
A warm welcome to you,
and your 39 trillion bacteria.
uBiome

uBiome

Carlos Betancourt C.
Director of Product Engineering
@betacar

Matías Giménez
Clinical Product UI Engineer

Renzo Canepa
Explorer Product Engineer

Sebastián “Estel” Durán
Product Engineering Lead

uBiome

En uBiome secuenciamos ADN del microbioma humano.


Explorer™


SmartGut™


SmartJane™

Product Engineering team

- Customer facing products, services, and other tools.
- 25 personas, repartidos en 7 equipos.
- Distribuidos entre Estados Unidos, Perú, Chile, y Argentina.
- We like GIFs, memes, video games, asados, and constant learning.


WE'RE
HIRING!

en uBiome

Agenda

- ¿Por qué decidimos cambiar nuestro framework de frontend?
- ¿Por qué optamos por Vue.js?
- Nuestra aplicación y nuestras expectativas
- ¿Cómo enfrentamos la migración?
- Nuestra experiencia hasta ahora
- Desafíos pendientes

¿Por qué decidimos cambiar BlazeJS ?

- **uBiome está creciendo, nuestros sistemas están creciendo, nuestros equipos están creciendo**

    - Necesitamos mantener nuestra capacidad de poner en producción nuevas funcionalidades sin dañar la calidad de nuestro código
        - Cohesion, coupling, maintainability, readability, testing
        - Programación imperativa vs declarativa

    - Necesitamos que nuestros nuevos uBiomers puedan entender nuestros sistemas y trabajar en ellos en el menor tiempo posible
        - Meteor no tiene la mejor curva de aprendizaje

¿Por qué decidimos cambiar BlazeJS ?

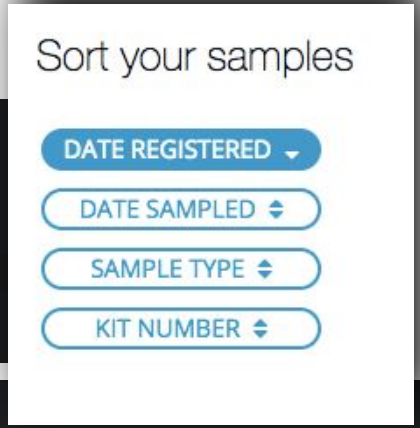

```
11 Template.mySamples.events({
10   'click .js-sort-samples': (e, tmpl) => {
 9     const sortDescending = $(e.currentTarget).hasClass('sort-desc');
 8     $(e.currentTarget).toggleClass('sort-desc');
 7     const sortProperty = $(e.currentTarget).data('sort');
 6
 5     tmpl.state.set('sortProperty', sortProperty);
 4     tmpl.state.set('sortDirection', (sortDescending ? 'desc' : 'asc'));
 3
 2     const sortedSamples = tmpl.sortSamples(Session.get('samples'), sortProperty, sortDescending);
 1     Session.set('samples', sortedSamples);
63
 1     // Handle button display
 2     tmpl.$('.js-sort-samples').removeClass('active');
 3     return $(e.currentTarget).addClass('active');
 4   },
```

¿Por qué decidimos cambiar BlazeJS ?

```
<button
  @click="sortSamples($event, 'samplingTime')"
  :class="getActiveSortingClass('samplingTime')"
>
  Date Sampled
  <i class="fa" :class="getSortDirectionIconClass('samplingTime')"></i>
</button>
```

```
1     methods: {
2       sortSamples(e, sortBy) {
3         this.sortBy = sortBy;
4         this.sortDirection = this.sortDirection === 'desc' ? 'asc' : 'desc';
5       },
6
```

```
1     computed: {
2       filteredSamples() {
3         const { samples, sortBy, sortDirection } = this;
4         return orderBy(samples, [sortBy], sortDirection);
5       },
6
```

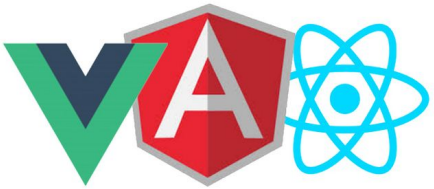


¿Por qué decidimos cambiar BlazeJS ?

