



Code Review



Proceso del **code review**

01

Nuevo cambio

La persona crea un PR describiendo los detalles de la implementación.

03

Comentarios

Los devs dejan sugerencias para mejorar el código ó incluso para validar escenarios que no se tomaron en cuenta.

02

Code review

Se pide review a los devs para que valide el nuevo código que se agrega/cambia.

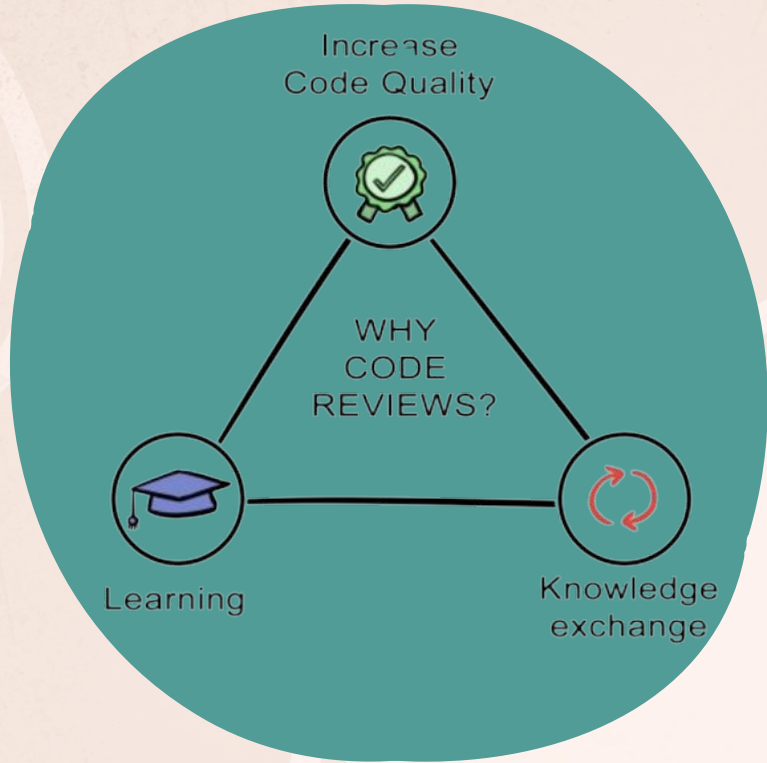
04

Aprobación

Los devs aprueban los cambios y el PR está listo para el siguiente paso.



Por qué hacerlos?



- Mejora habilidades.
- Hacer frente a la deuda técnica.
- Aprender nuevas formas de resolver un mismo problema.
- Intercambiar conocimiento.
- Aprender sobre la lógica del negocio.
- Debatir y aportar nuevas ideas.



The background features a light beige, textured surface with several abstract, colorful elements. On the left and right sides, there are large, stylized molecular or network-like structures. Each structure consists of a central red sphere connected to three teal spheres by thick teal lines. Additionally, there are several smaller, orange line-art diagrams scattered across the page. These diagrams consist of red dots connected by thin orange lines, forming various geometric shapes like triangles and polygons. Some of these diagrams are accompanied by a small teal 'x' or '+' symbol, possibly indicating a specific point of interest or a mathematical relationship.

Pautas para el code review



Para todos

- Aceptar que muchas decisiones de programación son opiniones.
- Solicitar cambios no es personal.
- Hacer buenas preguntas; no hacer demandas.
- Preguntar para aclarar.
- Evitar la propiedad selectiva del código. ("mío", "no mío", "tuyo")
- Ser explícito.
- No usar el sarcasmo.
- No usar la hipérbole. ("siempre", "nunca", "infinitamente", "nada")





Para todos

- Dejar de escribir y tener una conversación.
- Generar confianza: Escribir código de alta calidad. Escribir buenos commits. Escribir buenas pruebas. Si hace esto de manera constante, la gente confiará en usted y aceptará los PR más rápido y escuchará sus comentarios con más facilidad.





[#10364] Checkbox to delete upcoming events on Google Cal when unsynced ✓
 👁 1 82

#10404 by SamBelmor was merged on Aug 31, 2021 • Approved 🔄 5 tasks done

Reviewers

-
-
-
-
-
-

-
-
-
-
-
-

Mar 29th at 11:01 AM

and I were discussing about some preferences in terms of gem versioning [here](#), and I know that we are not specifying at all versions (nothing like equal or the pessimist syntax) for our gems because we are in the middle of the Rails upgrade

Is that something we are going to change once we finish with the Rails upgrade itself? I forgot what was the original plan about versioning

About not specifying at all the gems versions I think it has its own problems just like the ones mentioned in this quick blog post with a video <https://www.moncebelyamani.com/what-happens-when-you-don-t-specify-a-version-in-your-gemfile/>

What's your preference/suggestion/experience in this topic?

