# Description:

This project tries to approach games and gaming culture from the perspective of accesibility, doing so by categorizing each entry by which diversity it is accesible to (in this case I chose to tackle general categories: Hearing, Sensory, Cognitive, Motor and Visual). In order to have quicker access to a specific diversity, users can filter games through them, as well as go to a more detailed page for everygame in order to learn more about them and their developers.

# Models:

In order to collect and manipulate the data for this project I created two main models. One for the first view of the games and other for a more detailed view. As you can see in the comments, both of them could be much improved once we learn how to do relations between models.

Game's model:

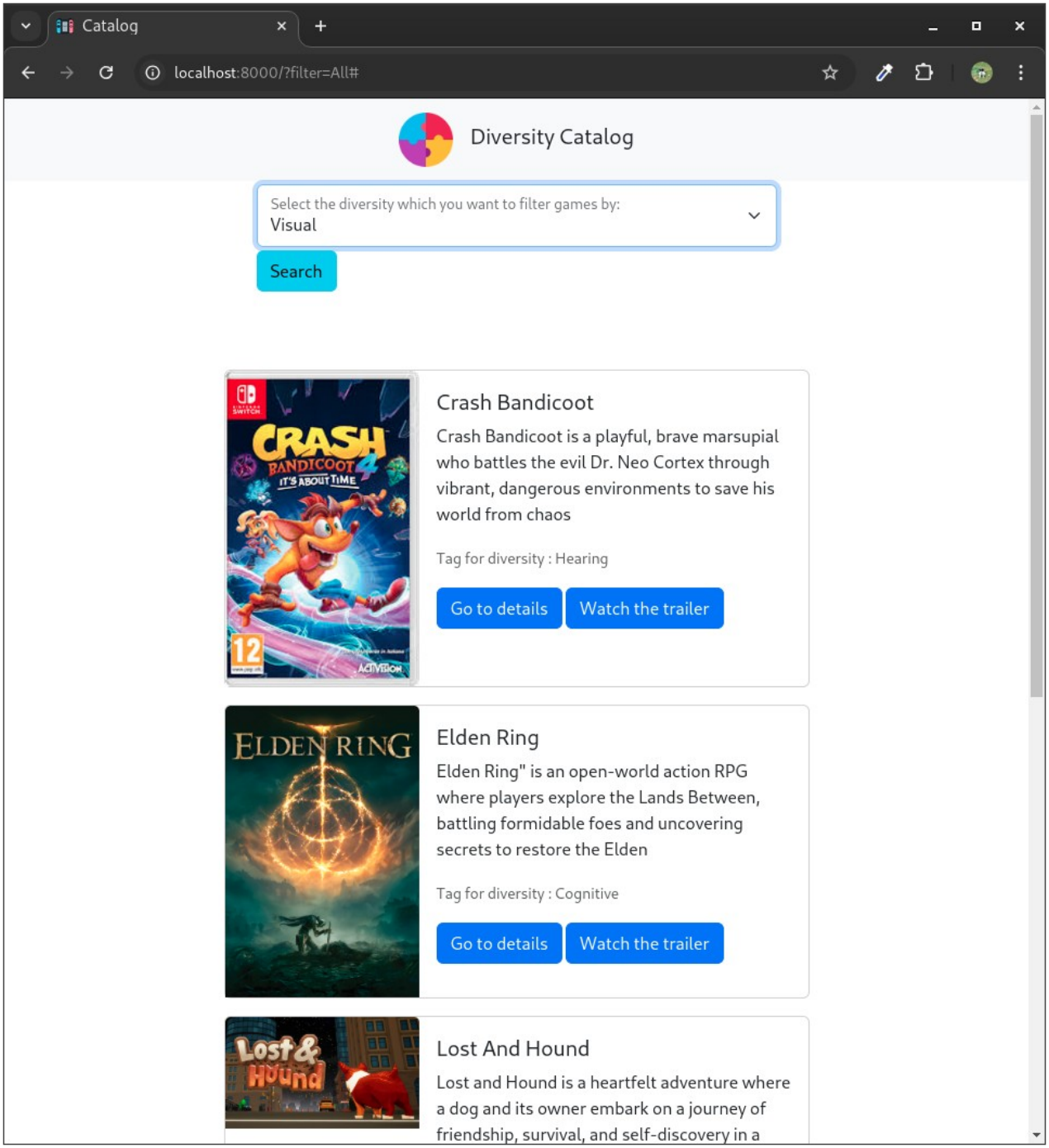
```python
class Game(models.Model):
    VISUAL = "Visual"
    MOTOR = "Motor"
    COGNITIVE = "Cognitive"
    HEARING = "Hearing"
    SENSORY = "Sensory"
    DIVERSITY_TYPE = [
        (VISUAL, "Visual"),
        (MOTOR, "Motor"),
        (COGNITIVE, "Cognitive"),
        (HEARING, "Hearing"),
        (SENSORY, "Sensory"),
    ]
    title = models.CharField(max_length=500)
    cover = models.ImageField(upload_to='catalogo/images')
    #TODO outra táboa para darlle múltiples diversities a cada
    xogo, xa que unha obra pode cubrir varias accesibilidades.
    diversity = models.CharField(choices=DIVERSITY_TYPE,
    default=VISUAL,max_length=10)
    description = models.TextField(max_length=150)
    trailer = models.URLField(blank=True)
```

Detail's model:

```python
class Detail(models.Model):
  score = models.IntegerField(
    validators=[
      MinValueValidator(1),
      MaxValueValidator(10)
    ]
  )
  description = models.TextField(blank=True, max_length=1500)
  #TODO también multivaluado
  developer= models.CharField(blank=True, max_length=50)
  developerPage = models.URLField(blank=True)
  # TODO prices táboa con web scraping sobre os prezos dos
  xogos
  # TODO plataformas es otro atributo multivaluado
```