- Testear en Meteor es dificil
  - Queremos elevar nuestros estándares de calidad
  - Queremos dormir tranquilos después de un deploy
  - Queremos deployar más seguido

- Meteor ya alcanzó su peak de popularidad
  - Github stars: Meteor 40K, React & Vue.js 90K
  - Dificultad para encontrar nuevos desarrolladores con expertise y ganas de trabajar en Meteor (poco atractivo)
  - Material de aprendizaje desactualizado

¿Por qué optamos por Vue.js?



- Aunque Meteor no soporta actualmente a Vue.js, lo escogimos debido a:
  - Existen librerías que permiten la integración
  - Curva de aprendizaje ideal (HTML, CSS, ES2015)
  - Excelente documentación y recursos para aprender
  - Ecosistema construído por el equipo de Vue.js (routing, state management, tooling)
  - Integración con editores (VSCode, Sublime, Vim)

¿Por qué optamos por Vue.js?

- Sintaxis similar a la de Angular 1
- Migrar de Meteor a Vue.js es más fácil que migrar a React
- Excelente performance
- Pequeño bundle size ~ 20K (gzipped)
  - Imprescindible para usuarios mobile

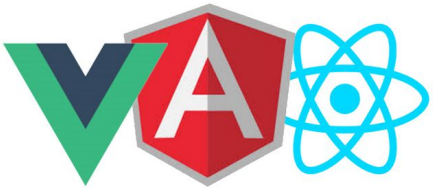
Approximations for GZipped versions

Name	Size
Ember 2.2.0	111K
Ember 1.13.8	123K
Angular 2	111K
Angular 2 + Rx	**143K**
Angular 1.4.5	51K
React 0.14.5 + React DOM	40K
React 0.14.5 + React DOM + Redux	42K
React 15.3.0 + React DOM	43K
React 16.2.0 + React DOM	31.8K
Vue 2.4.2	**20.9K**
Inferno 1.2.2	20K
Preact 7.2.0	**4kb**
Aurelia 1.0.2	63K

¿Por qué optamos por Vue.js?

- Excelente soporte y documentación para unit testing

- Animaciones y transiciones out the box



```
<div class="my-sample__notes my-sample__column">
  <transition name="fade">
    <sample-notes
      :sample="sample"
      :isSpoiled="isSpoiled"
      :updateSessionSample="updateSessionSample"
      :changeNotesMode="changeNotesMode"
      :currentNotesMode="notesMode"
    >
    </sample-notes>
  </transition>
</div>
```

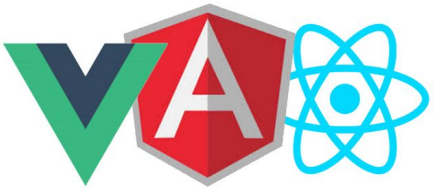
```
.fade-enter-active {
  transition: opacity .5s;
}


.fade-leave-active {
  transition: opacity .2s;
}

.fade-enter, .fade-leave-to {
  opacity: 0;
}
```

¿Por qué optamos por Vue.js?

- Encapsulación de estilos



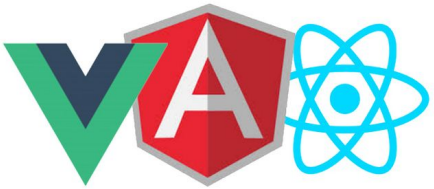
```
<style scoped>
  .example {
    color: red;
  }
</style>

<template>
  <div class="example">hi</div>
</template>
```

```
<style>
  .example[data-v-f3f3eg9] {
    color: red;
  }
</style>

<template>
  <div class="example" data-v-f3f3eg9>hi</div>
</template>
```

¿Por qué optamos por Vue.js?

