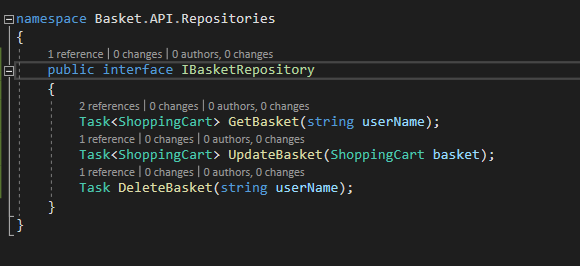
1 - Crie uma Pasta em Basket.API chamada Repositories

- Crie uma Classe/Interface IBasketRepository

\* Task<ShoppingCart> GetBasket(string userName);

\* Task<ShoppingCart> UpdateBasket(ShoppingCart basket);

\* Task DeleteBasket(string userName);

2 - Crie a Clase BasketRepository que vai implementar os contratos feitos pela interface, e faça um extende.

-Implementar Cache distribuido.

\* private readonly IDistributedCache \_redisCahe;

- Faça o construtor passando esse codigo dentro, fica assim:

\* public BasketRepository(IDistributedCache redisCahe)

{

\_redisCahe = redisCahe ?? throw new ArgumentNullException(nameof(redisCahe));

}

3 - Impletente os métodos criados pela Interface e forma Asyncrona "async"

**public async Task<ShoppingCart> GetBasket(string userName)**

{

var basket = await \_redisCache.GetStringAsync(userName);

if (String.IsNullOrEmpty(basket))

return null;

return JsonConvert.DeserializeObject<ShoppingCart>(basket);

}

**public async Task<ShoppingCart> UpdateBasket(ShoppingCart basket)**

{

await \_redisCache.SetStringAsync(basket.UserName, JsonConvert.SerializeObject(basket));

return await GetBasket(basket.UserName);

}

**public async Task DeleteBasket(string userName)**

{

await \_redisCache.RemoveAsync(userName);

}