# Screenshots:

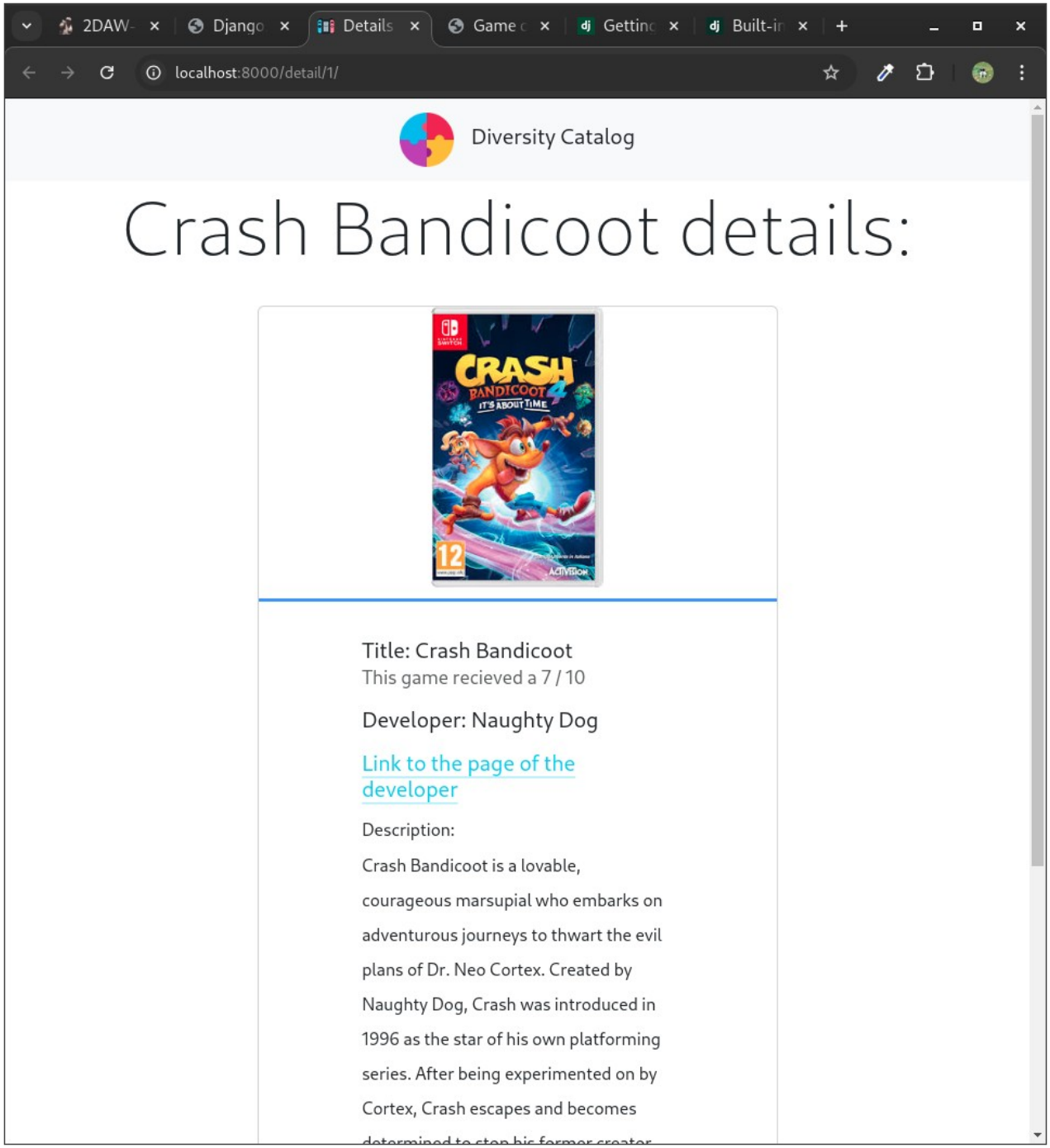
## *Main Page:*




Ñ

## ***Detail page:***

The nav Logo links to the main page to ease navigation
(https://www.flaticon.com/free-icon/unity_2076500)



# Challenges:

The main challenge of the project came out to be the fact that we couldn`t make relations between our models. I solved this by making that relationship in the urls and by accesing a different details object by the game_id that was sent.

Here, the detail app url takes the game_id from the object is was linked by the first picture (found at the catalogo home.html)

```
<a href="detail/{{game.id}}" class="btn
btn-primary" target="_blank">Go to
details</a>
```

```
app_name = 'detail'

urlpatterns = [
    path('<int:game_id>/',views.detail,
    name="detail")
]
```

Here, in the detail app views, we check if the game sent exists and so, take the information to the deatils that share that exact ID in our database in order to send it to the template and show it.

```
def detail(request, game_id):
    game = get_object_or_404(Game,pk=game_id)
    detail = Detail.objects.get(pk=game_id)
    return render(request,'detail/details.html',{'game':game,
    'detail':detail})
```

The main problem with this method is that we have to update both models each time we integrate a game object in our database in order to access it's details, a problem easily solved once we learn to make relationships between them via models.

## *Future challenges:*

It's my plan to keep working on this project and making it the final project, in order to do so, I'm likely to find myself in other trenches such as allowing user input to create and update the database. Making the web itself as accessible as posible, since it is at the core of the idea. Add guides or templates of them to help developers to make games more accesible.