```
class Recipe:
    """
    Represents a recipe with a name, instructions, list of ingredients, and a URL
(source).

    Attributes:
        name (str): The name of the recipe.
        instructions (str): The cooking instructions for the recipe.
        ingredients (list): A list of Ingredient objects used in the recipe.
        url (str): The URL source of the recipe, if available.

    Methods:
        add_ingredient(ingredient): Add an Ingredient to the list of ingredients.
        remove_ingredient(ingredient): Remove an Ingredient from the list of
ingredients.
        has_ingredient(ingredient_name): Check if the recipe has a specific
ingredient.
        __str__(): Return a string representation of the recipe (name and
ingredients).

    """
    def __init__(self, name, instructions, ingredients, url):
        """
        Initialize a Recipe instance.

        Args:
            name (str): The name of the recipe.
            instructions (str): The cooking instructions for the recipe.
            ingredients (list): A list of Ingredient objects used in the recipe.
            url (str): The URL source of the recipe, if available.
        """
        self.name = name
        self.instructions = instructions
        self.ingredients = ingredients
        self.url = url

    def add_ingredient(self, ingredient):
        """
        Add an ingredient to the recipe's list of ingredients.

        Args:
            ingredient (Ingredient): The Ingredient object to be added.
        """
        self.ingredients.append(ingredient)

    def remove_ingredient(self, ingredient):
        """
        Remove an ingredient from the recipe's list of ingredients.

        Args:
            ingredient (Ingredient): The Ingredient object to be removed.
        """
        if ingredient in self.ingredients:
            self.ingredients.remove(ingredient)

    def has_ingredient(self, ingredient_name):
        """
        Check if the recipe contains a specific ingredient.
```

```
        Args:
            ingredient_name (str): The name of the ingredient to check for.

        Returns:
            bool: True if the ingredient is found, False otherwise.
        """
        for ingredient in self.ingredients:
            if ingredient.name.lower() == ingredient_name.lower():
                return True
        return False

    def __str__(self):
        """
        Return a string representation of the recipe.

        Returns:
            str: A formatted string containing the recipe's name and a comma-
separated list of ingredient names.
        """
        return f"{self.name}: {', '.join(ing.name for ing in self.ingredients)}"
```