

## Features do Prisma Client

Com o conteúdo que praticamos nos vídeos, podemos afirmar que:

Selecione as alternativas corretas:

*Selecione 3 alternativas*

**A**

O Prisma Client difere de outros ORMs (por exemplo, o Sequelize) pois utiliza o Prisma Data Model para gerar os métodos “customizados” de acordo com os modelos e os relacionamentos entre eles.

**B**

O código abaixo retorna uma array com todos os posts de todos os usuários:

```
postsByUser: async (_, args) => {  
  return prisma.user  
    .findMany()  
    .posts()  
},
```

**C**

É possível acessar os modelos de forma *fluenta*, através de funções encadeadas, sempre observando os objetos retornados por cada uma delas.

**D**

Os dois resolvers abaixo para o campo `postsByReviewer` são equivalentes (no sentido de que retornam os mesmos dados):

```
postsByReviewer: async (_, args) => {  
  const reviewer = await prisma.review.findUnique(  
    {  
      where: {  
        id: args.id,  
      },  
    },  
  )  
  return prisma.post.findMany(  
    {  
      where: {  
        reviewerId: reviewer.id,  
      },  
    },  
  )  
}
```

```
      where: { id: Number(args.id) }
    })
    return prisma.post.findMany({
      where: {
        autorId: Number(revieweeReviewerId)
      }
    })
  },
}

E

postsByReviewer: async (_, args) => {
  return prisma.review
    .findUnique({ where: { id: Number(args.id) } })
    .reviewer()
    .posts()
},
```