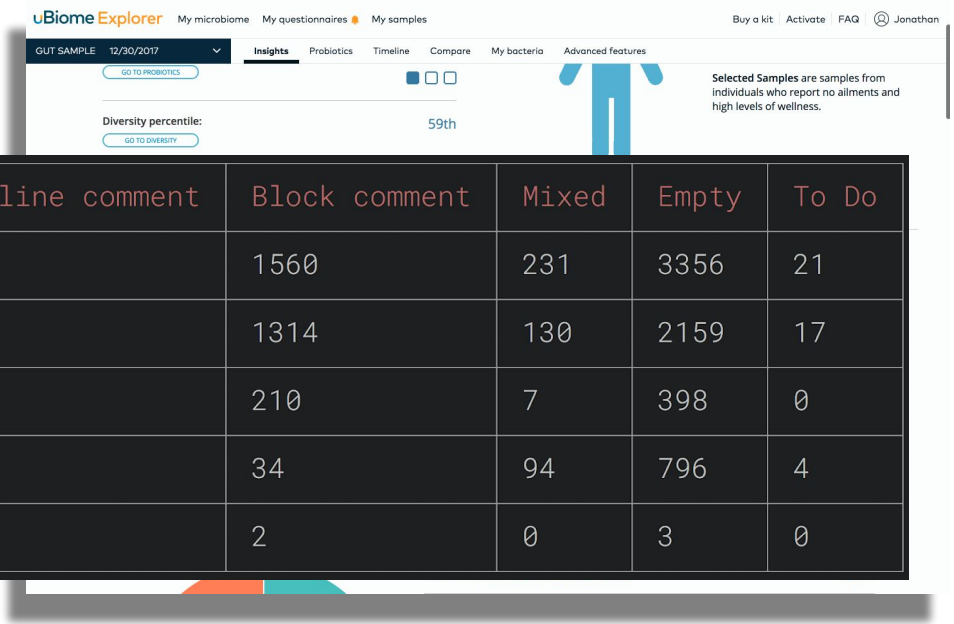


- Diseñado para crear sistemas escalables
  - Reusabilidad a través de componentes
  - Arquitectura (Flux) + programación declarativa + expertise => permiten escalabilidad
  - Herramientas para testear a distintos niveles
  - Buenos defaults => pocas decisiones sin perder flexibilidad

Nuestra aplicación y nuestras expectativas

Explorer

- UIs complejas
  - Múltiples puntos interacción



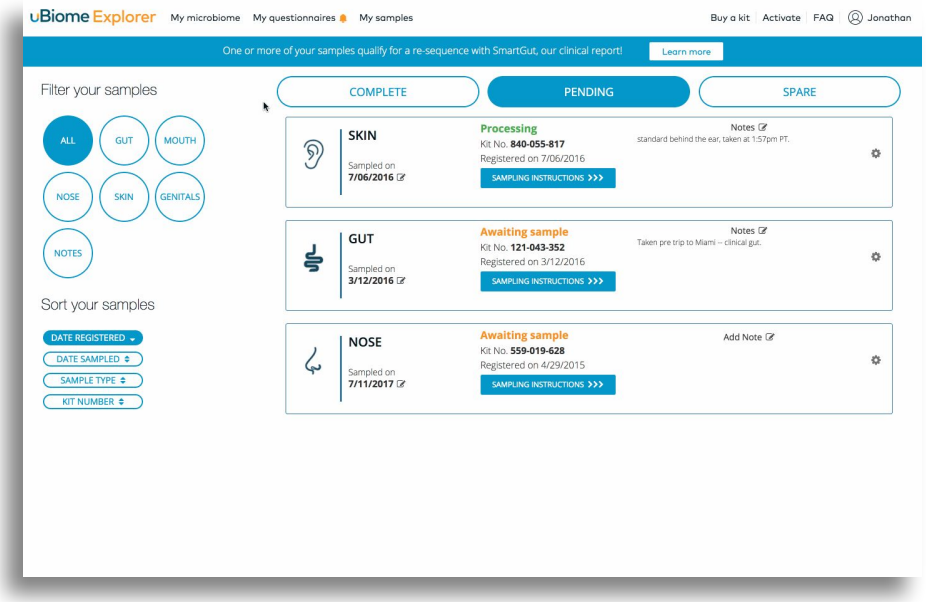
Extension	Physical	Source	Comment	Single-line comment	Block comment	Mixed	Empty	To Do
- Total -	35188	28375	3713	2153	1560	231	3356	21
js	18222	13381	2820	1506	1314	130	2159	17
html	4547	3963	210	0	210	7	398	0
less	12348	10965	681	647	34	94	796	4
css	71	66	2	0	2	0	3	0

código (js, html, less)