@Sam Belmor

moncebelyamani.com

[What Happens When You Don't Specify a Version in Your Gemfile](#)

Moncef Belyamani writes guides to help you become a more confident coder.



02

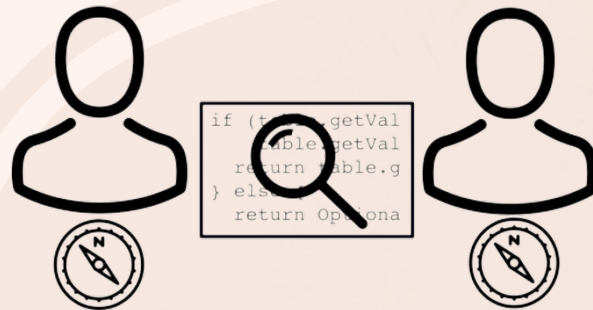
Al dueño del código

- Ser agradecido por las sugerencias del revisor. ("Buena decisión. Haré ese cambio").
- Explicar por qué existe el código. ("Es así por estas razones. ¿Sería más claro si cambio el nombre de esta clase/archivo/método/variable?").
- Tratar de entender la perspectiva del revisor.
- Tratar de responder a cada comentario.

02

Al dueño del código

- Delegar en herramientas siempre que sea posible (uso de linters).
- Agregar una buena descripción del PR.






SamBelmor 12 days ago

Author



Definitely, the JSON structure that you suggest looks better  🎉. This suggestion will be addressed in this [follow-up ticket](#).



 ys ago



This kind of styles are not needed if we follow the Bootstrap Markup 😊, by default, all Columns display in a flex mode -> <https://getbootstrap.com/docs/4.0/layout/grid/#how-it-works>



SamBelmor 13 days ago

Author



Yeap, you were right. Thanks 😊



03

Revisando el código de otros

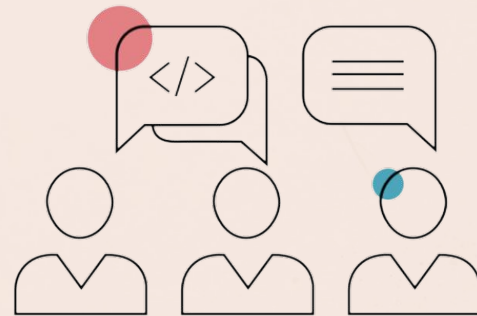


- Comprender por qué es necesario el cambio.
- Comunicar qué ideas le gustan mucho y cuáles no.
- Identificar formas de simplificar el código sin dejar de resolver el problema.
- Ofrecer implementaciones alternativas, pero asumir que el autor ya las consideró.
- Ser explícito con los cambios requeridos.
- Siempre terminar con un “Dime qué opinas al respecto”, “avísame si esto tiene sentido”, etc.
- Firmar el PR con un comentario 👍 o "LGTM".
- Recordar que está aquí para proporcionar comentarios, no para ser un guardián.

03

Revisando el código de otros

- Siempre tratar de enviar todos los comentarios que deseas que atiendan en un solo una primera iteración.
- Dejar un resumen final de comentarios dejados.
- Las sugerencias están bien; las pautas de estilo comunes están bien; pero a veces tienes que mostrar cierto nivel de flexibilidad si no se escribió exactamente de la misma manera en que lo habrías escrito.



1dw-node/src/components/UsersTable/UsersTable.js Outdated

Hide resolved

```
514 +   }  
515 + };  
516 +  
517 + renderUsersTableWithTwoTab = () => {
```



[Redacted] 🗨️ ⋮

I see potential for moving this logic to it's own component. Something like "UserTableTwoTabsLayout". It could be rendered if the user is a manager

```
{showTableWithTwoTabs ? <UserTableTwoTabsLayout/> : this.renderDefaultUsersTable()}
```

Or we can pass the `DefaultUsersTable` as a child element. However if it makes use of its own `BootstrapTable` element it would add the benefit of being able to modify it separately. What do you think?



SamBelmor on Jul 26, 2021

Author 🗨️ ⋮

Discussing this with Alex when I started the tech spec, we thought that it'll be better to have this logic in this component in case in the future we'll want to implement the same approach for other tabs: `Reports`, `Staff` ...



