

Esta seção tem por finalidade apresentar os conceitos necessários para devida compreensão e execução das etapas subsequentes.

Anomalias de Código

Uma Anomalia de Código, ou *Bad Smell*, pode ser definido como quaisquer características no código-fonte de um programa que *possivelmente* indiquem um problema maior (FOWLER, 2018). Ainda, as anomalias podem ser conceituados como sendo "*certas estruturas no código que indiquem a violação de princípios básicos de design e impactam negativamente na sua qualidade*" (SURYANARAYANA, 2014). Sendo assim, uma anomalia não necessariamente seria considerada como um defeito no código, entretanto pode servir como um indicador de problemas futuros os quais, por sua vez, poderiam desencadear a confecção de artefatos de software defeituosos.

Carga Cognitiva

Esforço cognitivo, ou carga cognitiva se refere ao nível de utilização de recursos psicológicos como **memórias, atenção, percepção, representação de conhecimento, raciocínio e criatividade** na resolução de problemas. Os principais fatores ligados ao aumento da carga cognitiva são: **experiência com atividades similares, grau de abstração, número de atividades simultâneas, horas de descanso, nível de fadiga**, entre outros. Enquanto a resolução de um problema sem nenhuma experiência prévia é considerado como muito desgastante, a resolução de um problema ao qual já foi desenvolvido um padrão de respostas usuais (esquema) envolve apenas um mínimo de esforço cognitivo que facilmente passa despercebido pelo sujeito (SWELLER, 1988).

Referências

FOWLER, Martin. **Refactoring: improving the design of existing code**. Addison-Wesley Professional, 2018.

SURYANARAYANA, Girish; SAMARTHYAM, Ganesh; SHARMA, Tushar. **Refactoring for software design smells: managing technical debt**. Morgan Kaufmann, 2014.

SWELLER, John. Cognitive load during problem solving: Effects on learning. **Cognitive science**, v. 12, n. 2, p. 257-285, 1988.

Próxima



Perfil

Estas perguntas têm por objetivo trazer uma visão geral sobre o perfil dos participantes.

Qual a sua idade? *

- menos de 20 anos
- de 20 a 29 anos
- de 30 a 39 anos
- de 40 a 49 anos
- mais de 49 anos

Qual a sua escolaridade? *

- Ensino Médio
- Ensino Técnico
- Ensino Superior Incompleto
- Ensino Superior Completo
- Especialização/MBA
- Mestrado/Doutorado
- Outros

Há quanto tempo trabalha na área de Tecnologia da Informação? *

- até 2 anos
- de 3 a 4 anos
- de 5 a 6 anos
- de 7 a 8 anos
- acima de 8 anos

Qual opção abaixo descreve melhor o seu cargo atual? *

- Desenvolvedor de Software/Programador
- Engenheiro de Software
- Arquiteto de Software
- Tech Lead
- Analista de Sistemas
- Outros

Email *

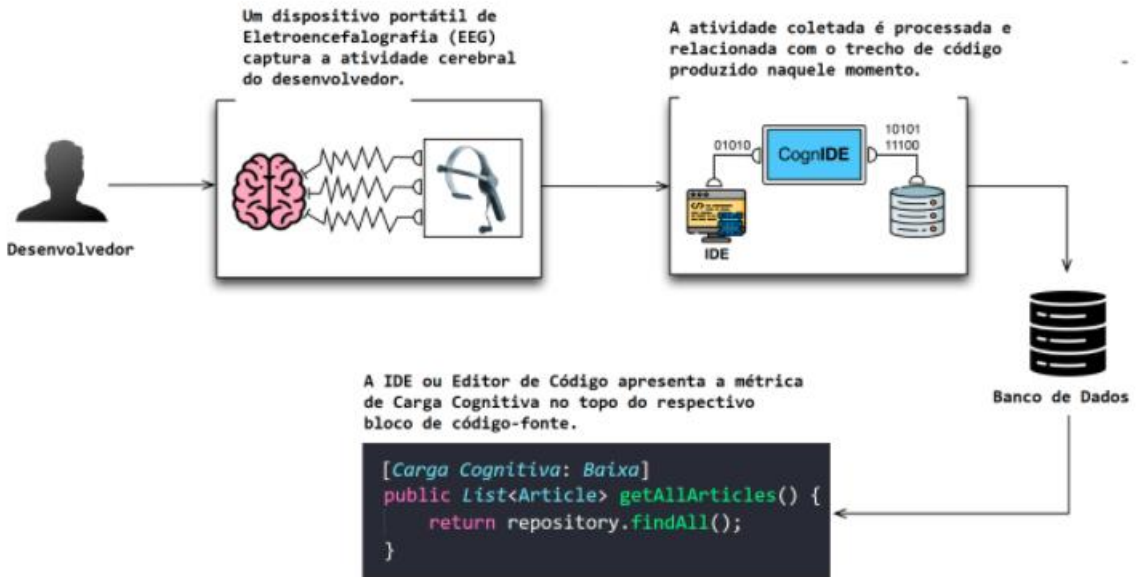
Este e-mail será utilizado apenas se for necessária entrevista e não será divulgado de forma alguma.