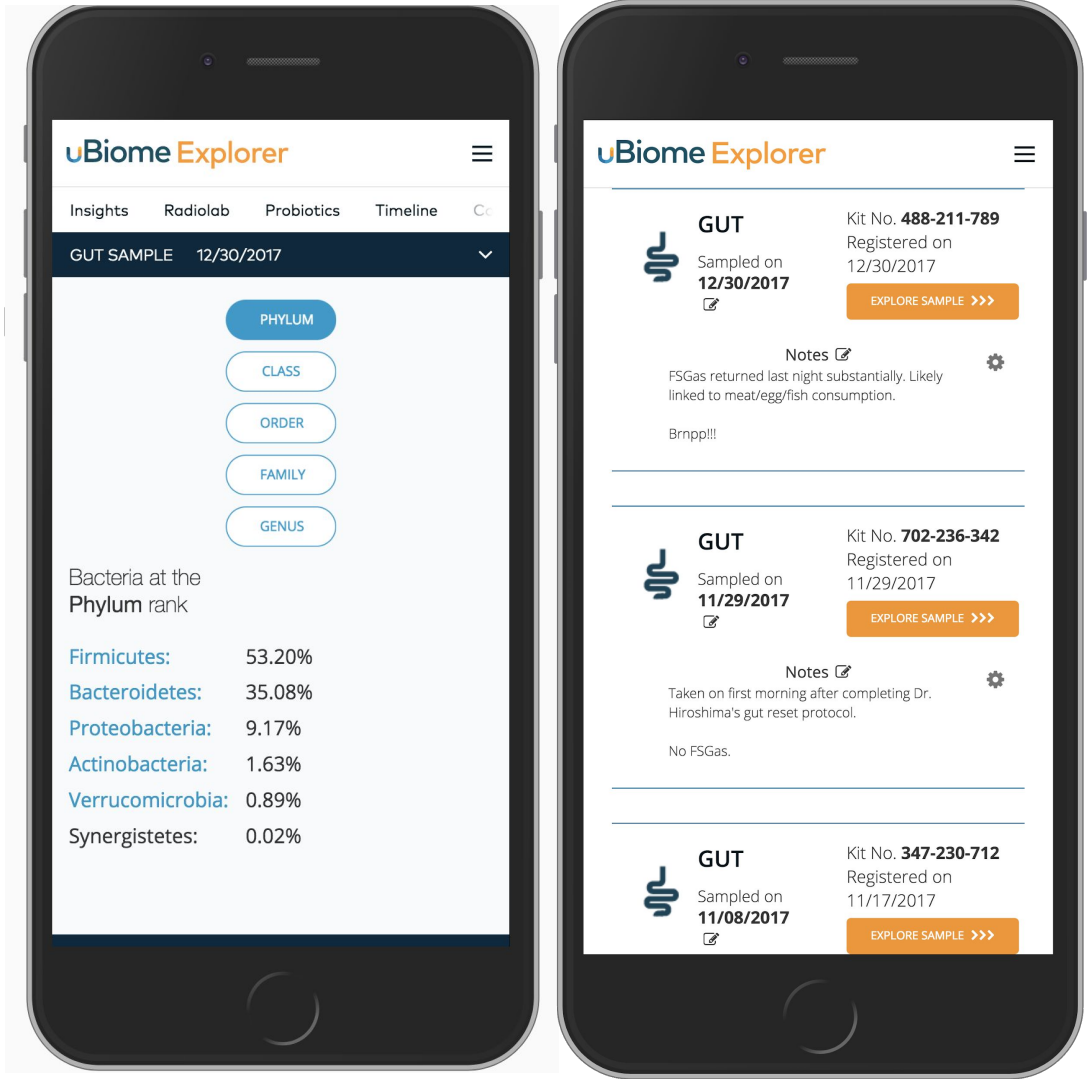
Nuestra aplicación y nuestras expectativas

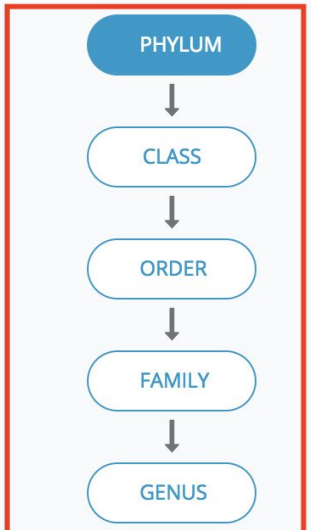
Expectativas

- Velocidad de desarrollo / deployments
- Reducir bugs y aumentar calidad
- Test coverage > 60%
- Mejorar la experiencia de nuestros usuarios móviles
  - Bundle size
  - Performance