[Redacted] 21

🗨️ ⋮

I agree with [@cone](#) the thing is that `UsersTable.js` is already doing too much, and by adding more, that would make it much more difficult to migrate in the future [@SamBelmor](#) and also because, now it would have more capabilities and some of the code it is very dependant, If we move with this solution, we would need to create a new ticket to refactor this soon, otherwise, you can follow a similar approach than the `EmptyMessage` that I'm suggesting that is a similar case for this really big component, What do you think [Redacted]

🗨️ 👍 1



SamBelmor on Jul 28, 2021

Author 🗨️ ⋮

Done



Reply...

app/serializers/fhir/capability_statement_serializer.rb

```
23 +   {  
24 +     'mode': 'server',  
25 +     'resource': [  
26 +       for_appointment,
```



[Redacted], 2021

🗨️ ⋮

👍 great organization



app/models/ability.rb

Hide resolved

```
3 + class Ability
4 +   include CanCan::Ability
5 +
6 +   def initialize(user)
```

SamBelmor on Feb 3 · edited ↵

Putting all the logic inside of the *initializer* is not a good idea.

I've seen projects where they have the abilities in a folder `models/abilities/` and for each of the abilities, it creates a new file. The advantages that we have with this approach are:

1. The code is very clean and intuitive.
2. If in the future we wanna add more abilities we only need to update that specific file.
3. We avoid having a lot of `if, else` statements.

What do you think about this:

1. Create a new file into the folder `models/abilities` and create a file for each ability that we have:

```
models/abilities/
├─ admin.rb
├─ clinic.rb
├─ internal_team.rb
├─ provider.rb
└─ service_user.rb
```

2. The files would look like this:

```
# models/abilities/admin.rb
module Abilities
  module Admin
    def abilities_for_admin
      can :manage, :all
    end
  end
end
```



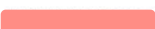
app/models/concerns/login_link.rb

Hide resolved

```
26 + def login_token_expired?
27 +   return true if login_token_generated_at.blank?
28 +
29 +   if Time.zone.now > (login_token_generated_at + token_validity)
```

SamBelmor on Feb 3

This line is not clear, if I want to know what is being validated here, I need to check the methods `login_token_generated_at & token_validity`, so what do you think if you put this line in their own method with an appropriate name? it would be more intuitive, isn't it?

 4

If the point is to make it readable, probably is better to have a variable instead of a method. A method is good when is being used from different places, but for this a simple variable should be enough

Also `expire_token!` uses `save!` so it will return true once the token is cleared so we can do something like this

```
def login_token_expired?
  return true if login_token_generated_at.blank?

  expire_time = login_token_generated_at + token_validity

  return false unless Time.zone.now > expire_time

  expire_token!
end
```

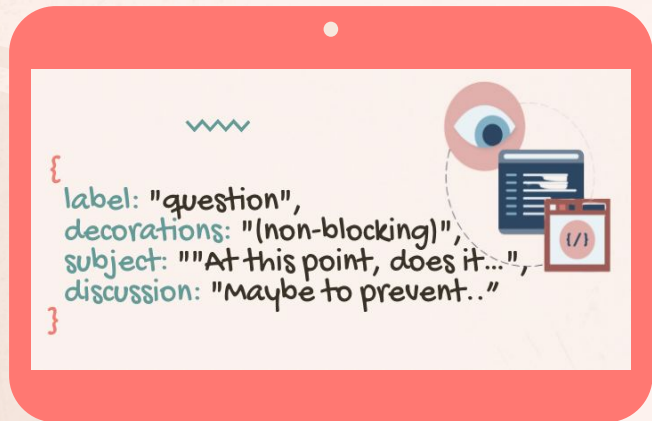
Reply...



Tipos de comentarios



Contenido de un comentario



Cuerpo

```
<label> (decorations): <subject>  
[discussion]
```



@amouse

question (non-blocking): At this point, does it matter which thread has won?
Maybe to prevent a race condition we should keep looping until they've all won?

Etiquetas



praise

Resaltan algo positivo.
Intenta dejar al menos uno
de estos comentarios.



nitpick

Solicitudes triviales
basadas en preferencias.



suggestion

Proponen mejoras al tema
actual.



issue

Estos problemas pueden
ser de cara al usuario o
detrás de escena.



todo

Son cambios pequeños,
triviales, pero necesarios.



question

Pedir al autor una
aclaración o una
investigación.



thought

Representan una idea que
surgió de la revisión.



chore

Estos comentarios hacen
referencia a algún proceso
común.



