and @Component?

@RestController: define classes que serão expostas na aplicação web para receber requisições, porém no modelo restful, no qual não é necessário uma view para exibir o resultado para o cliente. ela é uma combinação de @controller e @responsebody.

@Controller: tem a mesma função da @restcontroller, porém com a diferença que indica que está utilizando o padrão mvc e necessita de uma view para exibir seu resultado.

@Service: usada para definir as classes da camada de serviço

@Repository: usada para definir as dao’s

@Component: indica que a classe deve ser gerenciada pelo spring, se tornando um spring bean, permitindo assim que o spring use essa classe automaticamente. anotações como @configuration, @controller, @repository, etc derivam de @component

1. What is and how do you use dependency injection?

di é um padrão do desenvolvimento ágil que consiste em dizer que a classe depende de outra para executar determinada rotina. devido o uso da vem da teoria do “single responsibility principle” do solid, eu simplifico a função da classe e indico quais as outras classes são necessárias para eu executar determinada rotina.

um exemplo é se eu tiver uma classe de pedido, ela tem a sua responsabilidade de realizar o pedido, porém ela depende do serviço de estoque, do serviço de entrega. na classe de pedido faço a injeção de dependencia das classe de estoque, que irá verificar se existe estoque diponível para aquele produto e também do serviço de entrega, o qual é responsável por checar se entrega em determinada região ou não.

1. What is the design pattern you use most? Why?
2. How long have you been using Linux as the primary development platform?

utilizo linux a mais ou menos uns 20 anos, pois as aplicações que eu produzia, geralmete eram publicadas nesse ambiente.

## Multiprocessing and multi-threading

NÃO TENHO CONHECIMENTO DESSE TIPO DE IMPLEMENTAÇÃO. POR ISSO não VOU RESPONDER AS QUESTÕES ABAIXO.

Consider the code in parallel-pascal of Figure 6

The program may fail if there is more than one thread calling sign\_up() in parallel.

1. Why does it fail?
2. How would you fix it?
3. How would you fix it in Java?

## Virtualization, Containerization, Load Balancing

NÃO TENHO CONHECIMENTO DESSE TIPO DE IMPLEMENTAÇÃO. POR ISSO não VOU RESPONDER AS QUESTÕES ABAIXO.

1. What is the difference between virtualization and containerization?
2. How would you use an AWS Lambda function on the TCS system described in Section 3.1

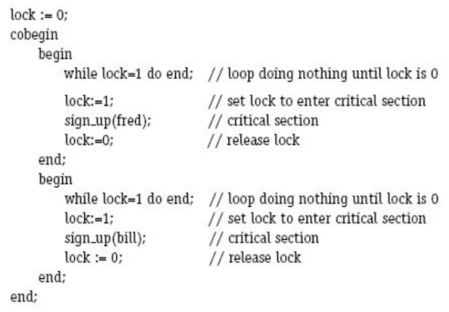


Figure 6: Example in parallel-pascal

# Data Literacy

## Statistics I

Your customer measures how his employees are using your product. To do that, he measures how many times each user accesses your system in a day. He measured that for more than 2 years and sent you a spreadsheet with more than 100K entries of the tuple (user\_id, number\_of\_openings, day\_of\_the\_year).

Explain how you will use this data to:

1. Find the best and the worst users (the ones that use more and less your system) 2. Detect if adoption is increasing or decreasing

melhores usuários: somaria a quantidade de acessos agrupados por usuário e ordenaria por ordem decrescente, os primeiros da lista são os que mais acessam o sistema

piores usuários: somaria a quantidade de acessos agrupados por usuário e ordenaria por ordem crescente, os primeiros da lista são os que menos acessam o sistema

1. Find if the training you gave on a certain day had a positive effect on system usage
2. Predict how will be the system usage next month

Please explain for each case what statistical and visualization tools you would use.

## Statistics II

Your friend promised you $10 if you find a ball hidden in a cup as shown in Figure

7. You chose cup number one. Before opening the selected cup, your friend decided

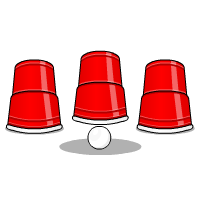


Figure 7: Find the ball game

to give you a chance to switch your selection. He raised cup number 2 and showed that it was empty. Now you can keep your initial selection (cup number one) or switch to cup number three.

1. Do you keep your initial choice or switch to cup number 3? Explain your decision.

mantenho o como 1. não tenho uma justificativa, apenas estou arriscando.

# Human Literacy

## Theory of money

Someone said to you: “Money is the root of all evil”.

1. What is money?

meio pelo qual se faz trocas comerciais

1. What would be your answer to your friend?

não concordo, a raiz de todos os males é a cobiça

# Cultural agility

## Multi culture

1. A software project is an opportunity to improve both hard and soft skills. In what ways a software project makes you a better person?

o desenvolvimento de um sistema exige uma troca de ideias de experiências entre o cliente e o desenvolvedor, entre o desenvolvedor e a equipe. isso faz com que haja uma interação social que vem a criar vínculos.

creio que me torno uma pessoa melhor no momento que interajo com outras e principalmente na absorvição de novos conhecimentos, de enfrentar desafios.

1. Do you have experience working in a multi-cultural team? Have you ever faced a cultural barrier? If yes, please shortly describe it.

não trabalhei ainda.

# Entrepreneurship

## Innovation

Have you ever innovated? If yes, please shortly describe your experience.

inovei e sempre e sempre procuro inovar, ir atrás de me superar buscando novos desafios.

eu saí de minha cidade, que é pequena para ir para a capital do país em busca de melhoria profissional e financeira, consegui, me estabilizei, cresci. mas por motivos pessoais retornei para minha cidade e aqui, dois anos e meio depois, resolvi sair da empresa que eu estava, no regime de clt e resolvi trabalhar por conta própria, visando além do ganho financeiro, sair da rotina de um lugar que já não me satisfazia mais como antes.

# Programming skills

Please click on this [link](https://bitbucket.org/nitryx-team/rh-dev-2/src/master/) and follow the instructions