- Mobi


uBiome Explorer
Insights
Radiolab
Probiotics
Timeline
GUT SAMPLE 12/30/2017
PHYLUM
CLASS
ORDER
FAMILY
GENUS
Bacteria at the Phylum rank
Firmicutes: 53.20%
Bacteroidetes: 35.08%
Proteobacteria: 9.17%
Actinobacteria: 1.63%
Verrucomicrobia: 0.89%
Synergistetes: 0.02%
uBiome Explorer
GUT
Sampled on 12/30/2017
Kit No. 488-211-789
Registered on 12/30/2017
EXPLORE SAMPLE >>>
Notes
FSGas returned last night substantially. Likely linked to meat/egg/fish consumption.
Brnpp!!!
GUT
Sampled on 11/29/2017
Kit No. 702-236-342
Registered on 11/29/2017
EXPLORE SAMPLE >>>
Notes
Taken on first morning after completing Dr. Hiroshima's gut reset protocol.
No FSGas.
GUT
Sampled on 11/08/2017
Kit No. 347-230-712
Registered on 11/17/2017
EXPLORE SAMPLE >>>

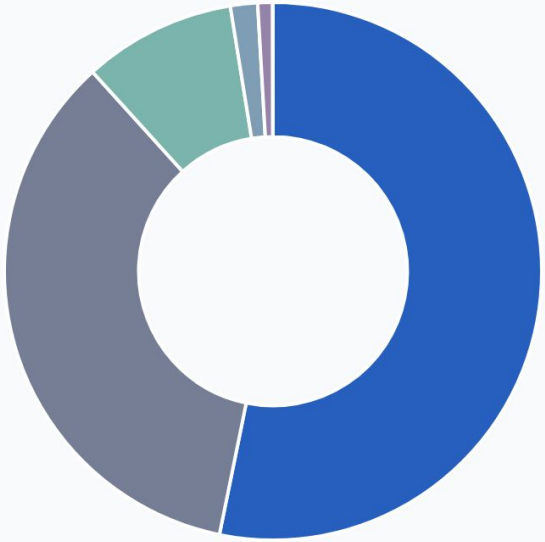


MyBacteria Filter.vue

Bacteria at the **Phylum** rank

Firmicutes:	53.20%
Bacteroidetes:	35.08%
Proteobacteria:	9.17%
Actinobacteria:	1.63%
Verrucomicrobia:	0.89%
Synergistetes:	0.02%

MyBacteriaBreakdown.vue



MyBacteriaChart.vue

MyBacteria.vue

¿Cómo enfrentamos la migración?

- Todo nuevo componente Vue.js debe estar acompañado de test unitarios
  - Jest + vue/test-utils (vue/test-utils = enzyme for Vue.js)

MyMicrobesRankFilter.vue

```
<script>
  export default {
    name: 'my-microbes-rank-filter',

    props: {
      rank: {
        type: String,
        required: true,
      },

      handleRankChange: {
        type: Function,
        required: true,
      },
    },
  };
</script>
```

__tests__/MyMicrobesRankFilter.spec.js

```
describe('MyMicrobes rank filter tests', () => {
  const handleRankChangeStub = jest.fn();
  const wrapper = shallow(MyMicrobesRankFilter, {
    propsData: {
      rank: 'genus',
      handleRankChange: handleRankChangeStub,
    },
  });

  it('handleRankChange should be called after clicking a button', () => {
    const aButton = wrapper.findAll('button').at(0);
    aButton.trigger('click');
    expect(handleRankChangeStub).toBeCalled();
  });
});
```

Nuestra experiencia hasta ahora

- Gracias a li[illegible]e Vue.js con Meteor ha

- Sin embar[illegible]
  - Hot-[illegible]ue una

```
<template>
  <div class="element1"></div>
  <div class="element2"></div>
</template>
```

```
<template name="sample">
  {{#with sample}}
  <div class="sample-sub-container row" >
    <...>
  </div>
  {{#if showInstructionsForType}}
    {{ > sampleInstructions activeSample=this changeSampleDate=clearSampleDateCb}}
  {{/if}}
  {{/with}}
</template>
```

Desafíos pendientes

- Migrar nuestra capa de routing

- Traspasar el manejo de estado a Vuex y acercar nuestra arquitectura a Flux

- Definir guías de estilo y how-to para facilitar el proceso de onboarding y para el traspaso de conocimientos entre equipos
  - Paquetes a usar
  - Cómo testear
  - Configuraciones eslint, prettier, etc

uBiome

Muchas gracias!

¿ Preguntas ?