Decoraciones



(non-blocking)	Un comentario con esta decoración no debe impedir que se acepte el tema en revisión. Esto es útil para las organizaciones que consideran el bloqueo de comentarios de forma predeterminada.
(blocking)	Un comentario con esta decoración debe impedir que se acepte el tema en revisión, hasta que se resuelva.
(if-minor)	Esta decoración le da cierta libertad al autor de que debe resolver el comentario solo si los cambios resultan ser menores o triviales.
(security)	Cuando hay un comentario con esta decoración, se debe revisar esa parte del código más minuciosamente.

Algunos ejemplos

Sam Belmor @SamBelmor · 1 month ago

suggestion: We could add more tests coverage for this...

As I already mentioned in the [app/models/event.rb](#) there are some scenarios that are not being validated correctly, so it would be great if you could expand in this test.

An idea that comes to my mind is this:

1. To have two contexts to test when the data is `valid` and `invalid`
2. **Context valid.** Creates a couple of events with `valid` data and tests that everything creates properly.
3. **Context invalid.** Creates a couple of events with `invalid` data and tests that are not being saved in the db, and check errors.

```
# spec/models/event_spec.rb
require 'rails_helper'

RSpec.describe Event, type: :model do
  let(:time) { Time.now }

  context 'when the data is valid' do
    let!(:event1) { Event.create(start_time: time, end_time: time, title: 'test') }
    let!(:event2) { Event.create(end_time: time, title: 'test2') }
    let!(:event3) { Event.create(end_time: time) }

    it 'creates something' do
      # expect(Event.count).to eq(number)
    end
  end

  context 'when the data is invalid' do
    let(:new_event) { Event.new(start_time: time) }
    let!(:event1) { Event.create(start_time: time_in_the_future, end_time: time) }

    it 'returns something' do
      # expect(new_event.valid?).to eq false
      # expect(new_event.errors.messages).to eq some_error
    end
  end
end
```

Let me know if this makes sense 🐱



Sam Belmor @SamBelmor · 1 month ago

issue: Doing some tests on local I faced with a couple of problems 🐛

First

- When I try to create a new event (in rails console) without the `end_time` it raises an error because when it tries to do the validation it compares `nil <= nil`. So, the first line of this method could be like this:

```
return if end_time.nil?
```

Please let me know if this makes sense 😊

Second

- If this validation fails still build the event. When I create a new event with a `start_time` greater than the `end_time` it builds it anyway. Probably you will need to add the error on `error.base || error.start_time` to avoid building a new event like this.

```
errors.add(:start_time, 'some error message') unless valid_time
```



Sam Belmor @SamBelmor · 1 month ago

suggestion (non-blocking): What do you think about putting this method in a helper to reuse it in other places?

Probably in `spec/support/helpers.rb` 🤔👉



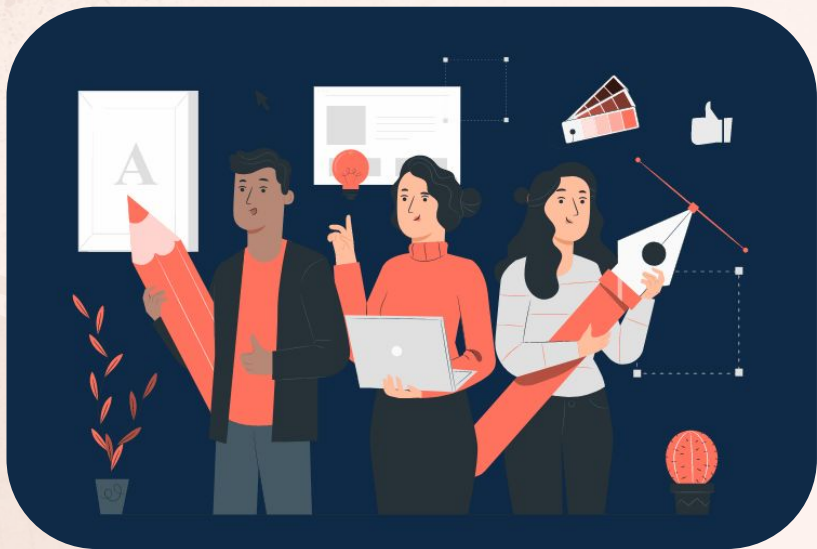
Sam Belmor @SamBelmor · 1 month ago

nitpick (non-blocking): We could do the `validates` in a single line too

Suggested change 🔄

Apply suggestion ▾

```
2 - validates :start_time, presence: true
2 + validates :start_time, :end_time, presence: true
```



Gracias

Sam Belmor