Voltar

Próxima

Percepção e Recomendação de Refactoring

Você é o líder técnico de um time de desenvolvimento, onde sua principal função é a de **garantir a qualidade do código**. Você foi incumbido de revisar novos artefatos utilizando uma ferramenta experimental, a qual indica o nível de **Carga Cognitiva** (ou esforço mental) apresentada pelo **desenvolvedor** durante a produção do código, conforme ilustrado a seguir:



Fazendo uso da funcionalidade apresentada, você deverá revisar os trechos de código apresentados a seguir, respondendo se você **percebeu** alguma anomalia de código (*Bad Smell*) e/ou o quanto você **recomendaria** o seu refactoring.

Questão 01

Analise o trecho de código a seguir e responda às perguntas:

```
@Service
public class CSVService {
    @Autowired
    ArticleRepository repository;

    public void save(MultipartFile file) {
        try {
            List<Article> articles = CSVHelper.csvToArticles(file.getInputStream());
            repository.saveAll(articles);
        } catch (IOException e) {
            throw new RuntimeException("Failed save the CSV data: " + e.getMessage());
        }
    }

    public ByteArrayInputStream load() {
        List<Article> articles = repository.findAll();

        ByteArrayInputStream in = CSVHelper.articlesToCSV(articles);
        return in;
    }

    public List<Article> getAllArticles() {
        return repository.findAll();
    }
}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

- Sim
- Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

- Certamente não recomendaria
- Não recomendaria
- Talvez recomendasse
- Recomendaria
- Certamente recomendaria

Questão 02

Analise o trecho de código a seguir e responda às perguntas:

```
public interface Responsible {  
  
}
```

```
public abstract class BaseResponse implements Responsible {  
  
    private String correlationId;  
  
    public BaseResponse(String correlationId) {  
        this.correlationId = correlationId;  
    }  
  
    public String getCorrelationId() {  
        return correlationId;  
    }  
  
}
```

```
public abstract class BaseResponseMessage extends BaseResponse {  
  
    private String messageType;  
  
    public BaseResponseMessage(String messageType, String correlationId) {  
        super(correlationId);  
        this.messageType = messageType;  
    }  
  
    public String getMessageType() {  
        return messageType;  
    }  
  
}
```

```
public class ResponseMessage extends BaseResponseMessage {  
  
    private String message;  
  
    public ResponseMessage(String message, String messageType, String correlationId) {  
        super(messageType, correlationId);  
        this.message = message;  
    }  
  
    public String getMessage() {  
        return message;  
    }  
  
    public void setMessage(String message) {  
        this.message = message;  
    }  
  
}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Questão 03

Analise o trecho de código a seguir e responda às perguntas:

```
[Carga Cognitiva: Alta]
@Service
public class CSVService {
    @Autowired
    ArticleRepository repository;

    public void save(MultipartFile file) {
        try {
            List<Article> articles = CSVHelper.csvToArticles(file.getInputStream());
            repository.saveAll(articles);
        } catch (IOException e) {
            throw new RuntimeException("Failed save the CSV data: " + e.getMessage());
        }
    }

    public ByteArrayInputStream load() {
        List<Article> articles = repository.findAll();

        ByteArrayInputStream in = CSVHelper.articlesToCSV(articles);
        return in;
    }

    public List<Article> getAllArticles() {
        return repository.findAll();
    }
}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

- Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

- Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Questão 04

Analise o trecho de código a seguir e responda às perguntas:

```
[Carga Cognitiva: Baixa]
@Service
public class CSVService {
    @Autowired
    ArticleRepository repository;

    public void save(MultipartFile file) {
        try {
            List<Article> articles = CSVHelper.csvToArticles(file.getInputStream());
            repository.saveAll(articles);
        } catch (IOException e) {
            throw new RuntimeException("Failed save the CSV data: " + e.getMessage());
        }
    }

    public ByteArrayInputStream load() {
        List<Article> articles = repository.findAll();

        ByteArrayInputStream in = CSVHelper.articlesToCSV(articles);
        return in;
    }

    public List<Article> getAllArticles() {
        return repository.findAll();
    }
}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

- Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

- Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Questão 05

Analise o trecho de código a seguir e responda às perguntas:

```
[Carga Cognitiva: Alta]
public interface Responsible {

}
```

```
[Carga Cognitiva: Alta]
public abstract class BaseResponse implements Responsible {

    private String correlationId;

    public BaseResponse(String correlationId) {
        this.correlationId = correlationId;
    }

    public String getCorrelationId() {
        return correlationId;
    }

}
```

```
[Carga Cognitiva: Alta]
public abstract class BaseResponseMessage extends BaseResponse {

    private String messageType;

    public BaseResponseMessage(String messageType, String correlationId) {
        super(correlationId);
        this.messageType = messageType;
    }

    public String getMessageType() {
        return messageType;
    }

}
```

```
[Carga Cognitiva: Alta]
public class ResponseMessage extends BaseResponseMessage {

    private String message;

    public ResponseMessage(String message, String messageType, String correlationId) {
        super(messageType, correlationId);
        this.message = message;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

- Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

- Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Questão 06

Analise o trecho de código a seguir e responda às perguntas:

```
[Carga Cognitiva: Baixa]
public interface Responsible {

}
```

```
[Carga Cognitiva: Baixa]
public abstract class BaseResponse implements Responsible {

    private String correlationId;

    public BaseResponse(String correlationId) {
        this.correlationId = correlationId;
    }

    public String getCorrelationId() {
        return correlationId;
    }

}
```

```
[Carga Cognitiva: Baixa]
public abstract class BaseResponseMessage extends BaseResponse {

    private String messageType;

    public BaseResponseMessage(String messageType, String correlationId) {
        super(correlationId);
        this.messageType = messageType;
    }

    public String getMessageType() {
        return messageType;
    }

}
```

```
[Carga Cognitiva: Baixa]
public class ResponseMessage extends BaseResponseMessage {

    private String message;

    public ResponseMessage(String message, String messageType, String correlationId) {
        super(messageType, correlationId);
        this.message = message;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Questão 07

Analise o trecho de código a seguir e responda às perguntas:

```
[Carga Cognitiva: Alta]
@PostMapping("/upload")
public ResponseEntity<ResponseMessage> uploadFile(@RequestParam("file") MultipartFile file) {
    String message = "";
    String TYPE = "text/csv";
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    Date date = new Date();

    if (!TYPE.equals(file.getContentType())) {
        try (BufferedReader fileReader = new BufferedReader(
            new InputStreamReader(file.getInputStream(), "UTF-8"));
            CSVParser csvParser = new CSVParser(
                fileReader,
                CSVFormat.DEFAULT.withFirstRecordAsHeader()
                    .withIgnoreHeaderCase()
                    .withTrim());
        ) {

            System.out.println("Datetime: " + formatter.format(date) + "Message: Parsing the CSV file.");
            List<Article> articles = new ArrayList<Article>();
            Iterable<CSVRecord> csvRecords = csvParser.getRecords();

            for (CSVRecord csvRecord : csvRecords) {
                Article article = new Article(Long.parseLong(csvRecord.get("Id")), csvRecord.get("Title"),
                    csvRecord.get("Description"), Boolean.parseBoolean(csvRecord.get("Published")));

                articles.add(article);
            }

            System.out.println("Datetime: " + formatter.format(date) + "Message: Saving the parsed articles.");
            repository.saveAll(articles);

            message = "Uploaded the file successfully: " + file.getOriginalFilename();
            return ResponseEntity.status(HttpStatus.OK).body(new ResponseMessage(message));
        } catch (Exception e) {

            System.out.println("Datetime: " + formatter.format(date) +
                "Message: An error has been occurred while uploading the file.");
            message = "Could not upload the file: " + file.getOriginalFilename() + "!";
            return ResponseEntity.status(HttpStatus.EXPECTATION_FAILED).body(new ResponseMessage(message));
        }
    }

    System.out.println("Datetime: " + formatter.format(date) + "Message: File uploaded in wrong format.");
    message = "Please upload a file in CSV format.";
    return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(new ResponseMessage(message));
}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

- Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

- Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Questão 08

Analise o trecho de código a seguir e responda às perguntas:

```
[Carga Cognitiva: Baixa]
@PostMapping("/upload")
public ResponseEntity<ResponseMessage> uploadFile(@RequestParam("file") MultipartFile file) {
    String message = "";
    String TYPE = "text/csv";
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
    Date date = new Date();

    if (!TYPE.equals(file.getContentType())) {
        try (BufferedReader fileReader = new BufferedReader(
            new InputStreamReader(file.getInputStream(), "UTF-8"));
            CSVParser csvParser = new CSVParser(
                fileReader,
                CSVFormat.DEFAULT.withFirstRecordAsHeader()
                    .withIgnoreHeaderCase()
                    .withTrim());
        ) {

            System.out.println("Datetime: " + formatter.format(date) + "Message: Parsing the CSV file.");
            List<Article> articles = new ArrayList<Article>();
            Iterable<CSVRecord> csvRecords = csvParser.getRecords();

            for (CSVRecord csvRecord : csvRecords) {
                Article article = new Article(Long.parseLong(csvRecord.get("Id")), csvRecord.get("Title"),
                    csvRecord.get("Description"), Boolean.parseBoolean(csvRecord.get("Published")));

                articles.add(article);
            }

            System.out.println("Datetime: " + formatter.format(date) + "Message: Saving the parsed articles.");
            repository.saveAll(articles);

            message = "Uploaded the file successfully: " + file.getOriginalFilename();
            return ResponseEntity.status(HttpStatus.OK).body(new ResponseMessage(message));
        } catch (Exception e) {

            System.out.println("Datetime: " + formatter.format(date) +
                "Message: An error has been occurred while uploading the file.");
            message = "Could not upload the file: " + file.getOriginalFilename() + "!";
            return ResponseEntity.status(HttpStatus.EXPECTATION_FAILED).body(new ResponseMessage(message));
        }
    }

    System.out.println("Datetime: " + formatter.format(date) + "Message: File uploaded in wrong format.");
    message = "Please upload a file in CSV format.";
    return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(new ResponseMessage(message));
}
```

Você percebeu alguma anomalia de código (Bad Smell) no trecho apresentado?

- Sim Não

O quanto você recomendaria o refactoring do trecho de código apresentado? *

- Certamente não recomendaria Não recomendaria Talvez recomendasse Recomendaria
 Certamente recomendaria

Voltar

Próxima

CognIDE

Introdução

Perfil

Questionário

Avaliação

Avaliação

As questões a seguir visam avaliar a sua experiência ao responder este questionário. Utilize as opções para dizer o quanto você concorda com cada uma das afirmações abaixo.

Clareza *

	Discordo totalmente	Discordo	Neutral	Concordo	Concordo totalmente
Tive tempo o suficiente para responder o questionário.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O treinamento me capacitou para executar o experimento.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As instruções do treinamento foram claras.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As questões apresentadas foram claras.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Percepção *

	Discordo totalmente	Discordo	Neutral	Concordo	Concordo totalmente
Não tive dificuldade em compreender o código apresentado.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Não tive dificuldade em compreender a métrica de Carga Cognitiva apresentada.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Efeitos *

	Discordo totalmente	Discordo	Neutral	Concordo	Concordo totalmente
Compreendi o que são anomalias (Bad Smells) de código.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compreendi o que é Carga Cognitiva.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Percepção de Uso *

	Discordo totalmente	Discordo	Neutral	Concordo	Concordo totalmente
A métrica de Carga Cognitiva auxiliou na percepção de anomalias de código.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A métrica de Carga Cognitiva influenciou na recomendação da refatoração do código apresentado.*	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Voltar

Submeter